



Asociación Civil para la Investigación,
Promoción y Desarrollo de los
Sistemas Electrónicos Embebidos

Seminario de Electrónica Sistemas Embebidos



Laboratorio de
Sistemas Embebidos

LPCXpresso - Systick



FACULTAD
DE INGENIERIA
Universidad de Buenos Aires

Ing. Juan Manuel Cruz (jmcruz@hasar.com)

Gerente de Ingeniería de Cia. Hasar SAIC

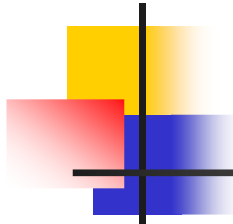


UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Profesor Asociado Ordinario - Técnicas Digitales II TN-FRBA

Profesor Adjunto Interino - Sistemas Embebidos FIUBA

Buenos Aires, 25 de Agosto de 2016

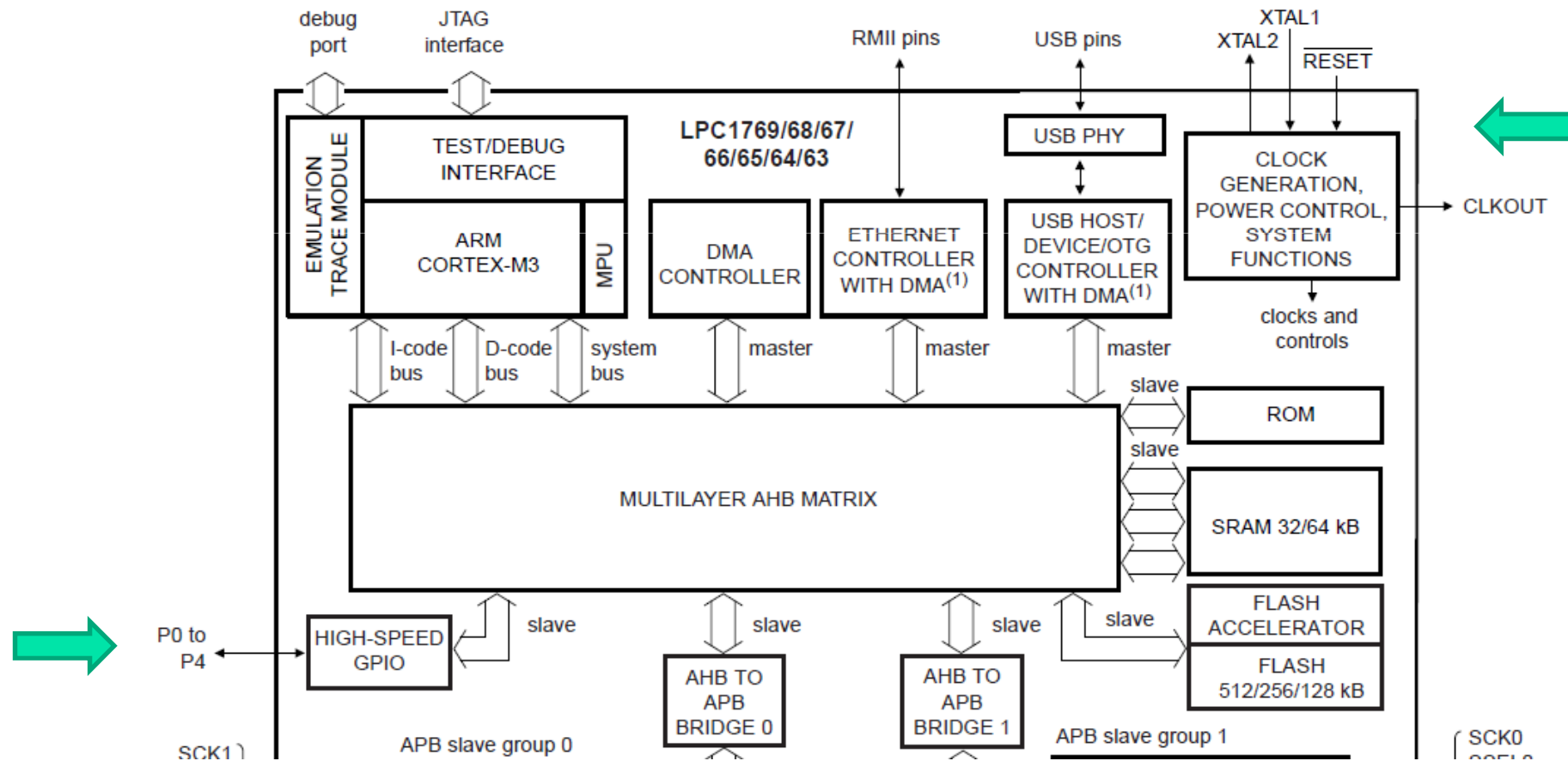


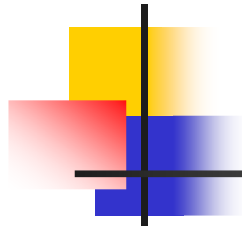
Temario

- SYSTICK
 - Systick del uC LPC1769
 - Led del LPC1769 LPCXpresso Board
 - LED => Actividad Práctica => TP1
 - Systick => Time Out => TP1

