

Comunicação RC5 com o dsPIC

Vitor Amadeu Souza
vitor@cerne-tec.com.br

Neste artigo, irei apresentar aos leitores um exemplo prático de comunicação via infravermelho, utilizando neste caso o protocolo RC5. A intenção do experimento será controlar um led conectado a uma das portas do microcontrolador, dependendo do dado recebido via este protocolo. Além disso, o dsPIC30F4013 será apresentado dentre suas características assim como o software para programar o mesmo, que será o C com base no mikroC da Mikroelektronika.

O RC5

O protocolo RC5, foi desenvolvido pela Philips e hoje encontramos este facilmente em diversos aparelhos eletrônicos de nosso cotidiano, como TVs, DVDs e Rádios por exemplo. Trata se de um protocolo relativamente simples, onde os dados referentes a tecla pressionada são enviadas via infravermelho modulado em uma frequência típica de 38kHz. A codificação adotada por este protocolo é do tipo manchester, onde o bit 1 e bit 0 podem ser observados na figura 1 e 2 respectivamente.

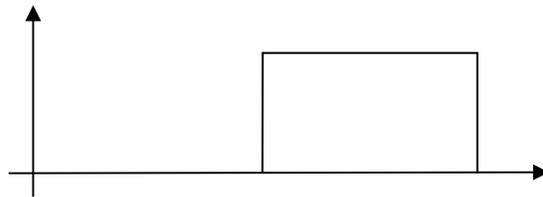


Figura 1 – bit 1 na codificação Manchester

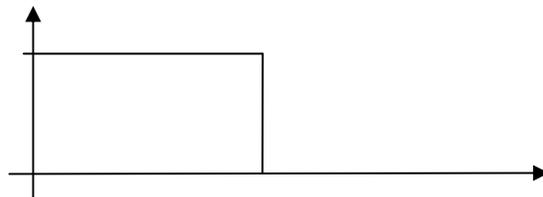


Figura 2 – bit 0 na codificação Manchester

Observe que o bit 1 é formado pela composição de um nível 0 seguido pelo 1. Mesma idéia é válida para o bit 0, onde este é formado pelo nível 1 seguido pelo 0. Note que o tempo total deste bit no RC5 é tipicamente de 1779 us e desta forma, o tempo de cada nível fica em aproximadamente 889 us.

O protocolo RC5 é formado pela composição de todos os bits apresentados na figura 3.

Protocolo RC5													
Bit0	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8	Bit9	Bit10	Bit11	Bit12	Bit13
Start1	Start2	Toggle	Endereço 4	Endereço 3	Endereço 2	Endereço 1	Endereço 0	Comando 5	Comando 4	Comando 3	Comando 2	Comando 1	Comando 0

Figura 3 – Protocolo RC5

Note que ao todo, temos 14 bits que são transmitidos neste protocolo. Os bits de Start1 e Start2, marcam o início da comunicação, estando estes sempre em nível 1. A função do bit de Toggle é indicar se o frame que está sendo enviado é o mesmo ou um diferente. Por exemplo, quando uma tecla é pressionada em um controle remoto, os dados enviados pelo mesmo são contínuos com um intervalo típico de 90 ms, podendo variar de acordo com o controle, observe a figura 4.

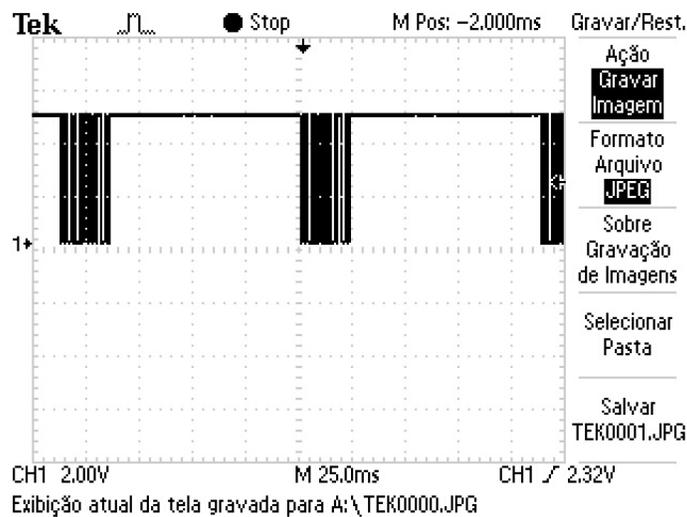


Figura 4 – Intervalo entre um frame e outro

Desta forma, o bit de toggle indica se o dado que está sendo recebido é o mesmo ou se é um novo dado, pois este inverte o seu estado cada vez que uma nova tecla é pressionada.

