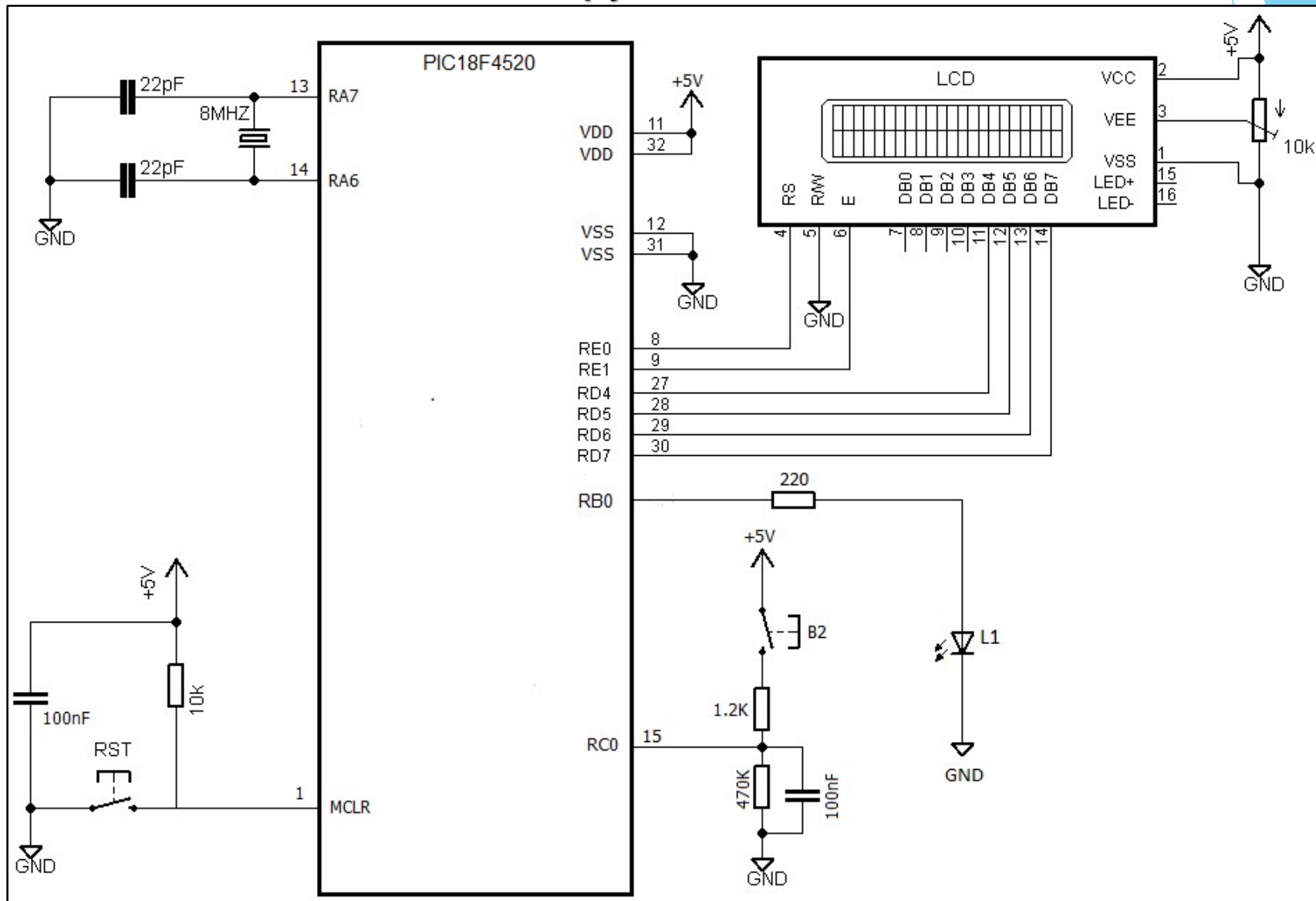
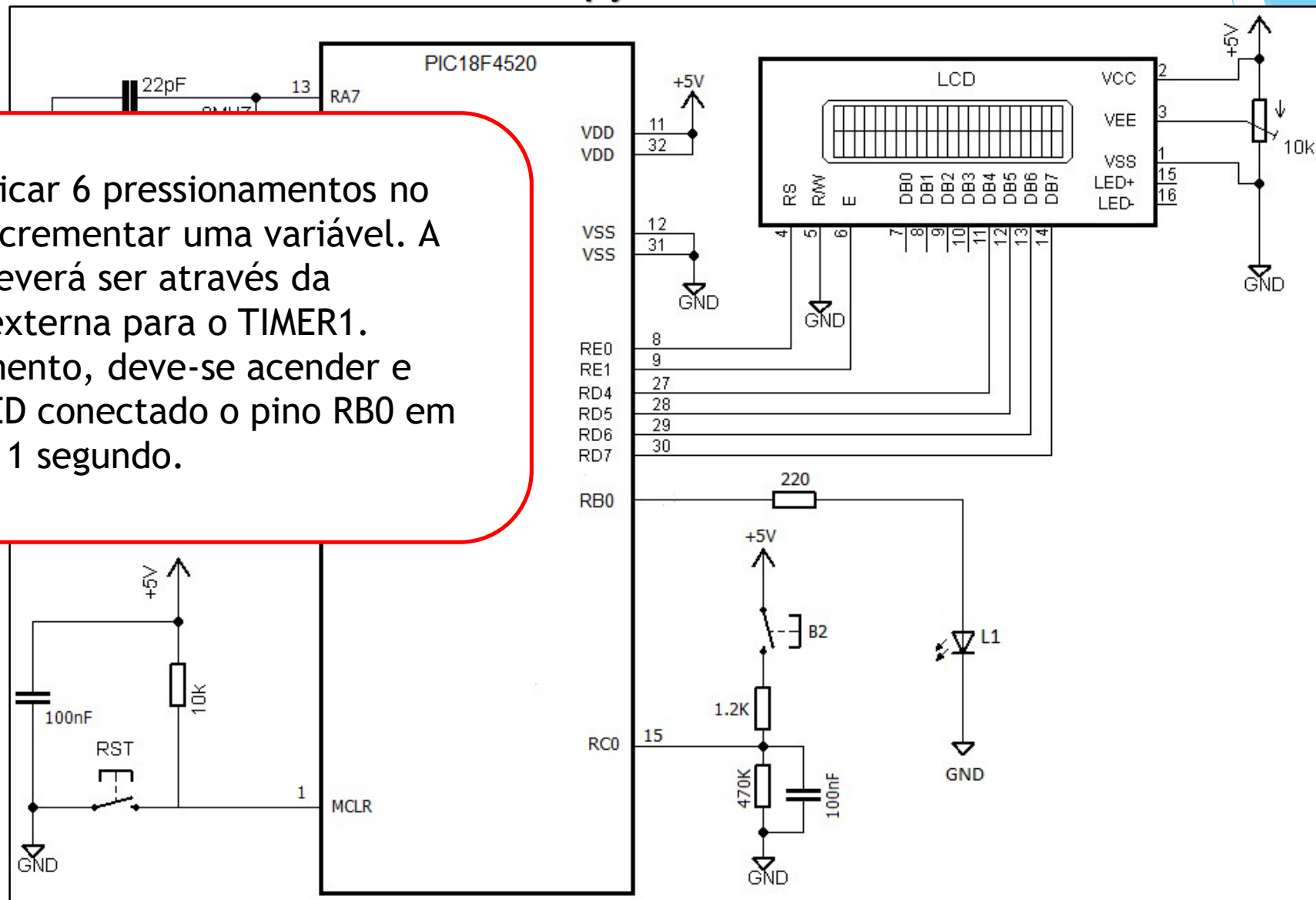


