

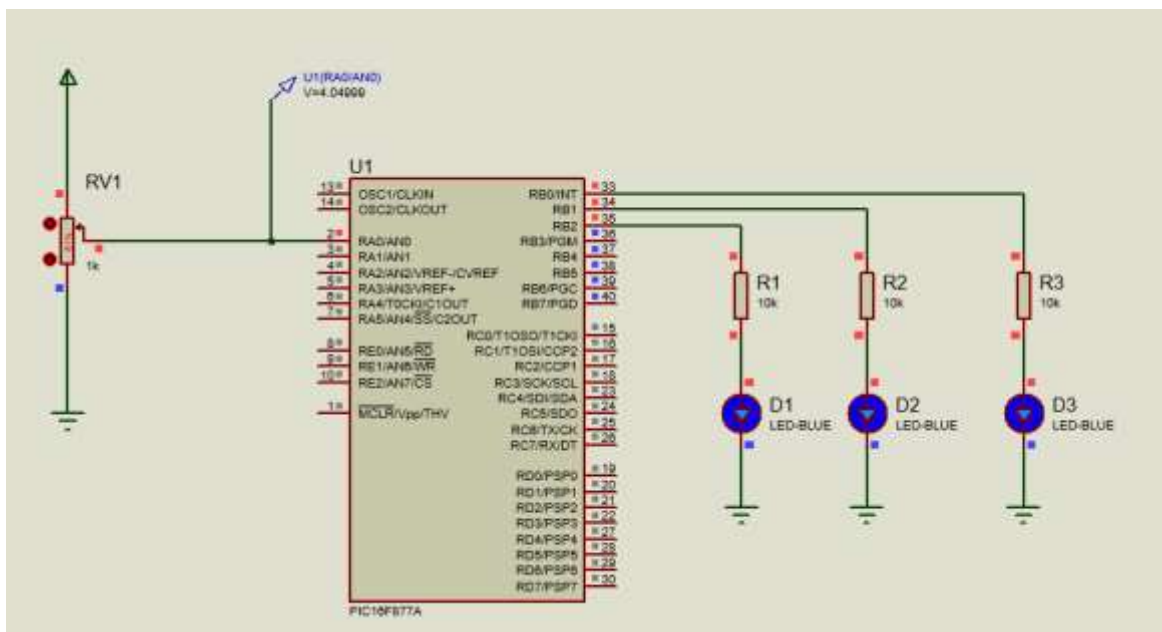
Lista de Exercícios – Aula 06

QUESTÃO 01. Utilize o PIC16F877A para que acenda três leds conforme os seguintes indicativos de tensão:

Led 1 – Acender com 1V aplicado ao pino AN0

Led 2 – Acender com 3V aplicado ao pino AN0

Led 3 – Acender com 4V aplicado ao pino AN0



```
#define LED1 PORTB.RB2
```

```
#define LED2 PORTB.RB1
```

```
#define LED3 PORTB.RB0
```

```
unsigned int leitor = 0;
```

```

void main() {

    ADCON0 = 0b10000001;    // Configura o
    registrador ADCON0

    ADCON1 = 0b01001110;    // Configura o
    registrador ADCON1


    TRISB = 0; // Define o PORTB como saída

    PORTB = 0;    // Define o PORTB como
    desligado


    while(1){

        leitor = ADC_READ(0); // Faz a leitura do pino
        A0 e armazena o valor na variável leitor


        if(leitor > 0){ // Condição para não ascender
        nenhum LED enquanto for maior que zero


        LED1 = 0;

        LED2 = 0;    // LEDS desligados

        LED3 = 0;


        }

        if(leitor > 205){ // Condição para ascender o

```

LED1 quanto atingir o valor que corresponde
a 1V

```
LED1 = 1;
```

```
LED2 = 0;      // Só o LED1 ligado
```

```
LED3 = 0;
```

```
}
```

if(leitor > 614){ // Condição para ascender o
LED2 quanto atingir o valor que corresponde
a 3V

```
LED1 = 1;
```

```
LED2 = 1;      // LED1 e LED2 ligados
```

```
LED3 = 0;
```

```
}
```

if(leitor > 819){ // Condição para ascender o
LED3 quanto atingir o valor que corresponde
a 4V

```
LED1 = 1;
```

```
LED2 = 1;      // LED1, LED2 e LED3 ligados
```

```
LED3 = 1;
```

```
}
```

