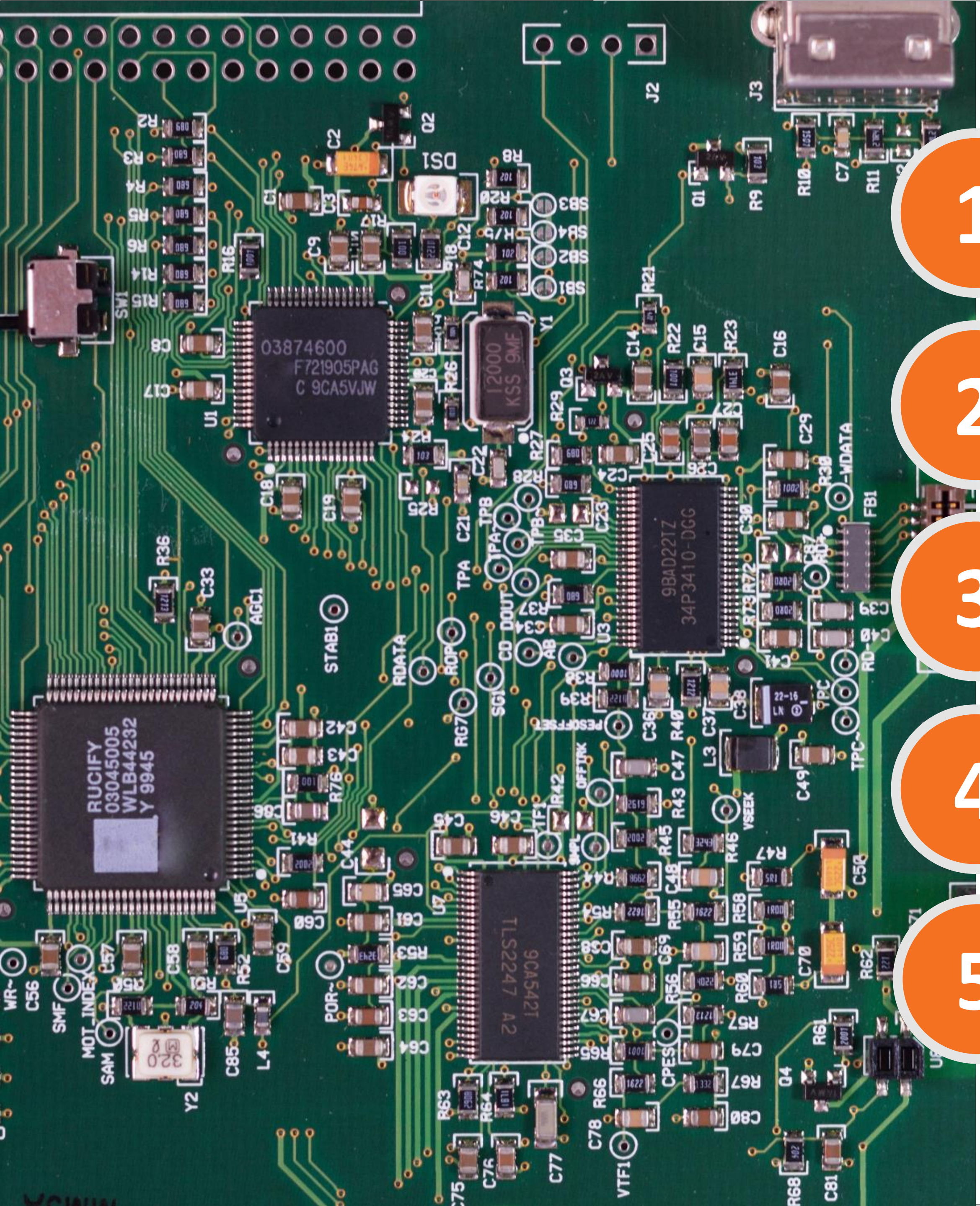
Lab Session 06

GPIO Interfacing
External LED Interfacing

MCT-236L: Embedded Systems-I Lab



1. PROBLEM STATEMENT
2. GPIO PIN SELECTION
3. GPIO CONFIGURATION STEPS
4. SOFTWARE PROGRAM
5. HARDWARE CONNECTIONS

PROBLEM STATEMENT

“Interface an external LED with TIVA C Series LaunchPad and program it to flash at a frequency of 1 Hz, continuously.”