Interrupção do TIMER1



Interrupção do TIMER1

Deve-se verificar 6 pressionamentos no botão B2 e incrementar uma variável. A verificação deverá ser através da interrupção externa para o TIMER1. No processamento, deve-se acender e apagar um LED conectado o pino RB0 em intervalos de 1 segundo.



Aqui vai ter uma explicação sobre os pulsos

Interrupção do TIMER1

Habilitando a Interrupção

```
INTCON = 0b11000000;
```

```
/*Habilita Interrupção Global e de periféricos
```

```
GIE = 1;
```

```
PEIE = 1;
```

```
TMR0IE = 0;
```

```
INT0IE = 0;
```

```
RBIE = 0;
```

```
TMR0IF = 0;
```

```
INT0IF = 0;
```

```
RBIF = 0; */
```

Interrupção do TIMER0

Configurando o TIMER0

```
T1CON = 0b00000111;    /*RD16 = 0 - Escrita/Leitura no modo 8 bits;
                          T1RUN = 0 - Não se aplica;
                          T1CKPS1 e T1CKPS0 = 0 - Prescaler de 1:1;
                          T1OSCEN = 0 - Não se aplica;
                          T1SYNC = 1 - Não se aplica;
                          TMR1CS = 1 - Seleciona fonte de clock externa pelo pino RC0/T13CKI;
                          TMR1ON = 1 - Habilita TIMER1;
                          */

TMR1IE = 1;              //Habilita interrupção para o TIMER 1

//Inicia o TIMER1 com o valor 65530 - 0xFFFF
TMR1H = 0xFF;            //Valor inicial para o registrador TMR1H
TMR1L = 0xFA;            //Valor inicial para o registrador TMR1L
```

Interrupção do TIMER1

Tratamento para Interrupção do TIMER1

```
void __interrupt() TMR1(void)
{
    if (TMR1IF)    //Se o bit TMR1IF for igual a 1
    {
        //trata a interrupção

        TMR1H = 0xFF;           //Valor inicial para o registrador TMR1H
        TMR1L = 0xFA;           //Valor inicial para o registrador TMR1L

        TMR1IF = 0; //Limpa o flag da interrupção
    }
}
```