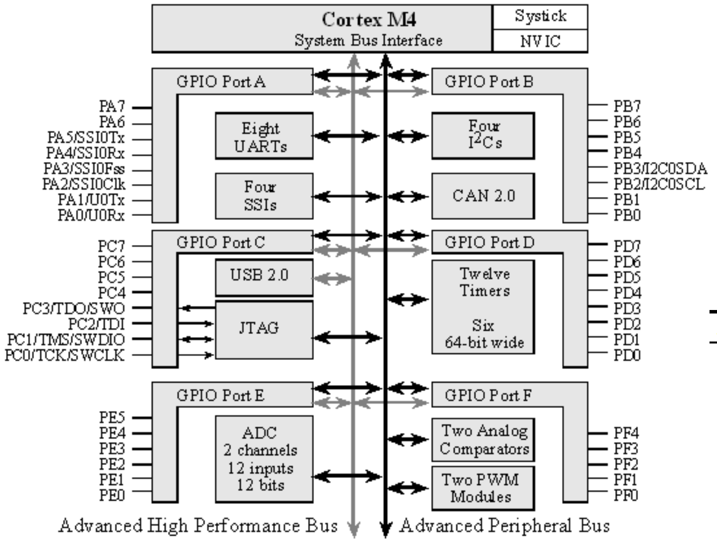
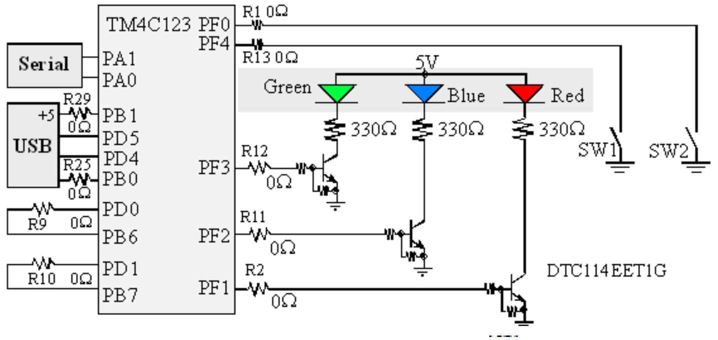
EET3350 Embedded Systems Design – Reference Sheet





If we wish to access bit	*Constant*
7	0x0200
6	0x0100
5	0x0080
4	0x0040
3	0x0020
2	0x0010
1	0x0008
0	0x0004

Port	Base address
PortA	0x40004000
PortB	0x40005000
PortC	0x40006000
PortD	0x40007000
PortE	0x40024000
PortF	0x40025000