GPIO PIN SELECTION

ARM Cortex-M4 Microcontroller

TM4C123 MICROCONTROLLER GPIOS

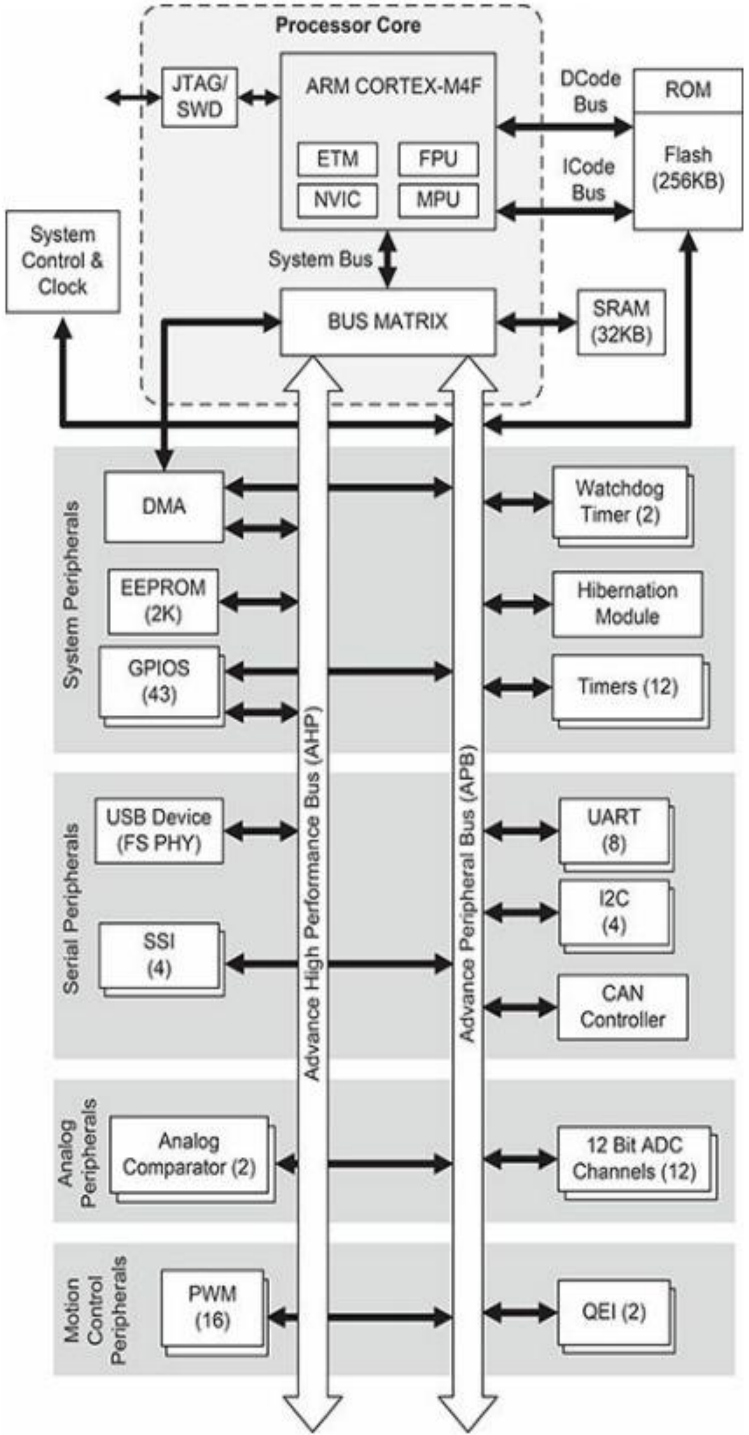
GPIO PIN SELECTION

- Out of 64-pins of TM4C123GH6PM, 43 pins are GPIO
- Grouped in six labeled **PortA** to **PortF**
  - **PortA** to **PortD** are 8-pins ports, **PortE** is 6-pins, and **PortF** is 5-pins port
- Some of the port pins also have special peripheral functionalities multiplexed
- When configured as GPIO these port pins have the following capabilities
  - Internal weak pull-up or pull-down resistors
  - Each port pin can be configured as open drain
  - Slew rate control capability is provided for 8 mA output drive
  - Some of the port pins are capable of tolerating 5V when configured as inputs
  - Configurable current sourcing capability for levels of 2 mA, 4 mA, and 8 mA

TM4C123 MICROCONTROLLER GPIOS

GPIO PIN SELECTION

- ARM Cortex-M4 architecture uses memory-based peripherals
- Two on-chip busses that connect the processor core to the peripherals
  - **Advanced Peripheral Bus** (**APB**) is a low-speed legacy bus
  - **Advanced High-performance Bus** (**AHB**) is a high-speed bus
- GPIO module has connectivity available on both the buses