<<http://www.sase.com.ar/asociacion-civil-sistemas-embbebidos/escuela/>>

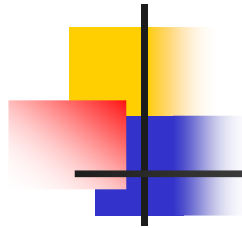
Systick del uC LPC1769





Systick del uC LPC1769

- **ARM Cortex-M3 system tick timer (SYSTICK)**
 - Es un **temporizador** de 24 bits, cuenta en forma **descendente** que en LPC17xx puede recibir clock de AHB interno o desde un pin
 - Cuando su cuenta llega a **cero** genera una **excepción** de **SysTick** (puede programarse para generar excepciones periódicas a intervalos regulares de Xms)
 - Además una excepción de **SysTick** puede generarse por programa
 - En el ambiente de un **Sistema Operativo** (OS o RTOS) puede usarse ésta excepción como **system tick**
 - Se puede usar además como un simple contador



Systick del uC LPC1769

Exception number	IRQ number	Offset	Vector
127	111	0x1FC	IRQ111
.	.	.	.
.	.	.	.
.	.	.	.
18		0x004C	IRQ2
17	2	0x0048	IRQ1
16	1	0x0044	IRQ0
15	0	0x0040	Systick
14	-1	0x003C	PendSV
13	-2	0x0038	Reserved
12			Reserved for debug
11			SVCall
10	-5	0x002C	Reserved
9			
8			
7			
6	-10	0x0018	Usage fault
5	-11	0x0014	Bus fault
4	-12	0x0010	Memory management fault
3	-13	0x000C	Hard fault
2	-14	0x0008	NMI
1		0x0004	Reset
		0x0000	Initial SP value.

Fig 147. Vector table

Systick del uC LPC1769

Table 674. System timer registers summary

Address	Name	Type	Required privilege	Reset value	Description
0xE000E010	CTRL	RW	Privileged	0x00000004	Table 675
0xE000E014	LOAD	RW	Privileged	0x00000000	Table 676
0xE000E018	VAL	RW	Privileged	0x00000000	Table 677
0xE000E01C	CALIB	RO	Privileged	0x000F423F ^[1]	Table 678

[1] SysTick calibration value. This value is specific to LPC17xx devices.

Table 675. SysTick CTRL register bit assignments

Bits	Name	Function
[31:17]	-	Reserved.
[16]	COUNTFLAG	Returns 1 if timer counted to 0 since last time this was read.
[15:3]	-	Reserved.
[2]	CLKSOURCE	Indicates the clock source: 0 = external clock 1 = processor clock.
[1]	TICKINT	Enables SysTick exception request: 0 = counting down to zero does not assert the SysTick exception request 1 = counting down to zero to asserts the SysTick exception request. Software can use COUNTFLAG to determine if SysTick has ever counted to zero.
[0]	ENABLE	Enables the counter: 0 = counter disabled 1 = counter enabled.



Systick del uC LPC1769

Table 676. LOAD register bit assignments

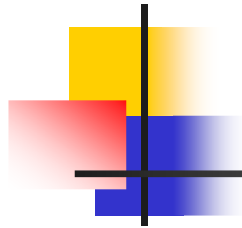
Bits	Name	Function
[31:24]	-	Reserved.
[23:0]	RELOAD	Value to load into the VAL register when the counter is enabled and when it reaches 0, see Section 34.4.4.2.1 .

Table 677. VAL register bit assignments

Bits	Name	Function
[31:24]	-	Reserved.
[23:0]	CURRENT	Reads return the current value of the SysTick counter. A write of any value clears the field to 0, and also clears the SysTick CTRL.COUNTFLAG bit to 0.

Table 678. CALIB register bit assignments

Bits	Name	Function
[31]	NOREF	Indicates whether a separate reference clock is available. This value is factory preset as described in Section 23.1 .
[30]	SKEW	Indicates whether the value of TENMS is precise. This can affect the suitability of SysTick as a software real time clock. This value is factory preset as described in Section 23.1 .
[29:24]	-	Reserved.
[23:0]	TENMS	This value is factory preset as described in Section 23.1 .