IO	Ain	0	1	2	3	4	5	6	7	8	9	14
PA0		*Port*	*U0Rx*							CAN1Rx		
PA1		*Port*	*U0Tx*							CAN1Tx		
PA2		Port		SSI0Clk								
PA3		Port		SSI0Fss								
PA4		Port		SSI0Rx								
PA5		Port		SSI0Tx								
PA6		Port			I_2C1SCL		M1PWM2					
PA7		Port			I_2C1SDA		M1PWM3					
PB0		Port	U1Rx						T2CCP0			
PB1		Port	U1Tx						T2CCP1			
PB2		Port			I_2C0SCL				T3CCP0			
PB3		Port			I_2C0SDA				T3CCP1			
PB4	Ain10	Port		SSI2Clk		M0PWM2			T1CCP0	CAN0Rx		
PB5	Ain11	Port		SSI2Fss		M0PWM3			T1CCP1	CAN0Tx		
PB6		Port		SSI2Rx		M0PWM0			T0CCP0			
PB7		Port		SSI2Tx		M0PWM1			T0CCP1			
PC4	C1-	Port	U4Rx	U1Rx		M0PWM6		IDX1	WT0CCP0	U1RTS		
PC5	C1+	Port	U4Tx	U1Tx		M0PWM7		PhA1	WT0CCP1	U1CTS		
PC6	C0+	Port	U3Rx					PhB1	WT1CCP0	USB0epen		
PC7	C0-	Port	U3Tx						WT1CCP1	USB0pflt		
PD0	Ain7	Port	SSI3Clk	SSI1Clk	I_2C3SCL	M0PWM6	M1PWM0		WT2CCP0			
PD1	Ain6	Port	SSI3Fss	SSI1Fss	I_2C3SDA	M0PWM7	M1PWM1		WT2CCP1			
PD2	Ain5	Port	SSI3Rx	SSI1Rx		M0Fault0			WT3CCP0	USB0epen		
PD3	Ain4	Port	SSI3Tx	SSI1Tx				IDX0	WT3CCP1	USB0pflt		
PD4	*USB0DM*	*Port*	*U6Rx*						*WT4CCP0*			
PD5	*USB0DP*	*Port*	*U6Tx*						*WT4CCP1*			
PD6		Port	U2Rx			M0Fault0		PhA0	WT5CCP0			
PD7		Port	U2Tx					PhB0	WT5CCP1	NMI		
PE0	Ain3	Port	U7Rx									
PE1	Ain2	Port	U7Tx									
PE2	Ain1	Port										
PE3	Ain0	Port										
PE4	Ain9	Port	U5Rx		I_2C2SCL	M0PWM4	M1PWM2			CAN0Rx		
PE5	Ain8	Port	U5Tx		I_2C2SDA	M0PWM5	M1PWM3			CAN0Tx		
PF0		Port	U1RTS	SSI1Rx	CAN0Rx		M1PWM4	PhA0	T0CCP0	NMI	C0o	
PF1		Port	U1CTS	SSI1Tx			M1PWM5	PhB0	T0CCP1		C1o	TRD1
PF2		Port		SSI1Clk		M0Fault0	M1PWM6		T1CCP0			TRD0
PF3		Port		SSI1Fss	CAN0Tx		M1PWM7		T1CCP1			TRCLK
PF4		Port					M1Fault0	IDX0	T2CCP0	USB0epen		

LDR Rd, [Rn]; load 32-bit number at [Rn] to Rd
LDR Rd, [Rn,#off]; load 32-bit number at [Rn+off] to Rd
LDR Rd, =value; set Rd equal to any 32-bit value (PC rel)
STR Rt, [Rn]; store 32-bit Rt to [Rn]
STR Rt, [Rn,#off] ; store 32-bit Rt to [Rn+off]

PUSH {Rt}; push 32-bit Rt onto stack
POP {Rd} ; pop 32-bit number from stack into Rd
ADR Rd, label ; set Rd equal to the address at label
MOV{S} Rd,< op2> ; set Rd equal to op2
MOV Rd, #im16; set Rd equal to im16, im16 is 0 to 65535

B label ; branch to label Always
BEQ label ; branch if Z == 1 Equal
BNE label ; branch if Z == 0 Not equal
BX Rm ; branch indirect to location specified by Rm
BL label ; branch to subroutine at label
BLX Rm ; branch to subroutine indirect specified by Rm

AND{S} {Rd,} Rn, <op2> ; Rd=Rn&op2 (op2 is 32 bits)
ORR{S} {Rd,} Rn, <op2> ; Rd=Rn|op2 (op2 is 32 bits)
EOR{S} {Rd,} Rn, <op2> ; Rd=Rn^op2 (op2 is 32 bits)
BIC{S} {Rd,} Rn, <op2> ; Rd=Rn&(~op2) (op2 is 32 bits)
ORN{S} {Rd,} Rn, <op2> ; Rd=Rn|(~op2) (op2 is 32 bits)
LSR{S} Rd, Rm, Rs ; logical shift right Rd=Rm>>Rs (unsignd)
LSR{S} Rd, Rm, #n ; logical shift right Rd=Rm>>n (unsigned)
ASR{S} Rd, Rm, Rs ; arithmetic shift right Rd=Rm>>Rs (sign)
ASR{S} Rd, Rm, #n ; arithmetic shift right Rd=Rm>>n (signed)
LSL{S} Rd, Rm, Rs ; shift left Rd=Rm<<Rs (signed, unsign)
LSL{S} Rd, Rm, #n ; shift left Rd=Rm<<n (signed, unsigned)

