



```

9
10 while(1){
11     LPC_GPIO1->FIOSET = (1 << LED4_PIN);
12     Delay(10000);
13     LPC_GPIO1->FIOCLR = (1 << LED4_PIN);
14     Delay(10000);
15 }
16 }

```

## 1.1 Pinos que possuem mais de uma função

Vários pinos possuem funções além de entrada/saída digital. Por exemplo podemos ter pinos TX, RX, SCL, SDA, etc. Para configurar se um pino será usado como entrada e saída ou será de função alternativa, usamos o registrador PINSEL.

Register	Controls	Table
PINSEL0	P0[15:0]	<a href="#">Table 8–78</a>
PINSEL1	P0 [31:16]	<a href="#">Table 8–79</a>
PINSEL2	P1 [15:0] (Ethernet)	<a href="#">Table 8–80</a>
PINSEL3	P1 [31:16]	<a href="#">Table 8–81</a>
PINSEL4	P2 [15:0]	<a href="#">Table 8–82</a>
PINSEL5	P2 [31:16]	not used
PINSEL6	P3 [15:0]	not used
PINSEL7	P3 [31:16]	<a href="#">Table 8–83</a>
PINSEL8	P4 [15:0]	not used
PINSEL9	P4 [31:16]	<a href="#">Table 8–84</a>
PINSEL10	Trace port enable	<a href="#">Table 8–85</a>

PINSEL0 to PINSEL9 Values	Function	Value after Reset
00	Primary (default) function, typically GPIO port	00
01	First alternate function	
10	Second alternate function	
11	Third alternate function	

Veja o código abaixo:

```

1
2 #include "LPC17xx.h"
3
4 #define LED_1 (1<<25) //Porta 3.25
5 #define LED_2 (1<<26) //      3.26
6
7 int main()
8 {
9     unsigned int cont,cont2;
10
11     LPC_PINCON->PINSEL7=0;          //porta 3 como GPIO ->
12
13     LPC_GPIO3->FIODIR|=LED_2|LED_1; //configura como saída
14
15     LPC_GPIO3->FIOSET=LED_1; //led apagado
16     LPC_GPIO3->FIOCLR=LED_2; //aceso
17
18     while(1)
19     {
20         if(LPC_GPIO3->FIOPIN&LED_1)
21             {LPC_GPIO3->FIOCLR|=LED_1; LPC_GPIO3->FIOSET|=LED_2;}
22         else
23             {LPC_GPIO3->FIOSET|=LED_1; LPC_GPIO3->FIOCLR|=LED_2;}
24
25         for(cont=0;cont<1524;cont++)
26             for(cont2=0;cont2<65536;cont2++); //delay
27     }
28 }

```