

Instituto Tecnológico de Buenos Aires

22.99 LABORATORIO DE MICROPROCESADORES

Guía de Ejercicios N°2: Introducción a Kinetis

Grupo 2

MÁSPERO, Martina	57120
MESTANZA, Joaquín Matías	58288
NOWIK, Ariel Santiago	58309
REGUEIRA, Marcelo Daniel	58300

Profesores

JACOBY, Daniel Andrés
MAGLIOLA, Nicolás

Presentado: 15/08/2019

Índice

Ejercicio 1

- ¿En cuál número de pin del MCU se encuentra el puerto PTA12?

En el datasheet del MCU, la sección 5.3 “K64 Pinouts” figura 37, se indica que el puerto PTA12 es el pin 64.

- ¿Cuáles pines pueden funcionar como entradas analógicas?

En el datasheet del MCU, la sección 5.2 “Unused analog interfaces” tabla 57, indica el nombre del módulo y los pines asociados:

Table 57. Unused analog interfaces

Module name	Pins	Recommendation if unused
ADC	ADC0_DP1, ADC0_DM1, ADC1_DP1, ADC1_DM1, ADC0_DP0/ADC1_DP3, ADC0_DM0/ADC1_DM3, ADC1_DP0/ADC0_DP3, ADC1_DM0/ADC0_DM3, ADC1_SE16/ADC0_SE22, ADC0_SE16/ADC0_SE21, ADC1_SE18	Ground
DAC ¹	DAC0_OUT, DAC1_OUT	Float
USB	VREGIN, USB0_GND, VOUT33 ²	Connect VREGIN and VOUT33 together and tie to ground through a 10 kΩ resistor. Do not tie directly to ground, as this causes a latch-up risk.
	USB0_DM, USB0_DP	Float

Figura 1: Pines de entrada analógica

- ¿Cuántos pines del puerto PTE se encuentran efectivamente disponibles en este modelo de MCU?

En la guía de usuario del kit Freedom, sección 16 “Input/output connectors” se muestran los pines accesibles en la placa de evaluación. Los pines disponibles de uso externo del puerto PTE son el PTE24, PTE25 y el PTE26.

- ¿Cuál es el rango de valores de tensión para detectar un 0 y un 1 lógico en un pin I/O? ¿Se puede enviar 5V a un pin?

En el datasheet del MCU, la sección 2.2.1 “Voltage and current operating requirements” tabla 1 se indican los márgenes de ruido correspondientes:

V_{IH}	Input high voltage <ul style="list-style-type: none">$2.7\text{ V} \leq V_{DD} \leq 3.6\text{ V}$$1.7\text{ V} \leq V_{DD} \leq 2.7\text{ V}$	$0.7 \times V_{DD}$	—	V
		$0.75 \times V_{DD}$	—	V
V_{IL}	Input low voltage <ul style="list-style-type: none">$2.7\text{ V} \leq V_{DD} \leq 3.6\text{ V}$$1.7\text{ V} \leq V_{DD} \leq 2.7\text{ V}$	—	$0.35 \times V_{DD}$	V
		—	$0.3 \times V_{DD}$	V

Figura 2: Márgenes de ruido

Por otro lado, en el datasheet del MCU, la sección 1.4 “Voltage and current operating ratings”, en el cuadro se indica que la máxima tensión de entrada a un pin I/O es de 5.5V.

- ¿Cuál es la máxima corriente que entrega un pin I/O?

En el datasheet del MCU, la sección 1.4 “Voltage and current operating ratings”, en el cuadro se indica que la máxima corriente por un pin I/O es de 25mA.

Ejercicio 2

Con el nivel de optimización original (None - O0), el código funciona normalmente. Con la opción Disassembly se puede obtener el fragmento de código de la función App_run:

```
App_Run:
00000410:  push    {r7, lr}
00000412:  add     r7, sp, #0
00000414:  ldr     r0, [pc, #12] ; (0x424 <App_Run+20>)
00000416:  bl      0x6b0 <delayLoop>
0000041a:  movs    r0, #154 ; 0x9a
0000041c:  bl      0x658 <gpioToggle>
00000420:  nop
00000422:  pop     {r7, pc}
00000424:  rev     r0, r0
00000426:  lsls    r3, r3, #3
```

En el cual se observa que el compilador procesa la función delayloop correctamente. En cambio, utilizando la opción Optimize Most - O3, el compilador elimina esa función dado que bloquea el programa (y puede realizarse lo mismo con un timer). En consecuencia queda solamente el toggle corriendo todo el tiempo sin delay, por lo que no se llega a ver el blink.

```
App_Run:
00000418:  movs    r0, #154 ; 0x9a
0000041a:  b.w     0x614 <gpioToggle>
0000041e:  nop
```

Ejercicio 3

Para utilizar el LED verde del RGB, se modificó el define correspondiente a dicho puerto, en board.h:

```
#define PIN_LED_GREEN    PORTNUM2PIN(PE,26) // PTE26
```

En base a esto, se modificó también el código en App.c:

```
/* Funcion que se llama 1 vez , al comienzo del programa */
void App_Init (void)
{
    gpioMode(PIN_LED_GREEN, OUTPUT);
}

/* Funcion que se llama constantemente en un ciclo infinito */
void App_Run (void)
{
    delayLoop(4*3600000UL);
    gpioToggle(PIN_LED_GREEN);
}
```

El valor en delayloop se modificó por prueba y error, dado que es un while que incrementa una variable, y para conocer el tiempo que demora se midió el del código de blink con el osciloscopio, que era delay-loop(4000000UL). La forma más adecuada de implementar un retardo en el cambio de estado de un pin sería utilizando un timer.