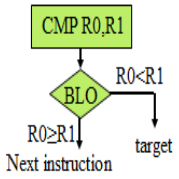
ADD{S} {Rd,} Rn, <op2> ; Rd = Rn + op2
ADD{S} {Rd,} Rn, #im12 ; Rd = Rn + im12, im12 is 0 to 4095
SUB{S} {Rd,} Rn, <op2> ; Rd = Rn - op2
SUB{S} {Rd,} Rn, #im12 ; Rd = Rn - im12, im12 is 0 to 4095
RSB{S} {Rd,} Rn, <op2> ; Rd = op2 – Rn
RSB{S} {Rd,} Rn, #im12 ; Rd = im12 Rn
CMP Rn, <op2> ; Rn op2 sets the NZVC bits
CMN Rn, <op2> ; Rn - (-op2) sets the NZVC bits
MUL{S} {Rd,} Rn, Rm ; Rd = Rn * Rm
UDIV{Rd,} Rn, Rm ; Rd = Rn/Rm   unsigned

Page 1, **REFERENCES:** “Embedded Systems - Shape the World” by Jonathan Valvano and Ramesh Yerraballi is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License. Based on a work at http://users.ece.utexas.edu/~valvano/arm/outline1.htm.

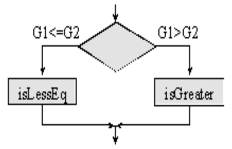
❑Unsigned conditional branch

❖follow SUBS CMN or CMP

```
BLO target    ; Branch if unsigned less than (if C=0, same as BCC)
BLS target    ; Branch if unsigned less than or equal to (if C=0 or Z=1)
BHS target    ; Branch if unsigned greater than or equal to
                (if C=1, same as BCS)
BHI target    ; Branch if unsigned greater than (if C=1 and Z=0)
```



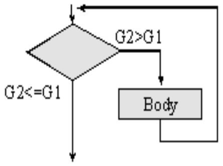
If-then-else



```
     LDR R2, =G1     ; R2 = &G1
     LDR R0, [R2]    ; R0 = G1
     LDR R2, =G2     ; R2 = &G2
     LDR R1, [R2]    ; R1 = G2
     CMP R0, R1      ; is G1 > G2 ?
     BHI high        ; if so, skip to high
low  BL  isLessEq    ; G1 <= G2
     B   next        ; unconditional
high BL  isGreater   ; G1 > G2
next
```

```
uint32_t G1,G2;
if(G1>G2){
  isGreater();
}
else{
  isLessEq();
}
```

While Loops



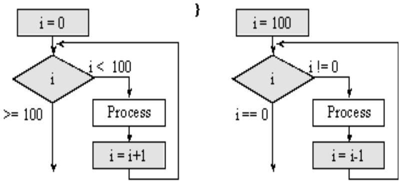
```
     LDR R4, =G1     ; R4 -> G1
     LDR R5, =G2     ; R5 -> G2
loop LDR R0, [R5]    ; R0 =  G2
     LDR R1, [R4]    ; R1 =  G1
     CMP R0, R1      ; is G2 <= G1?
     BLS next        ; if so, skip to next
     BL  Body        ; body of the loop
     B   loop
next
```

```
uint32_t G1,G2;
while(G2 > G1){
  Body();
}
```

For Loops

```
for(i=0; i<100; i++){
  Process();
}
```

```
for(i=100; i!=0; i--){
  Process();
}
```