Os bits de endereço, indicam que tipo de dispositivo o mesmo deseja se comunicar. Ou seja, caso seja uma TV por exemplo, o código é diferente de CD Player por exemplo. Na figura 5 está apresentada alguns endereços típicos de RC5.

Endereço RC5	Dispositivo
0	TV1
1	TV2

3	Vídeo
5	VCR1
6	VCR2

Figura 5 – Endereços típicos RC5

Os bits de comando estão diretamente associados ao de endereço, onde no caso de uma TV, o mesmo indicaria o canal no qual o telespectador deseja ir por exemplo e no caso de um rádio, a faixa de um CD no qual o ouvinte deseja escutar.

Na figura 6 está apresentada uma transmissão de um frame em RC5, neste caso sendo enviado o endereço 0 por se tratar de um controle de TV e comando 0, por estar sendo pressionada a tecla 0.

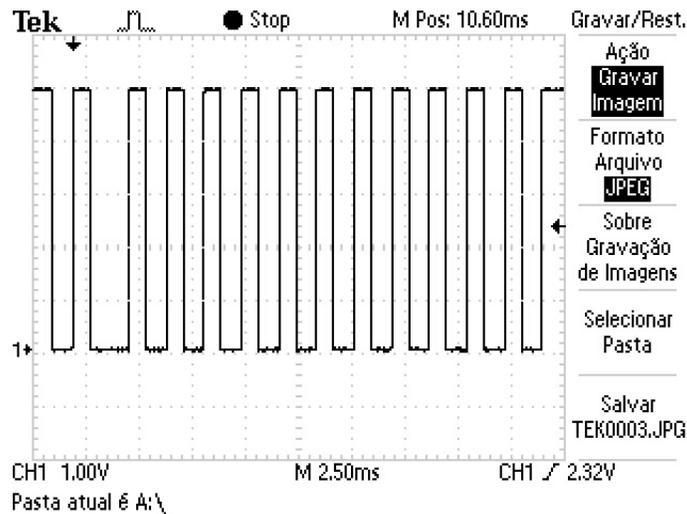


Figura 6 – Protocolo RC5

Note que a comunicação inicia pelos dois bits de Start, seguida do bit de toggle, 5 bits de endereço e finalmente os 6 bits de comando.

O Receptor RC5

Conforme informado anteriormente, os dados trafegados em infravermelho no RC5 estão modulados a uma frequência de 38 kHz. Existem receptores que já fazem esta filtragem deste sinal e o mesmo já informa em sua saída, os dados na forma digital da mesma forma em que está apresentado na figura 6 por exemplo. Neste experimento, irei utilizar o receptor TSOP4838, onde a sua pinagem e formato estão apresentados na figura 7.

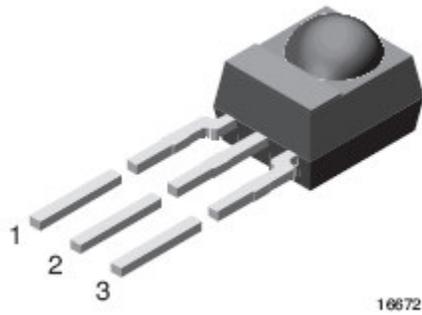


Figura 7 – Receptor TSOP4838

Estes receptores são fabricados pela Vishay e mais dados sobre o funcionamento do mesmo podem ser obtidos no site do fabricante, neste caso no endereço www.vishay.com. A grande vantagem em se utilizar este receptor, é a facilidade de encontrar o mesmo no comércio eletrônico.

Processador

Para esta experiência será utilizado o dsPIC30F4013 da Microchip (www.microchip.com). Vejamos algumas características deste microcontrolador na tabela 1.

Características
3 fontes de interrupção externa
8 níveis selecionáveis de prioridade
48 kB de memória de programa
2 kB de memória RAM
1 kB de memória EEPROM
Processamento de até 30 MIPS
I2C
CAN 2.0
UART

Tabela 1 – Características do dsPIC

A pinagem deste microcontrolador pode ser observada na figura 8.

