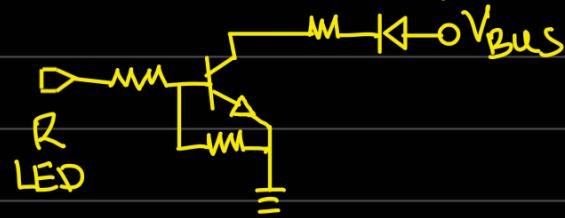


Connection of the onboard LED (TIVA Launchpad)

The 3 LEDs are connected to pins PF1, PF3, PF2
R G B

LED schematic



Same for the other 2.

The ends of RAM-ROM maps

For TIVA the ROM starts at 0x20000000 and ends at 0x20007FFF. The address is therefore 0x8000 bits wide \rightarrow 0x8000 \rightarrow 1000 0000 0000 0000 bit addressable locations. This is $2^{15} = 32,768$ or $\frac{32768}{1024} = 32\text{KB}$ size

There is a blank state just before 0x20000000 where the debugger can't reach —

Memory 1				
Go to	0x20000000	Memory		
0xffff'ff70	-----	-----	-----	
0xffff'ff80	-----	-----	-----	
0xffff'ff90	-----	-----	-----	
0xffff'ffa0	-----	-----	-----	
0xffff'ffb0	-----	-----	-----	
0xffff'ffc0	-----	-----	-----	
0xffff'ffd0	-----	-----	-----	
0xffff'ffe0	-----	-----	-----	
0xffff'fff0	-----	-----	-----	
0x2000'0000	00000000	20000115	00000000	00000000
0x2000'0010	00000000	00000000	00000000	00000000
0x2000'0020	4c1bb51c	68a26863	f1046921	90010018
0x2000'0030	90006960	f0006820	6060f82f	b510bd13

To blink LED, we need to know the map:

(i) Search memory map in datasheet

Table 2-4. Memory Map

Start	End	Description	For details, see page ...
Memory			
0x0000.0000	0x0003.FFFF	On-chip Flash	540
0x0004.0000	0x1FFF.FFFF	Reserved \rightarrow the blank state	-
0x2000.0000	0x2000.7FFF	Bit-banded on-chip SRAM	525
0x2000.8000	0x21FF.FFFF	Reserved	-
0x2200.0000	0x220F.FFFF	Bit-band alias of bit-banded on-chip SRAM starting at 0x2000.0000	525
0x2210.0000	0x3FFF.FFFF	Reserved	-

(ii) The RGB LED is in port F as per launchpad board manual on pins PF1, PF2, PF3. Look for port F in peripherals.

0x4002.4000	0x4002.4FFF	GPIO Port E	658
0x4002.5000	0x4002.5FFF	GPIO Port F	658
0x4002.6000	0x4002.7FFF		

(iii) Copy starting address to debugger. This is empty for now because the hardware is switched off. This is called clock gating. We need to turn the port F on.

(iv) Search "clock gating" in datasheet. From this we find the clock gating register for the GPIO (RCCGPIO). If we enable this we can access the GPIO registers.

(v) The RCCGPIO register has most bits reserved (RO or Read only). Bit 0-5 is changeable (R/W or read-write)

General-Purpose Input/Output Run Mode Clock Gating Control (RCCGPIO)															
Base 0x400F.E000 Offset 0x608 Type RW, reset 0x0000.0000															
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16															
reserved															
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0															
Type	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	R5	R4	R3	R2	R1
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit/Field Name Type Reset Description															
31:6	reserved	RO	0	Software should not rely on the value of a reserved bit. To provide compatibility with future products, the value of a reserved bit should be preserved across a read-modify-write operation.											
5	R5	RW	0	GPIO Port F Run Mode Clock Gating Control											
To enable port F hardware															
Value	Description														
0	GPIO Port F is disabled.														
1	Enable and provide a clock to GPIO Port F in Run mode.														
4	R4	RW	0	GPIO Port E Run Mode Clock Gating Control											
Value Description															
0	GPIO Port E is disabled.														
1	Enable and provide a clock to GPIO Port E in Run mode.														

(vi) Copy base address (0x400FE000) and add offset (0x608) to it. Go to symbolic window and search for this address to find RCCGPIO register. Set bit 5 to 1 to enable this. The data to be entered is : 0x0000 0020 (enables bit 5 only).