Address	7	6	5	4	3	2	1	0	Name
$400F.E108	--	--	GPIOF	GPIOE	GPIOD	GPIOC	GPIOB	GPIOA	SYSCTL_RCGC2_R
$4000.43FC	DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA	GPIO_PORTA_DATA_R
$4000.4400	DIR	DIR	DIR	DIR	DIR	DIR	DIR	DIR	GPIO_PORTA_DIR_R
$4000.4420	SEL	SEL	SEL	SEL	SEL	SEL	SEL	SEL	GPIO_PORTA_AFSEL_R
$4000.4510	PUE	PUE	PUE	PUE	PUE	PUE	PUE	PUE	GPIO_PORTA_PUR_R
$4000.451C	DEN	DEN	DEN	DEN	DEN	DEN	DEN	DEN	GPIO_PORTA_DEN_R
$4000.4524	1	1	1	1	1	1	1	1	GPIO_PORTA_CR_R
$4000.4528	0	0	0	0	0	0	0	0	GPIO_PORTA_AMSEL_R
$4000.53FC	DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA	GPIO_PORTB_DATA_R
$4000.5400	DIR	DIR	DIR	DIR	DIR	DIR	DIR	DIR	GPIO_PORTB_DIR_R
$4000.5420	SEL	SEL	SEL	SEL	SEL	SEL	SEL	SEL	GPIO_PORTB_AFSEL_R
$4000.5510	PUE	PUE	PUE	PUE	PUE	PUE	PUE	PUE	GPIO_PORTB_PUR_R
$4000.551C	DEN	DEN	DEN	DEN	DEN	DEN	DEN	DEN	GPIO_PORTB_DEN_R
$4000.5524	1	1	1	1	1	1	1	1	GPIO_PORTB_CR_R
$4000.5528	0	0	AMSEL	AMSEL	0	0	0	0	GPIO_PORTB_AMSEL_R
$4000.63FC	DATA	DATA	DATA	DATA	JTAG	JTAG	JTAG	JTAG	GPIO_PORTC_DATA_R
$4000.6400	DIR	DIR	DIR	DIR	JTAG	JTAG	JTAG	JTAG	GPIO_PORTC_DIR_R
$4000.6420	SEL	SEL	SEL	SEL	JTAG	JTAG	JTAG	JTAG	GPIO_PORTC_AFSEL_R
$4000.6510	PUE	PUE	PUE	PUE	JTAG	JTAG	JTAG	JTAG	GPIO_PORTC_PUR_R
$4000.651C	DEN	DEN	DEN	DEN	JTAG	JTAG	JTAG	JTAG	GPIO_PORTC_DEN_R
$4000.6524	1	1	1	1	JTAG	JTAG	JTAG	JTAG	GPIO_PORTC_CR_R
$4000.6528	AMSEL	AMSEL	AMSEL	AMSEL	JTAG	JTAG	JTAG	JTAG	GPIO_PORTC_AMSEL_R
$4000.73FC	DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA	GPIO_PORTD_DATA_R
$4000.7400	DIR	DIR	DIR	DIR	DIR	DIR	DIR	DIR	GPIO_PORTD_DIR_R
$4000.7420	SEL	SEL	SEL	SEL	SEL	SEL	SEL	SEL	GPIO_PORTD_AFSEL_R
$4000.7510	PUE	PUE	PUE	PUE	PUE	PUE	PUE	PUE	GPIO_PORTD_PUR_R
$4000.751C	DEN	DEN	DEN	DEN	DEN	DEN	DEN	DEN	GPIO_PORTD_DEN_R
$4000.7524	CR	1	1	1	1	1	1	1	GPIO_PORTD_CR_R
$4000.7528	0	0	AMSEL	AMSEL	AMSEL	AMSEL	AMSEL	AMSEL	GPIO_PORTD_AMSEL_R
$4002.43FC			DATA	DATA	DATA	DATA	DATA	DATA	GPIO_PORTE_DATA_R
$4002.4400			DIR	DIR	DIR	DIR	DIR	DIR	GPIO_PORTE_DIR_R
$4002.4420			SEL	SEL	SEL	SEL	SEL	SEL	GPIO_PORTE_AFSEL_R
$4002.4510			PUE	PUE	PUE	PUE	PUE	PUE	GPIO_PORTE_PUR_R
$4002.451C			DEN	DEN	DEN	DEN	DEN	DEN	GPIO_PORTE_DEN_R
$4002.4524			1	1	1	1	1	1	GPIO_PORTE_CR_R
$4002.4528			AMSEL	AMSEL	AMSEL	AMSEL	AMSEL	AMSEL	GPIO_PORTE_AMSEL_R
$4002.53FC				DATA	DATA	DATA	DATA	DATA	GPIO_PORTF_DATA_R
$4002.5400				DIR	DIR	DIR	DIR	DIR	GPIO_PORTF_DIR_R
$4002.5420				SEL	SEL	SEL	SEL	SEL	GPIO_PORTF_AFSEL_R
$4002.5510				PUE	PUE	PUE	PUE	PUE	GPIO_PORTF_PUR_R
$4002.551C				DEN	DEN	DEN	DEN	DEN	GPIO_PORTF_DEN_R
$4002.5524				1	1	1	1	CR	GPIO_PORTF_CR_R
$4002.5528				0	0	0	0	0	GPIO_PORTF_AMSEL_R