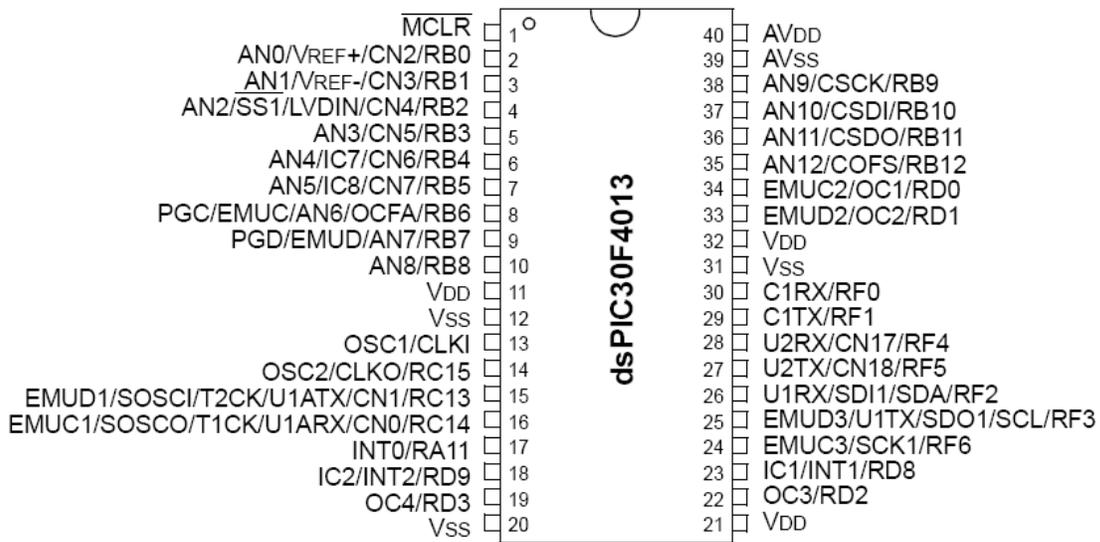


Figura 8 – Pinagem do dsPIC30F4013

Esquema Elétrico

Este experimento foi desenvolvido utilizando a placa dsPIC MASTER da Cerne Tecnologia, conforme pode ser observado na figura 9:

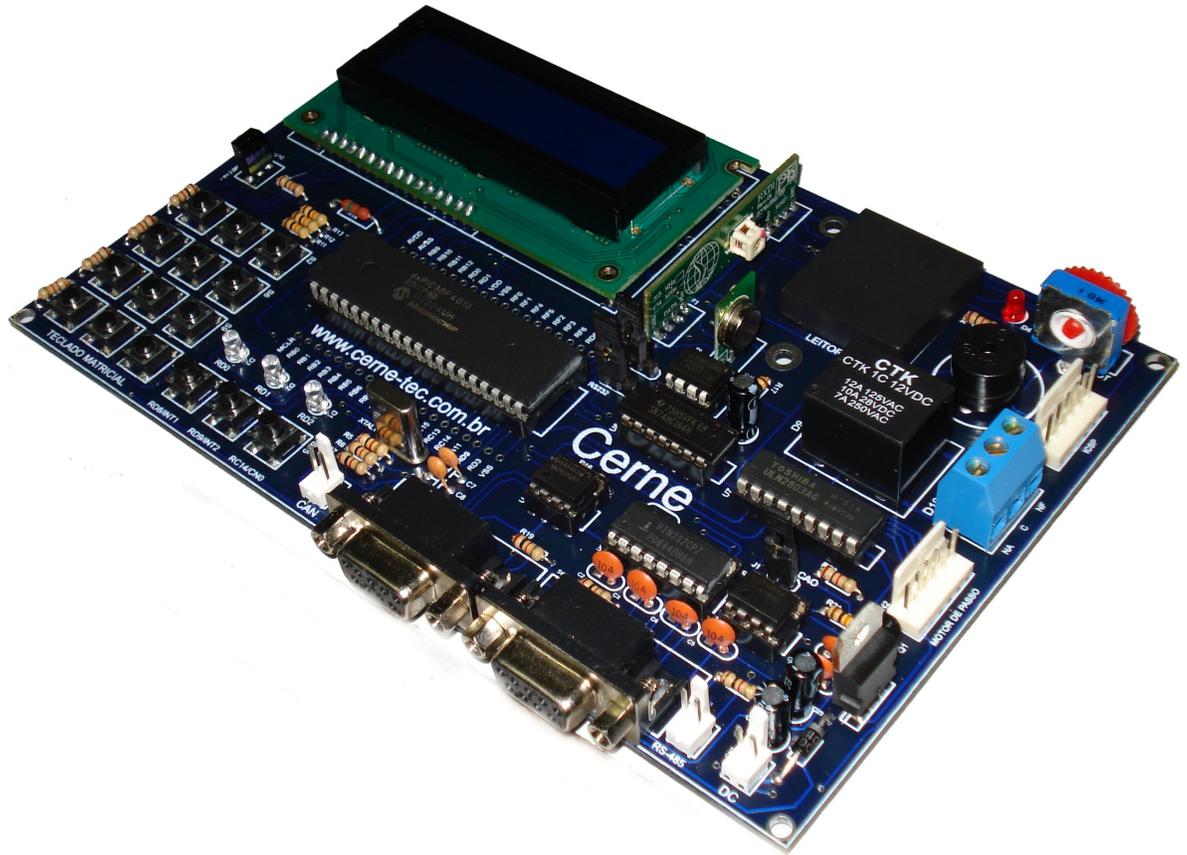


Figura 9 – Placa dsPIC MASTER

O esquema elétrico, para os leitores que desenvolverem este projeto em protoboard ou placa padrão está apresentado na figura 10.

Obs: Para os leitores interessados na placa dsPIC MASTER, a mesma se encontra a venda no site www.sabermarketing.com.br.

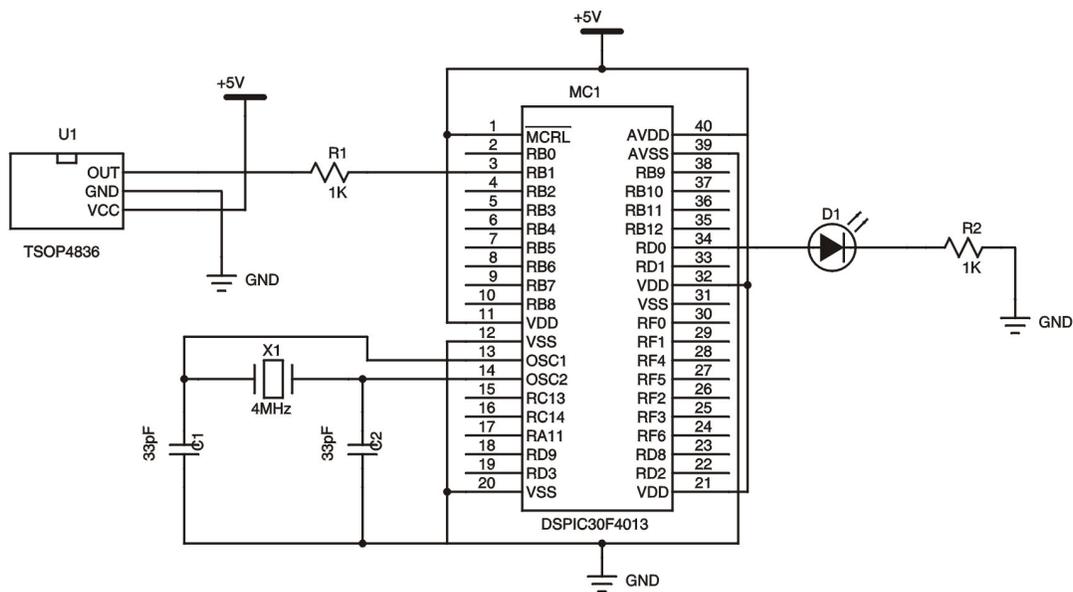


Figura 10 – Esquema Elétrico

Observe que apesar das poucas linhas utilizadas, este microcontrolador foi escolhido de forma a permitir facilmente a expansão do sistema, colocando displays lcd ou gráfico por exemplo. A lista de material para este experimento está apresentada na tabela 2.

Quantidade	Componente	Referência
2	Capacitor 33pF	X1
1	dsPIC30F4013	MC1
1	Led 3mm	D1
2	Resistor 1K	R1, R2
1	TSOP4836	U1

Tabela 2 – Lista de Material

Fluxograma

O fluxograma que irá reger o funcionamento deste sistema está apresentado na figura 11. Note que o sistema fica constantemente verificando se há o início de um novo frame na linha RB1 e caso seja detectado tal frame e o código de comando seja o 1, o led será acionado e caso seja o 2, o mesmo será desligado.