TM4C123 MICROCONTROLLER GPIOS

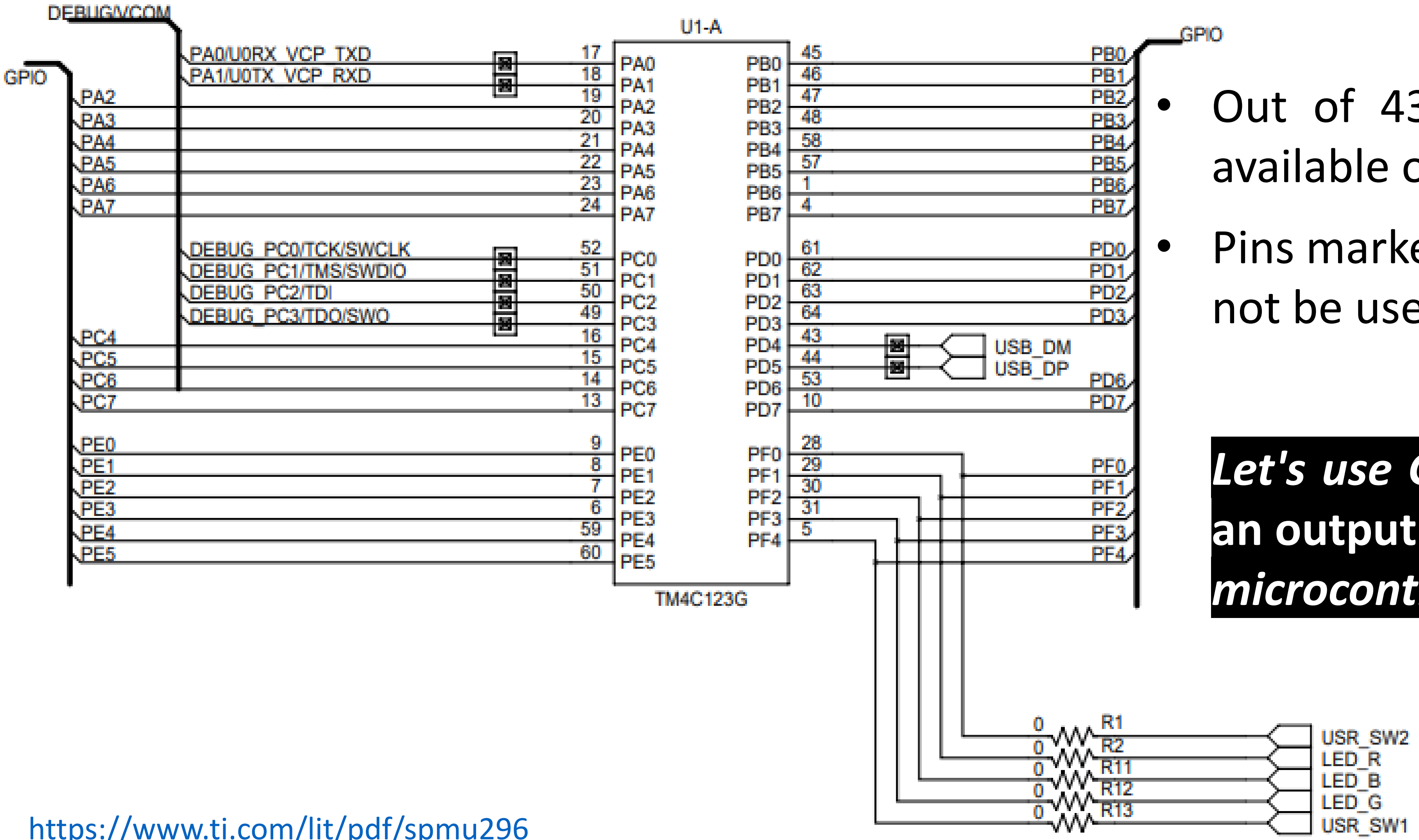
GPIO PIN SELECTION

- Start address is also known as **Base Address** of GPIO Port
- Address space allocated to GPIO is allocated to different configurations, control, status, and data registers
- Each configuration register has unique **Offset** for every GPIO port, and it can be accessed by adding this offset to base address
- All registers associated with each GPIO Port are accessible by both buses but with different addresses

Bus	Start Address	End Address	Description
Advanced Peripheral Bus (**APB**)	0x4000 4000	0x4000 4FFF	GPIO Port A Registers
	0x4000 5000	0x4000 5FFF	GPIO Port B Registers
	0x4000 6000	0x4000 6FFF	GPIO Port C Registers
	0x4000 7000	0x4000 7FFF	GPIO Port D Registers
	0x4002 4000	0x4002 4FFF	GPIO Port E Registers
	0x4002 5000	0x4002 5FFF	GPIO Port F Registers
Advanced High-Performance Bus (**AHB**)	0x4005 8000	0x4005 8FFF	GPIO Port A Registers
	0x4005 9000	0x4005 9FFF	GPIO Port B Registers
	0x4005 A000	0x4005 AFFF	GPIO Port C Registers
	0x4005 B000	0x4005 BFFF	GPIO Port D Registers
	0x4005 C000	0x4005 CFFF	GPIO Port E Registers
	0x4005 D000	0x4005 DFFF	GPIO Port F Registers

TIVA C Series LaunchPad

GPIO PIN SELECTION



- Out of 43, only 35 GPIO pins are available on TIVA C Series LaunchPad
- Pins marked with a crossed box must not be used for GPIO interfacing

Let's use GPIO PortE Pin 1 (PE1), **as an output pin,** ***for interfacing LED to microcontroller***

https://www.ti.com/lit/pdf/spmu296

GPIO CONFIGURATION STEPS

LED Interfacing at GPIO Port E Pin 1

LED Interfacing (PE1 as Digital Output)