	31-28	27-24	23-20	19-16	15-12	11-8	7-4	3-0	
$4000.452C	PMC7	PMC6	PMC5	PMC4	PMC3	PMC2	PMC1	PMC0	GPIO_PORTA_PCTL_R
$4000.552C	PMC7	PMC6	PMC5	PMC4	PMC3	PMC2	PMC1	PMC0	GPIO_PORTB_PCTL_R
$4000.652C	PMC7	PMC6	PMC5	PMC4	0x1	0x1	0x1	0x1	GPIO_PORTC_PCTL_R
$4000.752C	PMC7	PMC6	PMC5	PMC4	PMC3	PMC2	PMC1	PMC0	GPIO_PORTD_PCTL_R
$4002.452C			PMC5	PMC4	PMC3	PMC2	PMC1	PMC0	GPIO_PORTE_PCTL_R
$4002.552C				PMC4	PMC3	PMC2	PMC1	PMC0	GPIO_PORTF_PCTL_R
$4000.6520	LOCK (write 0x4C4F434B to unlock, other locks) (reads 1 if locked, 0 if unlocked)								GPIO_PORTC_LOCK_R
$4000.7520	LOCK (write 0x4C4F434B to unlock, other locks) (reads 1 if locked, 0 if unlocked)								GPIO_PORTD_LOCK_R
$4002.5520	LOCK (write 0x4C4F434B to unlock, other locks) (reads 1 if locked, 0 if unlocked)								GPIO_PORTF_LOCK_R

```
PE0 EQU 0x4005C004                  ;bit-specific address Port E bit 0
LED_Init
    LDR R1, =SYSCTL_RCGCGPIO_R      ; R1 -> SYSCTL_RCGCGPIO_R
    LDR R0, [R1]                    ; previous value
    ORR R0, R0, #0x00000010         ; activate clock for Port E
    STR R0, [R1]
    NOP
    NOP                             ; allow time to finish activating
    LDR R1, =GPIO_PORTE_DIR_R       ; R1 -> GPIO_PORTE_DIR_R
    LDR R0, [R1]                    ; previous value
    ORR R0, R0, #0x01               ; PE0 output
    STR R0, [R1]                    ; set direction register
    LDR R1, =GPIO_PORTE_AFSEL_R     ; R1 -> GPIO_PORTE_AFSEL_R
    LDR R0, [R1]                    ; previous value
    BIC R0, R0, #0x01               ; disable alt funct
    STR R0, [R1]                    ; set alternate function register
    LDR R1, =GPIO_PORTE_DEN_R       ; R1 -> GPIO_PORTE_DEN_R
    LDR R0, [R1]                    ; previous value
    ORR R0, R0, #0x01               ; enable PE0 digital port
    STR R0, [R1]                    ; set digital enable register
    BX  LR
```

```
LED_Off
    LDR R1, =PE0                    ; R1 is 0x4005C004
    MOV R0, #0
    STR R0, [R1]                    ; affect just PE0
    BX  LR
LED_On
    LDR R1, =PE0                    ; R1 is 0x4005C004
    MOV R0, #1
    STR R0, [R1]                    ; affect just PE0
    BX  LR
LED_Toggle
    LDR R1, =PE0                    ; R1 is 0x4005C004
    LDR R0, [R1]                    ; previous value
    EOR R0, R0, #1                  ; flip bit 0
    STR R0, [R1]                    ; affect just PE0
    BX  LR
```

