

Lab 1: Pre-Lab Assignment - SOLUTIONS

Fall 2018

Student Name: _____

Date: _____

1. (2 points) Enable the clock of GPIO Port A (for joystick), Port B (for Red LED) and Port E (for Green LED)

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AHB2ENR														RNGEN		AESEN			ADCEN	OTGFSEN					GPIOHEN	GPIOHEN	GPIOFEN	GPIOEEN	GPIODEN	GPIOCEN	GPIOBEN	GPIOAEN
Mask	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1
Value																											1				1	1

MASK = 0x00000013;

RCC->AHB2ENR |= 0x00000013;

Alternative solution (using constants in C):

RCC->AHB2ENR |= (RCC_AHB2ENR_GPIOAEN | RCC_AHB2ENR_GPIOBEN | RCC_AHB2ENR_GPIOEEN);

2. (2 points) Pin Initialization for Red LED (PB 2)

a. Configure PB 2 as Output

GPIO Mode: Input (00), Output (01), Alternative Function (10), Analog (11, default)

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODER	MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]		MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
Mask	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
Value																											0	1				

In this case, you need two masks: one to clear bits 5 and 4 and another to set bit 4. It is also possible to clear only bit 5.

GPIOB Mode Register MASK Value = **0x00000030 and 0x00000010** (in HEX)

GPIOB->MODER &= ~(0x00000030); //Clear bits 4 and 5

GPIOB->MODER |= 0x00000010; //Set bit 4

Alternative solution:

GPIOB->MODER &= ~(0x03<<4);

GPIOB->MODER |= 1<<4;

Alternative solution (using constants in C):

GPIOB->MODER &= ~(GPIO_MODER_MODE2);

GPIOB->MODER |= GPIO_MODER_MODE2_0;

b. Configure PB 2 Output Type as Push-Pull

Push-Pull (0, reset), Open-Drain (1)

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OTYPER	Reserved																OT15	OT14	OT13	OT12	OT11	OT10	OT9	OT8	OT7	OT6	OT5	OT4	OT3	OT2	OT1	OT0
Mask																	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
Value																															0	

GPIOB Output Type Register MASK Value = **0x00000004** (in HEX)

GPIOB->OTYPER &= ~(0x00000004);

Alternative solutions:

GPIOB->OTYPER &= ~(1<<3);

GPIOB->OTYPER &= ~(GPIO_OTYPER_OT2);

c. Configure PB 2 Output Type as No Pull-up No Pull-down

NO PUPD (00, reset), Pullup (01), Pulldown (10), Reserved (11)

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PUPDR	PUPDR15[1:0]		PUPDR14[1:0]		PUPDR13[1:0]		PUPDR12[1:0]		PUPDR11[1:0]		PUPDR10[1:0]		PUPDR9[1:0]		PUPDR8[1:0]		PUPDR7[1:0]		PUPDR6[1:0]		PUPDR5[1:0]		PUPDR4[1:0]		PUPDR3[1:0]		PUPDR2[1:0]		PUPDR1[1:0]		PUPDR0[1:0]	
Mask	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
Value																											0	0				

GPIOB Pull-up Pull-down Register MASK Value = **0x00000030** (in HEX)

GPIOB->PUPDR &= ~(0x00000030);

Alternative solutions:

GPIOB->PUPDR &= ~(0x03<<4);

GPIOB->PUPDR &= ~GPIO_PUPDR_PUPD2;

3. (2 points) Pin Initialization for Green LED (PE 8)

a. Configure PE 8 as Output

GPIO Mode: Input (00), Output (01), Alternative Function (10), Analog (11, default)

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODER	MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]		MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
Mask	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Value															0	1																

In this case, you need two masks: one to clear bits 17 and 16 and another to set bit 16. It is also possible to clear only bit 17.