GPIO CONFIGURATION STEPS

Step 1: Clock Enable

- Enable clock for GPIO Port whether entire port or few pins are to be configured as GPIO
- **RCGC_GPIO_R** register, mapped to memory address **0x400FE608**, is used enable clock for GPIO
- Bit 0 to 5 of **RCGC_GPIO_R** can be set to enable clock to **PortA** to **PortF**, respectively
- For enabling clock for **PortE**, only bit 4 need to be set while all other bits are zero

Clock enable for	Port F	Port E	Port D	Port C	Port B	Port A	8-bit Hexadecimal Value
For clock enable set	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
Bit values	0	**1**	0	0	0	0	**0x10**

LED Interfacing (PE1 as Digital Output)

GPIO CONFIGURATION STEPS

Step 2: GPIO Bus Selection

- Bus configuration is selected (either **APB** or **AHB**) by selecting appropriate base address of the GPIO Port
- For selecting PortE on APB, 0x40024000 is selected as Base Address for all remaining configuration registers

Bus	Start Address	Description
Advanced Peripheral Bus (**APB**)	0x4000 4000	GPIO Port A Registers
	0x4000 5000	GPIO Port B Registers
	0x4000 6000	GPIO Port C Registers
	0x4000 7000	GPIO Port D Registers
	0x4002 4000	**GPIO Port E Registers**
	0x4002 5000	GPIO Port F Registers
Advanced High-Performance Bus (**AHB**)	0x4005 8000	GPIO Port A Registers
	0x4005 9000	GPIO Port B Registers
	0x4005 A000	GPIO Port C Registers
	0x4005 B000	GPIO Port D Registers
	0x4005 C000	GPIO Port E Registers
	0x4005 D000	GPIO Port F Registers

LED Interfacing (PE1 as Digital Output)

GPIO CONFIGURATION STEPS

Step 3-5: Configurations

- PortE on APB (Base Address = 0x40024000)
- Mask for Pin 1 = 0000 0010 = 0x02

Step	Configuration	Register	Offset	Register Address (Base Address + Offset)	Configuration Setting	Register Value
3	Mode Control Config.	**GPIO_AFSEL_R**	0x420	0x40024420	Clear bit 1 to use as GPIO pin	0x00
4	Pad Control Config.	**GPIO_DEN_R**	0x51C	0x4002451C	Set bit 1 to digital enable GPIO pin	0x02
5	Data Control Config.	**GPIO_DIR_R**	0x400	0x40024400	Set bit 1 to configure as pin as output	0x02
		GPIO_DATA_R	0x000-0x3FC	0x40024008	Set bit 1 to 1 to read/write GPIO pin (register offset is 0x008 i.e., 0000 0000 1000)	

LED Interfacing (PE1 as Digital Output)