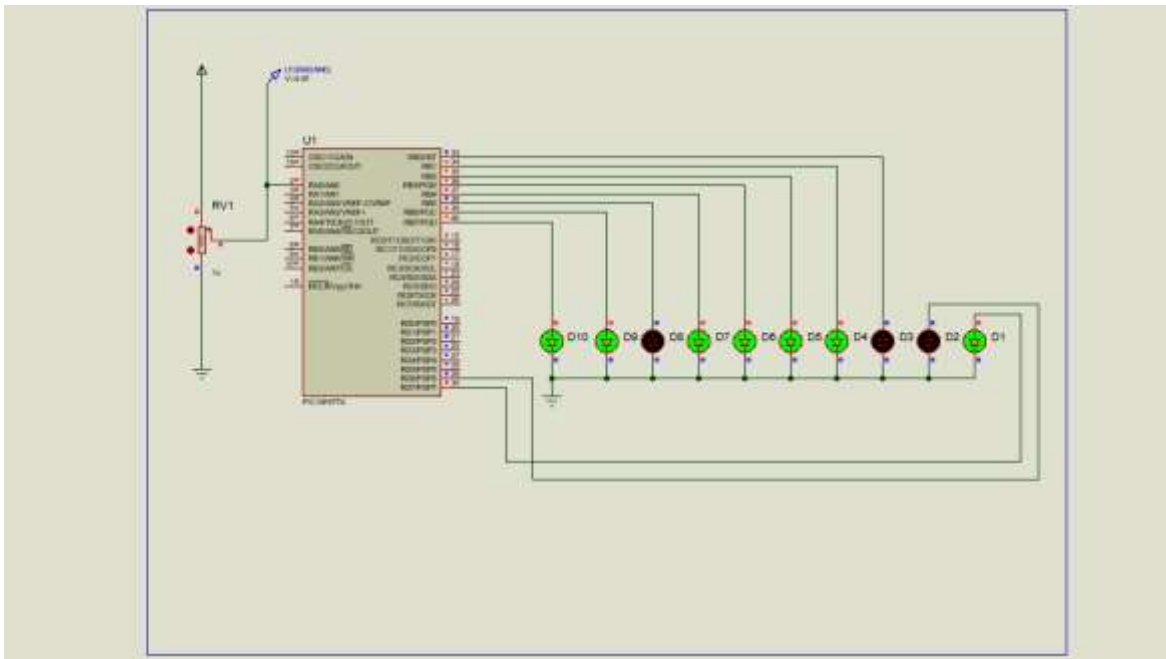
```
Delay_ms(100);
```

```
}
```

```
}
```

QUESTÃO 02. Utilize o PIC16F877A para que apresente nos ports B e D o valor binário correspondente ao valor decimal presente nos registradores ADRESH e ADRESL. Utilize o portB para apresentar o resultado binário do registrador ADRESH e no portD o resultado binário do registrador ADRESL. Utilize ADRESH e ADRESL justificados a esquerda.

Atenção: Dica – Escreva nos ports B e D após testar o fim do conversão A/D, ou seja, teste o bit GO_DONE, caso o mesmo atinja 0 dizemos que a conversão foi finalizada e o resultado pode ser armazenado nos ports. Não esqueça de levar o GO_DONE a 1 após a finalização da conversão para que uma nova conversão seja realizada.



```
void main(){
```

```
    TRISB = 0;
```

```
    TRISD = 0;
```

```
    ADCON0.ADON = 1;    //Habilita o conversor AD
```

```
    ADCON0.ADFM = 0;    // Justifica a Esquerda
```

```
    while(1) {           //Inicia loop infinito
```

```
        ADCON0.GO_DONE = 1;           //Inicia a conversao
```

```
        while (!ADCON0.GO_DONE) {      //Aguarda o fim da conversao
```

```
    }
```

```
PORTB = ADRESH;           //Transfere o valor para porta B
PORTD = ADRESL;           //Transfere valor para porta C
Delay_ms (100);           // Atraso de 100 ms
}
}
```