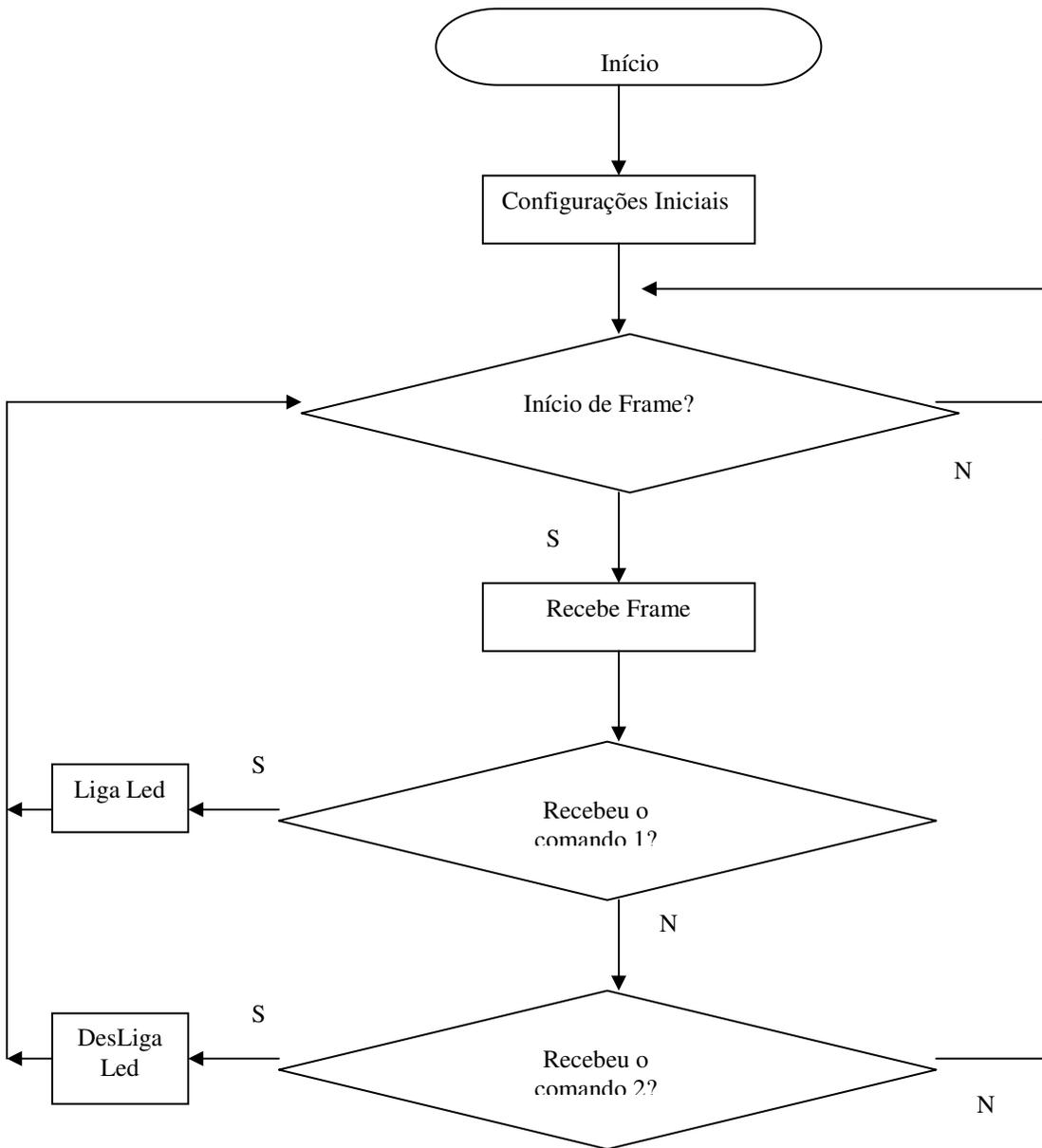


Figura 11 – Fluxograma

Ambiente de Programação

O software utilizado neste experimento foi o dsPIC mikroC. A versão DEMO deste software pode ser baixada gratuitamente no site www.mikroe.com. Após a instalação deste software, inicialize o mesmo, teremos acesso a tela da figura 12.

