

Campus Araranguá	3aTarefa de Sistemas D	igitais Embarcados
Iome do aluno:	_ Matrícula:	<b>Data:</b> 14/03/2016
isciplina: ARA7560 Professor: Fábio Rodrigues de la Rocha		Turma(s): 0865

## 1 Introdução

Nesta aula vamos estudar como funciona o sistema de Entradas e Saídas do lpc1768, especificamente a parte de entrada e saída digital.

O lpc1768 possui 5 portas de dados de 32 bits P0, P1, P2, P3 e P4. Mas para cada porta apenas alguns dos 32 bits podem ser utilizados. A razão para isso está na tentativa de manter uma compatibilidade com elementos anteriores da família ARM. (Detalhes vide página 118 do Manual do usuário)

```
P0[30:0][1]; P0[14:12] are not available.
P1[31:0][2]; P1[2], P1[3], P1[7:5], P1[13:11] are not available.
P2[13:0];
P3[26:25];
P4[29:28]
```

Para cada porta existem registradores para indicar se vamos ler um valor, escrever um valor 1, escrever um valor 0 e fazer uma máscara para habilitar a alteração de um valor como mostra a tabela abaixo:

Generic Name	Description	Access	Reset value[1]	PORTn Register Name & Address
FIODIR	Fast GPIO Port Direction control register. This register individually controls the direction of each port pin.	R/W	0	FIO0DIR - 0x2009 C000 FIO1DIR - 0x2009 C020 FIO2DIR - 0x2009 C040 FIO3DIR - 0x2009 C060 FIO4DIR - 0x2009 C080
FIOMASK	Fast Mask register for port. Writes, sets, clears, and reads to port (done via writes to FIOPIN, FIOSET, and FIOCLR, and reads of FIOPIN) alter or return only the bits enabled by zeros in this register.	R/W	0	FIO0MASK - 0x2009 C010 FIO1MASK - 0x2009 C030 FIO2MASK - 0x2009 C050 FIO3MASK - 0x2009 C070 FIO4MASK - 0x2009 C090
FIOPIN	Fast Port Pin value register using FIOMASK. The current state of digital port pins can be read from this register, regardless of pin direction or alternate function selection (as long as pins are not configured as an input to ADC). The value read is masked by ANDing with inverted FIOMASK. Writing to this register places corresponding values in all bits enabled by zeros in FIOMASK.	R/W	0	FIO0PIN - 0x2009 C014 FIO1PIN - 0x2009 C034 FIO2PIN - 0x2009 C054 FIO3PIN - 0x2009 C074 FIO4PIN - 0x2009 C094
	<b>Important:</b> if an FIOPIN register is read, its bit(s) masked with 1 in the FIOMASK register will be read as 0 regardless of the physical pin state.			
FIOSET	Fast Port Output Set register using FIOMASK. This register controls the state of output pins. Writing 1s produces highs at the corresponding port pins. Writing 0s has no effect. Reading this register returns the current contents of the port output register. Only bits enabled by 0 in FIOMASK can be altered.	R/W	0	FIO0SET - 0x2009 C018 FIO1SET - 0x2009 C038 FIO2SET - 0x2009 C058 FIO3SET - 0x2009 C078 FIO4SET - 0x2009 C098
FIOCLR	Fast Port Output Clear register using FIOMASK. This register controls the state of output pins. Writing 1s produces lows at the corresponding port pins. Writing 0s has no effect. Only bits enabled by 0 in FIOMASK can be altered.	WO	0	FIO0CLR - 0x2009 C01C FIO1CLR - 0x2009 C03C FIO2CLR - 0x2009 C05C FIO3CLR - 0x2009 C07C FIO4CLR - 0x2009 C09C

```
#include "LPC17xx.h"

.

int main(void){
    SystemInit();
    LPC_GPI01->FIODIR |= (1 << LED4_PIN);
    SysTick_Config(SystemCoreClock/(9600*4) - 1); /* Generate interrupt each 1 ms */</pre>
```

```
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 }</pre>
```

## 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.

Controls	Table
P0[15:0]	Table 8–78
P0 [31:16]	Table 8–79
P1 [15:0] (Ethernet)	Table 8-80
P1 [31:16]	Table 8-81
P2 [15:0]	Table 8–82
P2 [31:16]	not used
P3 [15:0]	not used
P3 [31:16]	Table 8-83
P4 [15:0]	not used
P4 [31:16]	Table 8-84
Trace port enable	Table 8-85
	P0[15:0] P0 [31:16] P1 [15:0] (Ethernet) P1 [31:16] P2 [15:0] P2 [31:16] P3 [15:0] P3 [31:16] P4 [15:0] P4 [31:16]

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"
4 #define LED_1 (1<<25) //Porta 3.25
5 #define LED_2 (1<<26) //
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 saida
14
      LPC_GPIO3->FIOSET=LED_1; //led apagado
15
16
      LPC_GPIO3->FIOCLR=LED_2; //aceso
17
18
      while(1)
19
          if(LPC_GPIO3->FIOPIN&LED_1)
20
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++)</pre>
26
              for(cont2=0;cont2<65536;cont2++); //delay</pre>
27
28 }
```