GPIO CONFIGURATION STEPS

Step 1-5: Summary

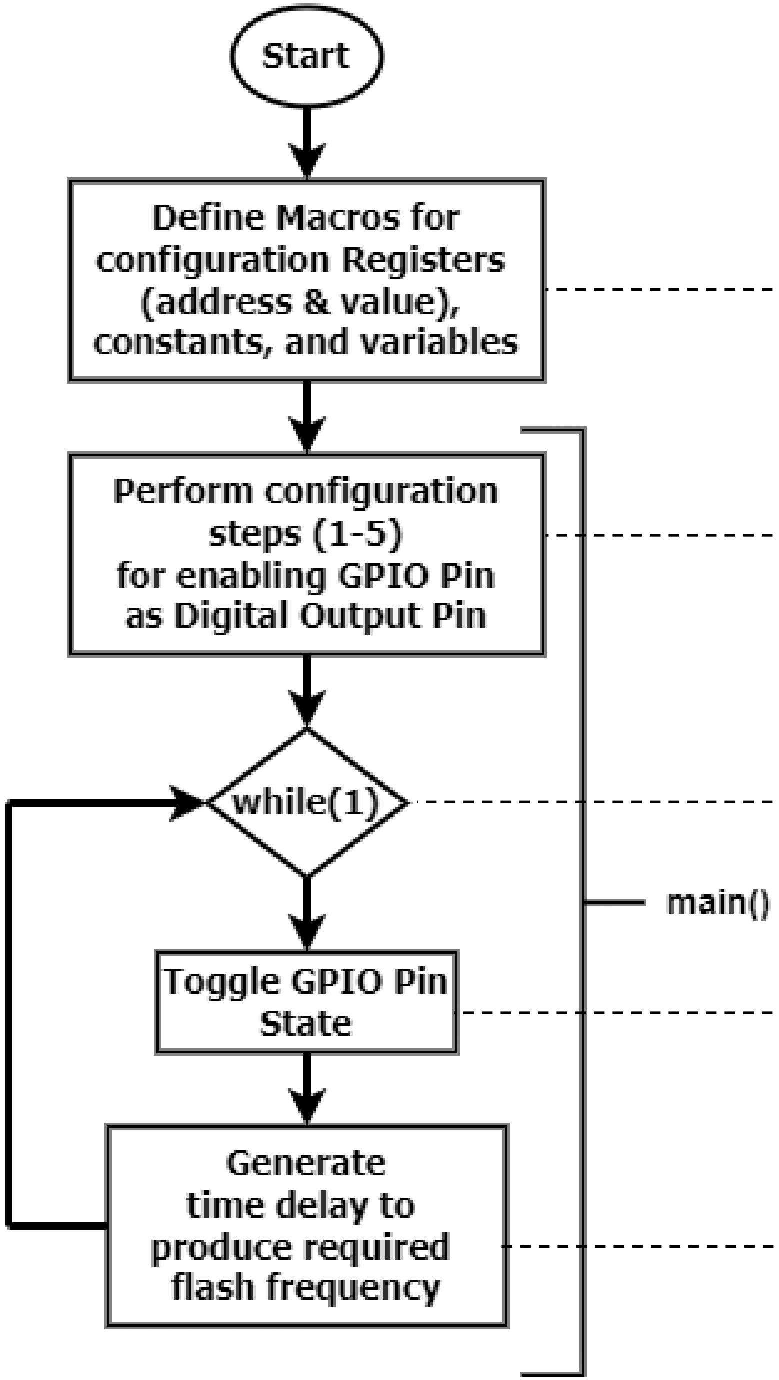
Step	Configuration	Register	Register Address (Base Address + Offset)	Configuration Setting	Register Value
1	Clock Enable	**RCGC_GPIO_R**	0x400FE608	Enable clock on GPIO PortE	0x10
2	Bus Selection	**-**	-	APB is selected for GPIO PortE	-
3	Mode Control Config.	**GPIO_AFSEL_R**	0x40024420	Clear bit 1 to use as GPIO pin	0x00
4	Pad Control Config.	**GPIO_DEN_R**	0x4002451C	Set bit 1 to digital enable GPIO pin	0x02
5	Data Control Config.	**GPIO_DIR_R**	0x40024400	Set bit 1 to configure pin as output	0x02
		GPIO_DATA_R	0x40024008	Set bit 1 to 1 to read/write GPIO pin (register offset is 0x008 i.e., 0000 0000 1000)	

SOFTWARE PROGRAM

LED Interfacing at GPIO Port E Pin 1

Program Flowchart

SOFTWARE PROGRAM


Start
Define Macros for configuration Registers (address & value), constants, and variables
Perform configuration steps (1-5) for enabling GPIO Pin as Digital Output Pin
while(1)
Toggle GPIO Pin State
Generate time delay to produce required flash frequency
main()

Define register addresses as ***pointer*** of type ***volatile unsigned long***

Use bitwise logical operators to assign values to configuration registers.

Mandatory infinite loop

Mandatory C language function ***main()***

Use bitwise XOR operation to toggle value of data configuration register.

Use ***for*** (or any other) loop for wasting microcontroller time to generate delay

Bit Masking & Bitwise Operators

SOFTWARE PROGRAM

- In computer programming, we can’t perform any operation directly on individual bit.
- Byte (8-bits) is the minimum size of data that can be accessed and manipulated by different types of operators (arithmetic, logical, etc.)
- **Bit Masking** is a technique used for accessing individual bits by using bitwise operators
- Using a **mask**, individual bit or multiple bits in a byte, nibble, word, etc. can be set either on or off, or inverted from on to off (or vice versa) in a single bitwise operation

Bit Masking & Bitwise Operators

SOFTWARE PROGRAM

Bitwise AND Operation: Turning OFF Bits

Use bitwise logical operator AND for turning OFF the bits

The 1 bit in the **mask** select which bits we want to remain unchanged, and zero bits in the **mask** selects the bits to turn OFF

For example,

		Binary Value							
	Hex Value	b7	b6	b5	b4	b3	b2	b1	b0
value	0xF6	1	1	1	1	0	1	1	0
mask	0xCF	1	1	0	0	1	1	1	1
result	0xE6	1	1	**0**	**0**	0	1	1	0

(*result* = *value*&***mask***)

Bitwise AND operation for turning OFF **bit4** & **bit5**

Bitwise OR Operation: Turning ON Bits

Use bitwise logical operator OR for turning ON the bits

The 1 bit in the **mask** select which bits we want to turn ON, and zero bits in the **mask** leaves the bits unchanged

For example,

		Binary Value							
	Hex Value	b7	b6	b5	b4	b3	b2	b1	b0
value	0x86	1	0	0	0	0	1	1	0
mask	0x40	0	1	0	0	0	0	0	0
result	0xC6	1	**1**	0	0	0	1	1	0

(*result* = *value*|***mask***)

Bitwise OR operation for turning ON **bit6**

Bit Masking & Bitwise Operators

SOFTWARE PROGRAM

Bitwise XOR Operation: **Toggle Bits**

Use bitwise logical operator XOR for toggling the bits (change binary state, i.e., 0 to 1 or 1 to 0)