Memory 1				Symbolic Memory			
Go to	0x40025000	Memory	Location	Data	Variable	Value	
0x4002'4fa0	-----	-----	0x400f'e602	0x0			
0x4002'4fb0	-----	-----	0x400f'e604	0x0			
0x4002'4fc0	-----	-----	0x400f'e606	0x0			
0x4002'4fd0	-----	-----	0x400f'e608	0x0			
0x4002'4fe0	-----	-----	0x400f'e60a	0x0			
0x4002'4ff0	-----	-----	0x400f'e60c	0x0			
0x4002'5000	-----	-----	0x400f'e60e	0x0			
0x4002'5010	-----	-----	0x400f'e610	0x0			
0x4002'5020	-----	-----					
0x4002'5030	-----	-----					
0x4002'5040	-----	-----					
0x4002'5050	-----	-----					
0x4002'5060	-----	-----					

bit 5 not enabled

Memory 1				Symbolic Memory			
Go to	0x40025000	Memory	Location	Data	Variable	Value	
0x4002'4fc0	-----	-----	0x400f'e600	0x0			
0x4002'4fd0	-----	-----	0x400f'e604	0x0			
0x4002'4fe0	-----	-----	0x400f'e608	0x20			
0x4002'4ff0	-----	-----	0x400f'e60c	0x0			
0x4002'5000	00000000 00000000 00000000 00000000	00000000 00000000 00000000 00000000	0x400f'e60e	0x0			
0x4002'5010	00000000 00000000 00000000 00000000	00000000 00000000 00000000 00000000	0x400f'e610	0x0			
0x4002'5020	00000000 00000000 00000000 00000000	00000000 00000000 00000000 00000000					
0x4002'5030	00000000 00000000 00000000 00000000	00000000 00000000 00000000 00000000					
0x4002'5040	00000000 00000000 00000000 00000000	00000000 00000000 00000000 00000000					
0x4002'5050	00000000 00000000 00000000 00000000	00000000 00000000 00000000 00000000					
0x4002'5060	00000000 00000000 00000000 00000000	00000000 00000000 00000000 00000000					

bit 5 enabled

(vii) Need to configure GPIO pins PF1, PF2, PF3 as output bits. Therefore set bits 1,2,3 to 1 (1110) at the address 0x4D0U25400 at lowest nibble. This is the GPIO pin direction register (port F). Setting the bits of this register configures the corresponding pins as outputs. Clearing these bits sets the corresponding pins as input

(viii) Do the same at address 0x4002551C as before and set pins/bits 1,2,3 of the register to 1 (111D) at lowest nibble.

This enables the GPIO port F to be used as a digital output. The register at this address is the GPIODEN (port F) register.

(ix) Now at 0x40025038 write 0x02(0010) to the lowest nibble (this sets pin PF1 to 1). To get green led write 0x08 to lowest nibble (this sets pin PF3 to 1). To get blue LED write 0x04 (0100) to set pin PF2. 1110 (0xE) for all 3 LED's (white).

Explanation of address 0x40025038 for GPIO0DATA

The data register is virtually mapped to 256 locations. The data register supports bit specific addressing. This allows bits 9 to 2 of the CRPIODATA register to be set/reset to allow data to be read/write to the corresponding pins. For example:

CPIO DATA - port F is at 0x40025000.

To write data to it the bits 9:2 of the address must be masked so the bits 9:2 are 0x00.

Since PF1, PF2, PF3 are to be written to, we set the bits 9:2 of the address, corresponding to the 3 pins of 0000 0000 0000 → bits 9:2

`000000111000` → bits
= `0x38` PF3 PF2 PF1 corresponding to pins set

so this makes our new APIODATA address $0x4002\ 5000 + 0x038$

This is the address we then write LED data to toggle on red|green|blue LED's.

* * * End.

10.2.1.2 Data Register Operation

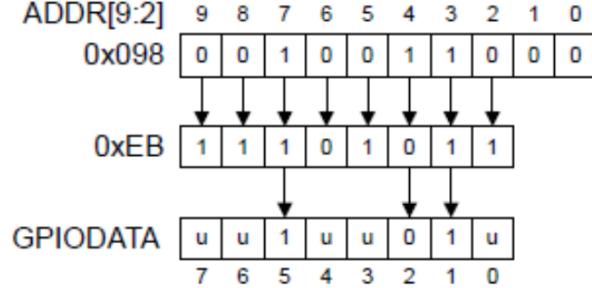
*** official explanation

To aid in the efficiency of software, the GPIO ports allow for the modification of individual bits in the **GPIO Data (GPIODATA)** register (see page 662) by using bits [9:2] of the address bus as a mask. In this manner, software drivers can modify individual GPIO pins in a single instruction without affecting the state of the other pins. This method is more efficient than the conventional method of performing a read-modify-write operation to set or clear an individual GPIO pin. To implement this feature, the **GPIODATA** register covers 256 locations in the memory map.

During a write, if the address bit associated with that data bit is set, the value of the **GPIODATA** register is altered. If the address bit is cleared, the data bit is left unchanged.

For example, writing a value of 0xEB to the address **GPIODATA** + 0x098 has the results shown in Figure 10-3, where **u** indicates that data is unchanged by the write. This example demonstrates how **GPIODATA** bits 5, 2, and 1 are written.

Figure 10-3. GPIO DATA Write Example



During a read, if the address bit associated with the data bit is set, the value is read. If the address bit associated with the data bit is cleared, the data bit is read as a zero, regardless of its actual value. For example, reading address **GPIO DATA** + 0x0C4 yields as shown in Figure 10-4. This example shows how to read **GPIO DATA** bits 5, 4, and 0.

Figure 10-4. GPIODATA Read Example