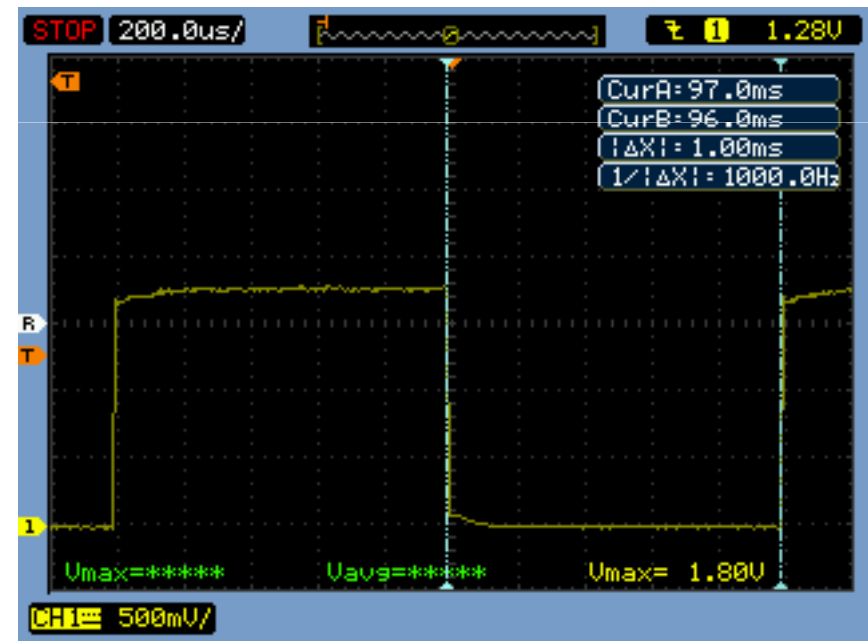
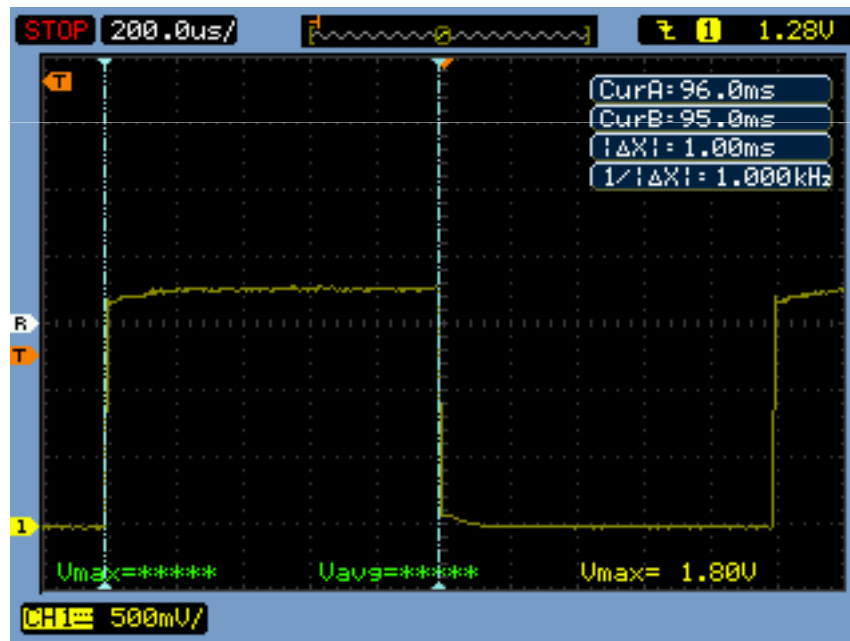


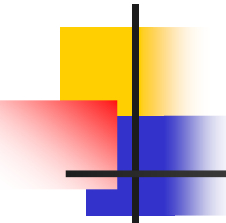
Systick => Actividad Práctica

- Importar el proyecto => **periph_systick**
 - Abrir el archivo **Systick.c** e identificar las funciones:
 - **SysTick_Handler** (manejador de excepciones de Systick p/acciones periódicas o temporizas)
 - **main** (en él las funciones asociadas a Systick)
- Hacer parpadear el **LED** a 1 mS siguiendo las consignas del **TP1**

Systick => Actividad Práctica

- El ORC en **LED** muestra: **1mS** encendido / **1mS** apagado





Systick => Time Out

- Una forma de implementar un **timeout** es mediante una variable que se decrementa periódicamente cuando es distinta de cero (c/1mS), así quien necesite usarla le dará valor inicial y preguntará si ya llegó a cero

```
volatile uint32_t msLED_Toggle = 0;           // counter for 1mS SysTicks
```

```
if (msLED_Toggle == 0) {  
    Board_LED_Toggle(0);  
    msLED_Toggle = 1000;  
}  
else  
    msLED_Toggle --;
```

Systick => Time Out

- El ORC en **LED** muestra: **1S** encendido / **1S** apagado