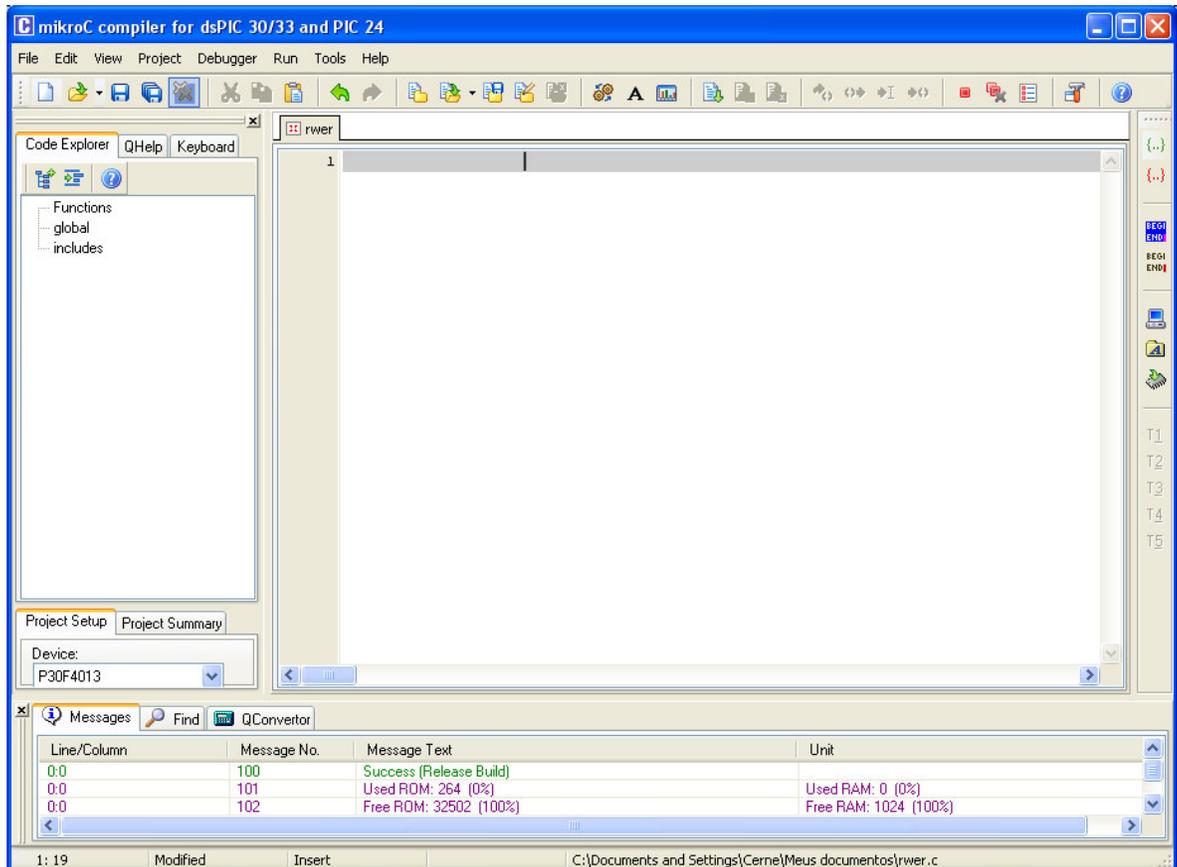


Figura 12 – Ambiente dsPIC mikroC

O mikroC trabalha com o conceito de projetos, onde para compilar um arquivo em C, precisamos criar um projeto para esta função. Vá ao menu Project -> New Project. A tela da figura 13 será apresentada.