The 1 bit in the **mask** select which bits we want to toggle, and zero bits in the **mask** selects the bits to remain unchanged

For example,

		Binary Value							
	Hex Value	b7	b6	b5	b4	B3	b2	b1	b0
value	0x67	0	1	1	0	0	1	1	1
mask	0x11	0	0	0	1	0	0	0	1
result	0xE6	0	1	1	**1**	0	1	1	**0**

(*result = value^**mask***)

Bitwise XOR operation for toggling **bit0** & **bit4**

Frequency Calculation

SOFTWARE PROGRAM

- Microcontroller on TIVA LaunchPad run at default clock frequency of 16 MHz (16 million clock cycles per seconds)
- To generate a desired time delay, a calculated number of clock cycles of the microcontroller can be dissipated
- For external LED to flash at 1 Hz frequency (*f*), it must be turned ON and OFF for 0.5 second each (50% duty cycle) as it's flash time period is 1 second (T=1/*f*)
- A single iteration of a loop, with no body, in embedded C language roughly takes 8 clock cycles to execute, i.e. 1/2 MHz = 0.5 µs
- Therefore, by controlling the number of iterations of the loop, we may generate any time duration required to produce LED flash rate

HARDWARE CONNECTIONS

LED Interfacing at GPIO Port E Pin 1

TM4C123GH6PM GPIO

HARDWARE CONNECTIONS

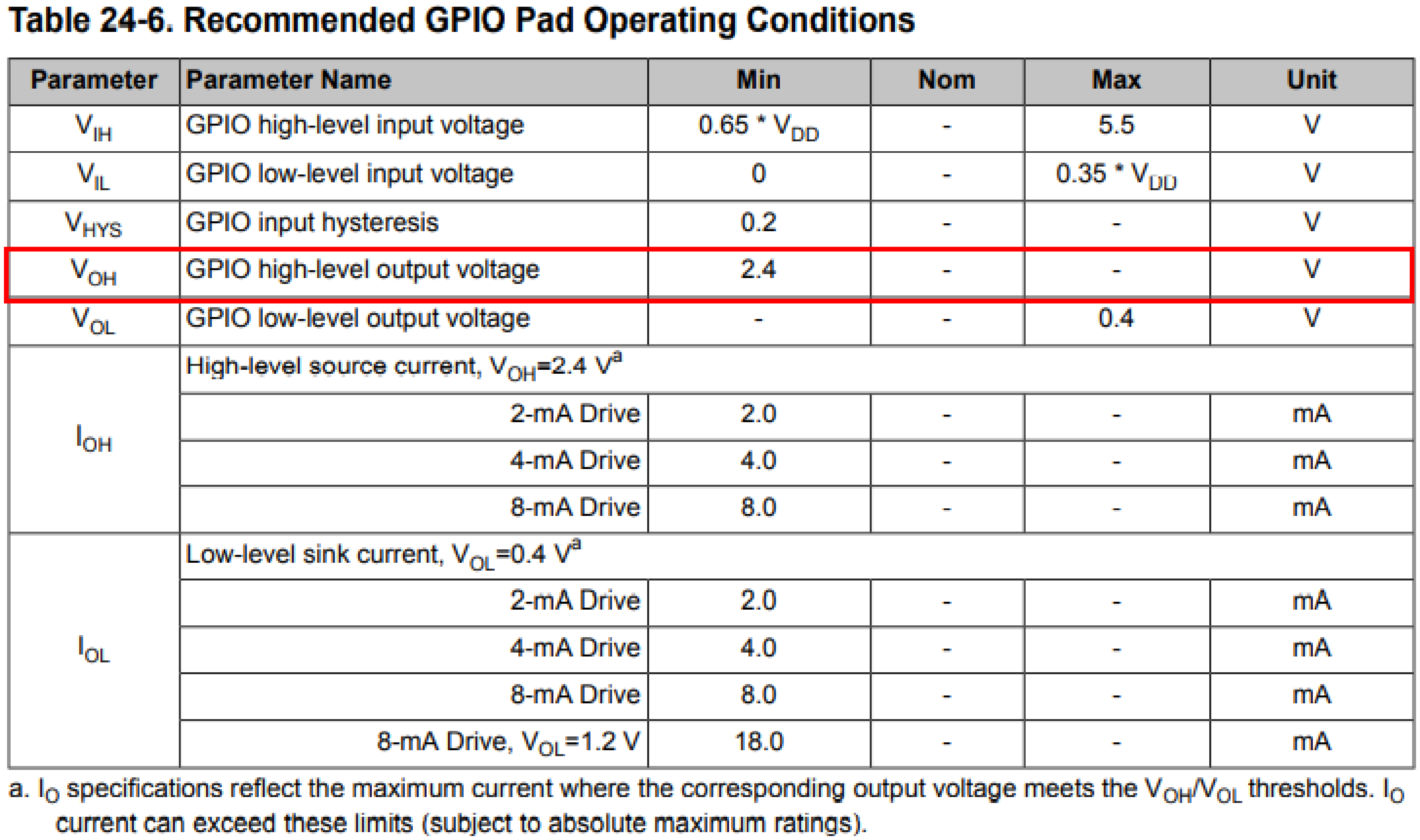


Table 24-6. Recommended GPIO Pad Operating Conditions

Parameter	Parameter Name	Min	Nom	Max	Unit
V_{IH}	GPIO high-level input voltage	0.65 * V_{DD}	-	5.5	V
V_{IL}	GPIO low-level input voltage	0	-	0.35 * V_{DD}	V
V_{HYS}	GPIO input hysteresis	0.2	-	-	V
V_{OH}	GPIO high-level output voltage	2.4	-	-	V
V_{OL}	GPIO low-level output voltage	-	-	0.4	V
I_{OH}	High-level source current, V_{OH}=2.4 V[a]				
	2-mA Drive	2.0	-	-	mA
	4-mA Drive	4.0	-	-	mA
	8-mA Drive	8.0	-	-	mA
I_{OL}	Low-level sink current, V_{OL}=0.4 V[a]				
	2-mA Drive	2.0	-	-	mA
	4-mA Drive	4.0	-	-	mA
	8-mA Drive	8.0	-	-	mA
	8-mA Drive, V_{OL}=1.2 V	18.0	-	-	mA

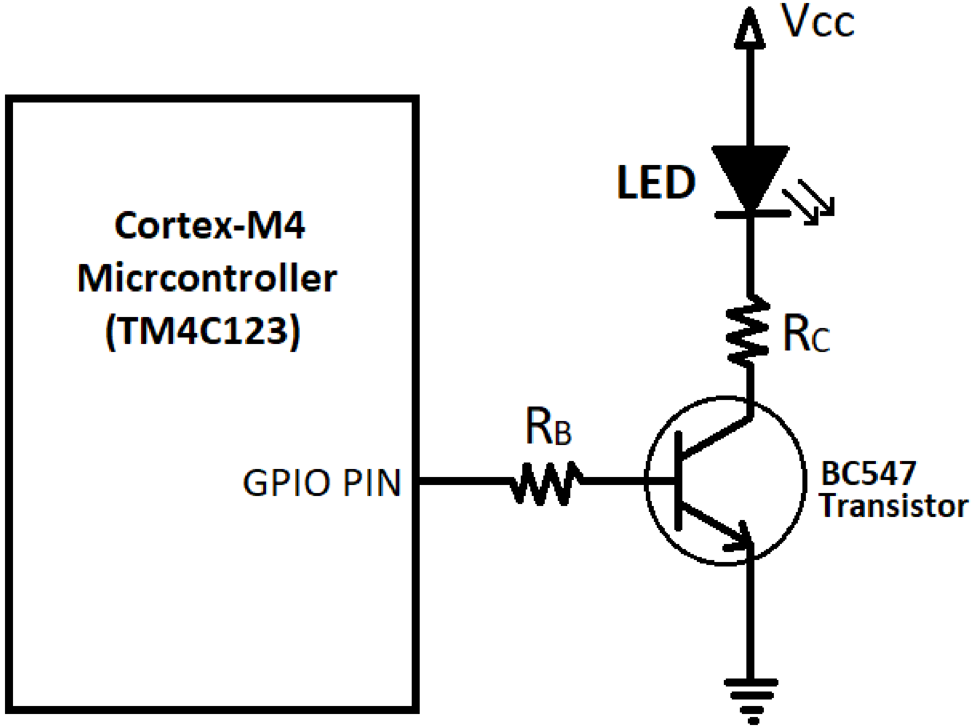
a. I_O specifications reflect the maximum current where the corresponding output voltage meets the V_{OH}/V_{OL} thresholds. I_O current can exceed these limits (subject to absolute maximum ratings).