REFERENCES:

HLMP-4700, HLMP-4719, HLMP-4740 HLMP-1700, HLMP-1719, HLMP-1790

T-1$^3/_4$ (5 mm), T-1 (3 mm), Low Current LED Lamps

Avago TECHNOLOGIES

Data Sheet

Package Description	Color	Device HLMP-	Luminous Intensity Iv (mcd) at 2 mA Min.	Typ.	Max.	2θ$^{1/2}$	Package Outline
T-1 3/4 Tinted Diffused	Red	4700	1.5	2.3	–	50	A
		4700-C00xx	1.5	2.3	–		
		4700-CD0FH	1.5	2.3	4.2		
	Yellow	4719	0.9	2.1	–		
		4719-A00xx	0.9	2.1	–		
	Green	4740	1.0	2.3	–		
		4740-A00xx	1.0	2.3	–		
		4740-AB000	1.0	2.3	3.2		

Electrical/Optical Characteristics at $T_A = 25°C$

Symbol	Description	T-1$^3/_4$	T-1	Min.	Typ.	Max.	Units	Test Conditions
V_F	Forward Voltage	4700	1700		1.7	2.0	V	2 mA
		4719	1719		1.8	2.5		
		4740	1790		1.9	2.2		
V_R	Reverse Breakdown Voltage	4700	1700	5.0			V	$I_R = 50\ \mu A$
		4719	1719	5.0				
		4740	1790	5.0				

Absolute Maximum Ratings

Parameter		Maximum Rating	Units
Power Dissipation (Derate linearly from 92°C at 1.0 mA/°C)	Red Yellow Green	14 17.5 15.4	mW
DC and Peak Forward Current		7	mA
Transient Forward Current (10 µs Pulse)[1]		500	mA
Reverse Voltage (I_R = 50 µA)		5.0	V
Operating Temperature Range	Red/Yellow Green	-40 to 100 -20 to 100	°C °C

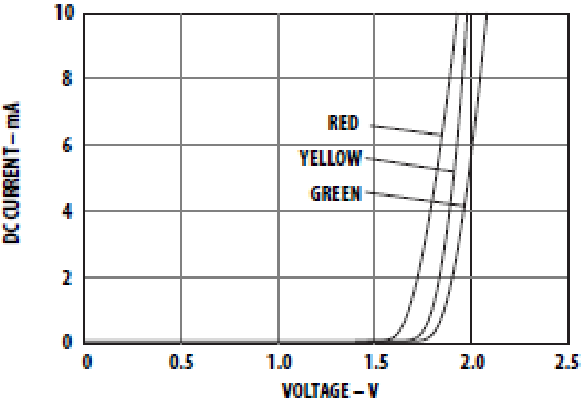


Figure 2. Forward current vs. forward voltage.

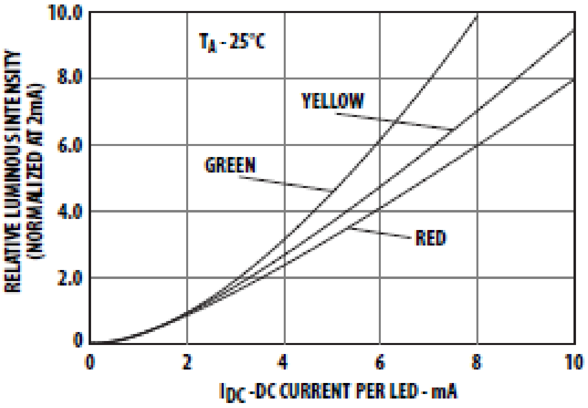


Figure 3. Relative luminous intensity vs. forward current.