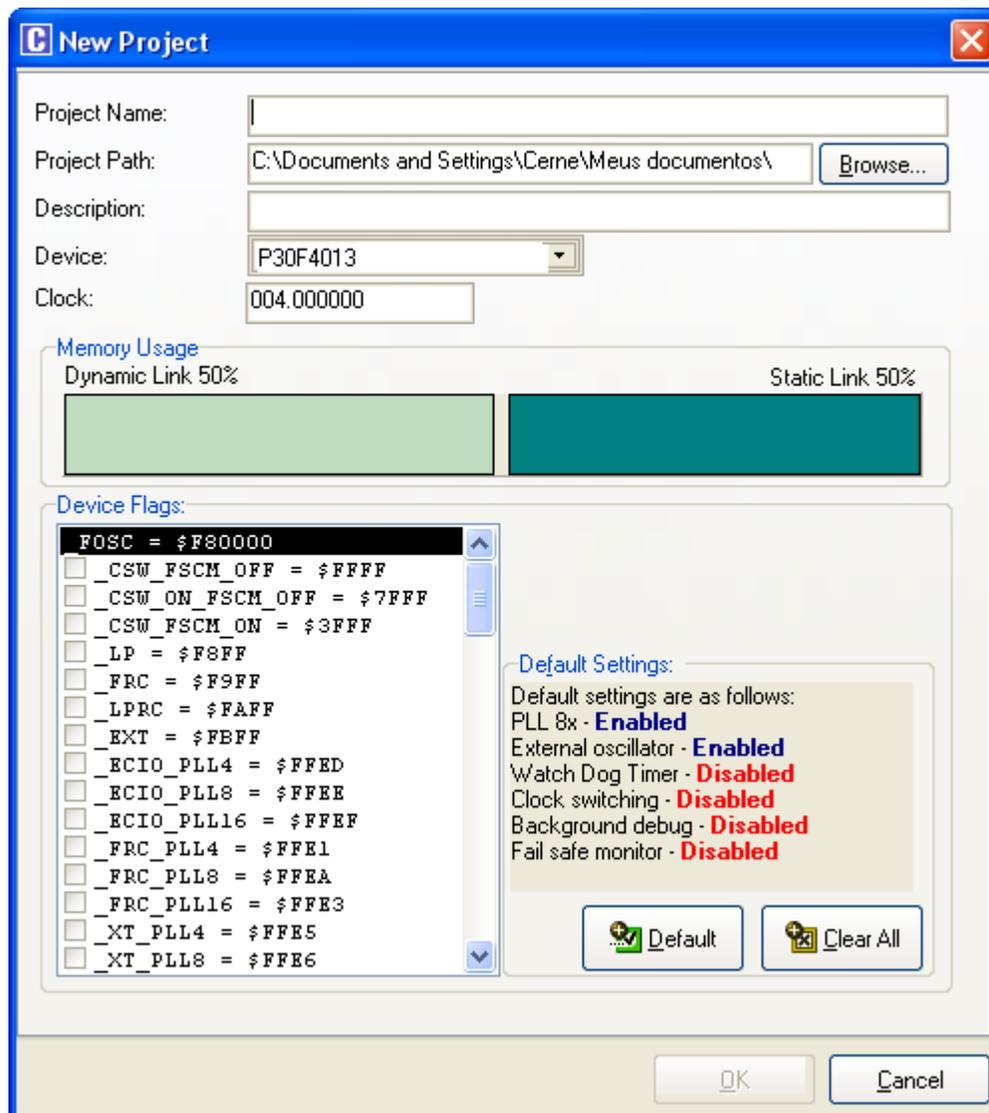


Figura 13 – Criando um novo projeto

Em Project Name informe o nome do projeto, neste caso por exemplo rc5. Em project path escolha a pasta onde o mesmo ficará salvo. O campo Description é opcional e serve para informarmos o que o projeto faz, podendo ficar em branco por enquanto. O campo device serve para informar qual dispositivo está sendo utilizado no projeto. Escolha neste caso o dsPIC30F4013. No campo clock, devemos informar a velocidade da fonte de clock ligada ao dsPIC. De acordo com o esquema elétrico, que informa a velocidade de 4 MHz informe este valor neste campo. Através do campo Device Flags poderemos fazer a configuração dos bits de configuração do dsPIC. Marque neste campo os flags apresentados na tabela 3.

Device Flags Selecionados
_CSW_FSCM_OFF
_XT
_WDT_OFF
_MCLR_DIS

Tabela 3 – Ajustando os configurations bits

Obs: Recomendo ao leitor que verifique o datasheet deste microcontrolador afim de uma explanação melhor dos configurations bits.

Feito isso, pressione Ok. Teremos a tela da figura 13 apresentada:

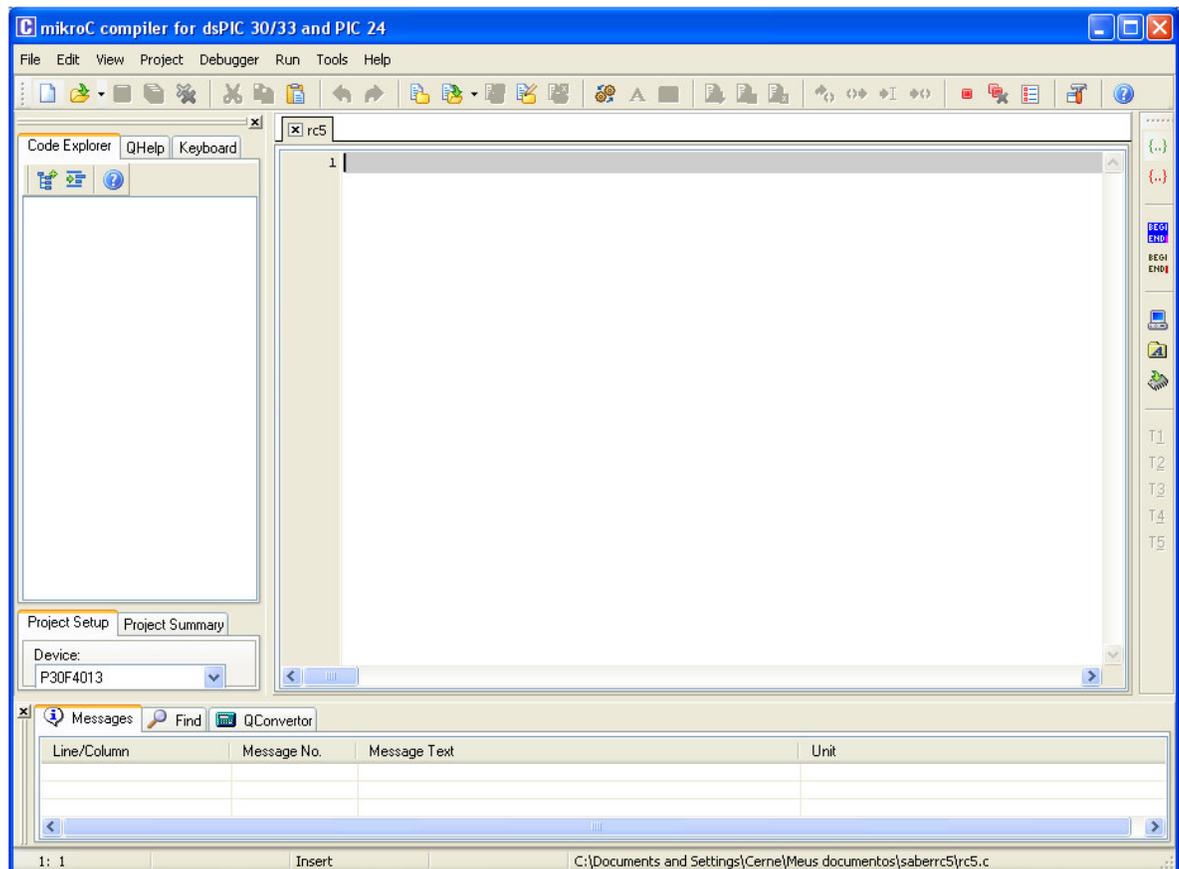


Figura 13 – Tela do mikroC


```

        delay_us(885);
        decodifica_bit();
        if (last_rx) sistema|=2;           //Bit 1 de sistema

        delay_us(885);
        decodifica_bit();
        if (last_rx) sistema|=1;         //Bit 0 de sistema

        delay_us(885);
        decodifica_bit();
        if (last_rx) comando|=32;        //Bit 5 de comando

        delay_us(885);
        decodifica_bit();
        if (last_rx) comando|=16;       //Bit 4 de comando

        delay_us(885);
        decodifica_bit();
        if (last_rx) comando|=8;        //Bit 3 de comando

        delay_us(885);
        decodifica_bit();
        if (last_rx) comando|=4;        //Bit 2 de comando

        delay_us(885);
        decodifica_bit();
        if (last_rx) comando|=2;        //Bit 1 de comando

        delay_us(885);
        decodifica_bit();
        if (last_rx) comando|=1;        //Bit 0 de comando

        if (comando==1)                  //O comando é 1?
            led=1;                        //Sim, então liga o led

        if (comando==2)                  //O comando é 2?
            led=0;                        //Sim, então desliga o led

    }
}while(1);
}

```

Box 1 – Código para execução do projeto

Vamos analisar o código com mais detalhes, a começar pela rotina *decodifica_bit*. Esta função terá a incumbência de saber se o dado que está sendo recebido é o 1 ou o 0 de acordo com o formato utilizado na codificação Manchester. Já no bloco main (principal), logo após o início do mesmo, a direção dos pinos é ajustada através dos registradores TRISB e TRISD, sendo um definido como entrada que é o caso do PORTB e outro como saída que é o TRISD. Em seguida, o programa fica preso em loop testando continuamente o estado do pino de entrada do receptor de RC5. No momento em que o mesmo vai ao nível 0, inicia a recepção dos 14 bits que compõem o protocolo RC5 e salva este resultado nas variáveis comando e sistema. Após a recepção completa do dado, a variável comando é testada, para saber se o seu conteúdo é 1 ou 2 e dependendo do mesmo, acionar ou não a saída do microcontrolador, permitindo acionar ou não o led.

Conclusão

Este tipo de comunicação é muito utilizada nos dias atuais e com o advento de *casas inteligentes* podemos aperfeiçoar mais este sistema, permitindo por exemplo o acionamento de lâmpadas de uma residência através de um controle remoto ou até mesmo o controle de sistemas onde o contato feito pelo ser humano possa ser prejudicial a sua saúde.