GPIOE Mode Register MASK Value = **0x00030000 and 0x00010000** (in HEX)

GPIOE->MODER &= ~(0x00030000); //Clear bits 17 and 16

GPIOE->MODER |= 0x00010000; //Set bit 16

Alternative solutions:

GPIOE->MODER &= ~0x03<<16;

GPIOE->MODER |= 1<<16;

Alternative solution (using constants in C):

GPIOE->MODER &= ~GPIO_MODER_MODE8;

GPIOE->MODER |= GPIO_MODER_MODE8_0;

b. Configure PE 8 Output Type as Push-Pull

Push-Pull (0, reset), Open-Drain (1)

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OTYPER	Reserved																OT15	OT14	OT13	OT12	OT11	OT10	OT9	OT8	OT7	OT6	OT5	OT4	OT3	OT2	OT1	OT0
Mask																	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	
Value																								0								

GPIOE Output Type Register MASK Value = **0x00000100** (in HEX)**GPIOE->OTYPER &= ~(0x00000100);****Alternative solutions:****GPIOE->OTYPER &= ~(1<<8);****GPIOE->OTYPER &= ~(GPIO_OTYPER_OT8);****c. Configure PE 8 Output Type as No Pull-up No Pull-down**

NO PUPD (00, reset), Pullup (01), Pulldown (10), Reserved (11)

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PUPDR	PUPDR15[1:0]		PUPDR14[1:0]		PUPDR13[1:0]		PUPDR12[1:0]		PUPDR11[1:0]		PUPDR10[1:0]		PUPDR9[1:0]		PUPDR8[1:0]		PUPDR7[1:0]		PUPDR6[1:0]		PUPDR5[1:0]		PUPDR4[1:0]		PUPDR3[1:0]		PUPDR2[1:0]		PUPDR1[1:0]		PUPDR0[1:0]	
Mask	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Value															0	0																

GPIOE Pull-up Pull-down Register MASK Value = **0x00030000** (in HEX)**GPIOE->PUPDR &= ~(0x00030000);****Alternative solutions:****GPIOE->PUPDR &= ~(0x03<<16);****GPIOE->PUPDR &= ~GPIO_PUPDR_PUPD8;**

4. (2 points) Pin Initialization for Joy Stick

a. Configure PA0 (Center), PA1 (Left), PA2 (Right), PA3 (Up), and PA5 (Down) as Input

GPIO Mode: Input (00), Output (01), Alternative Function (10), Analog (11, default)

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODER	MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]		MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
Mask	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1	1	1	1	1	1	
Value																					0	0			0	0	0	0	0	0	0	0

GPIOA Mode Register MASK Value = **0x00000CFF** (in HEX)**GPIOA->MODER &= ~(0x00000CFF);****Alternative solution (using constants in C):****GPIOA->MODER &= ~(GPIO_MODER_MODE0 | GPIO_MODER_MODE1 | GPIO_MODER_MODE2 | GPIO_MODER_MODE3 | GPIO_MODER_MODE5);**

b. Configure PA0 (Center), PA1 (Left), PA2 (Right), PA3 (Up), and PA5 (Down) as Pull-down

NO PUPD (00, reset), Pullup (01), Pulldown (10), Reserved (11)

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PUPDR	PUPDR15[1:0]		PUPDR14[1:0]		PUPDR13[1:0]		PUPDR12[1:0]		PUPDR11[1:0]		PUPDR10[1:0]		PUPDR9[1:0]		PUPDR8[1:0]		PUPDR7[1:0]		PUPDR6[1:0]		PUPDR5[1:0]		PUPDR4[1:0]		PUPDR3[1:0]		PUPDR2[1:0]		PUPDR1[1:0]		PUPDR0[1:0]	
Mask	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1	1	1	1	1	1	
Value																					1	0			1	0	1	0	1	0	1	0

In this case, you need two masks: one to clear bits 11, 10, 7, 6, 5, 4, 3, 2, 1 and 0, and another to set bits 11, 7, 5, 3, and 1. It is also possible to clear only bits 10, 6, 4, 2 and 0.GPIOE Pull-up Pull-down Register MASK Value = **0x00000CFF** and **0x000008AA** (in HEX)**Alternative solution:****GPIOA->PUPDR &= ~(GPIO_PUPDR_PUPD0 | GPIO_PUPDR_PUPD1 | GPIO_PUPDR_PUPD2 | GPIO_PUPDR_PUPD3 | GPIO_PUPDR_PUPD5);****GPIOA->PUPDR |= (GPIO_PUPDR_PUPD0_1 | GPIO_PUPDR_PUPD1_1 | GPIO_PUPDR_PUPD2_1 | GPIO_PUPDR_PUPD3_1 | GPIO_PUPDR_PUPD5_1);**