https://www.ti.com/product/TM4C123GH6PM

Output Interfacing for LED

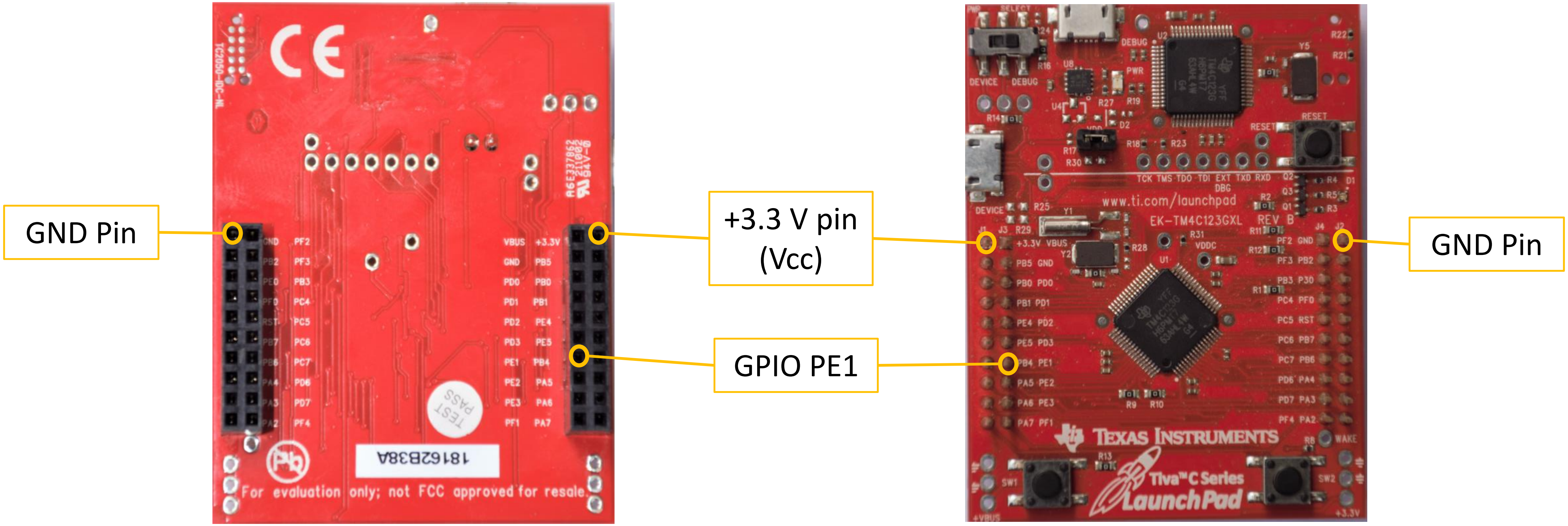
HARDWARE CONNECTIONS

- GPIO pins are not capable of providing enough sinking/sourcing current to derive peripherals
- GPIO high-level output voltage is **2.4 V** which is low to drive an LED
- Transistor can be used (as a switch) to provide the required voltage/current for LED Turn ON
- Values of resistors depends on the type of NPN Transistor used
- From **BC547 datasheet**, I_C (max.) = 100 mA & I_B = 5 mA
    - $R_C = (V_{CC}-0.7)/I_C = (3.3-0.7)/(100\text{x}10^{-3}) = 26\ \Omega$ (min.)
    - $R_B = V_{BE}/I_B = 2.4/(5\text{x}10^{-3}) = 480\ \Omega$
- R_C = **330 Ω** and R_B = **470 Ω** can be used



TIVA C Series LaunchPad

HARDWARE CONNECTIONS



GND Pin

+3.3 V pin (Vcc)

GPIO PE1

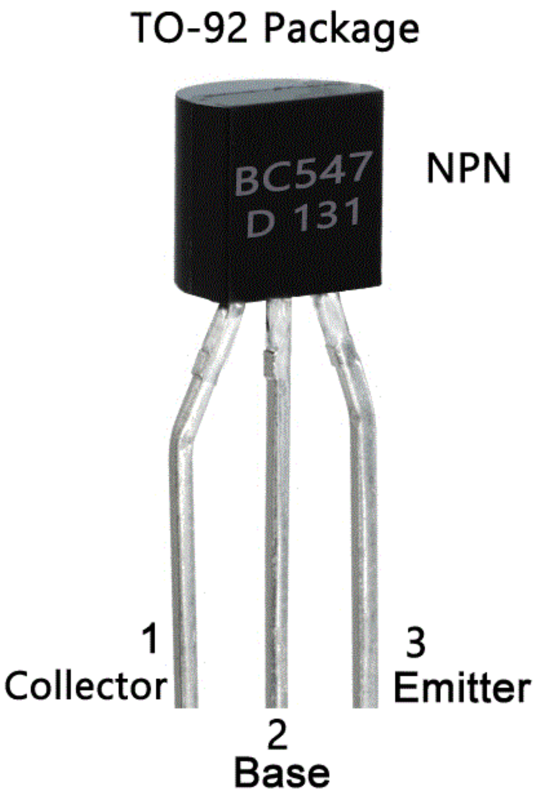
GND Pin

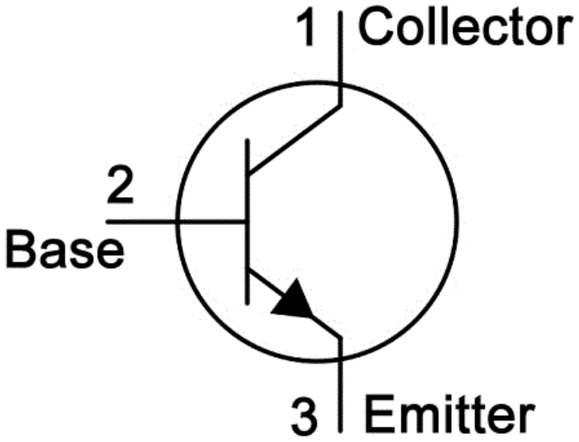
Female Connectors
Bottom Side

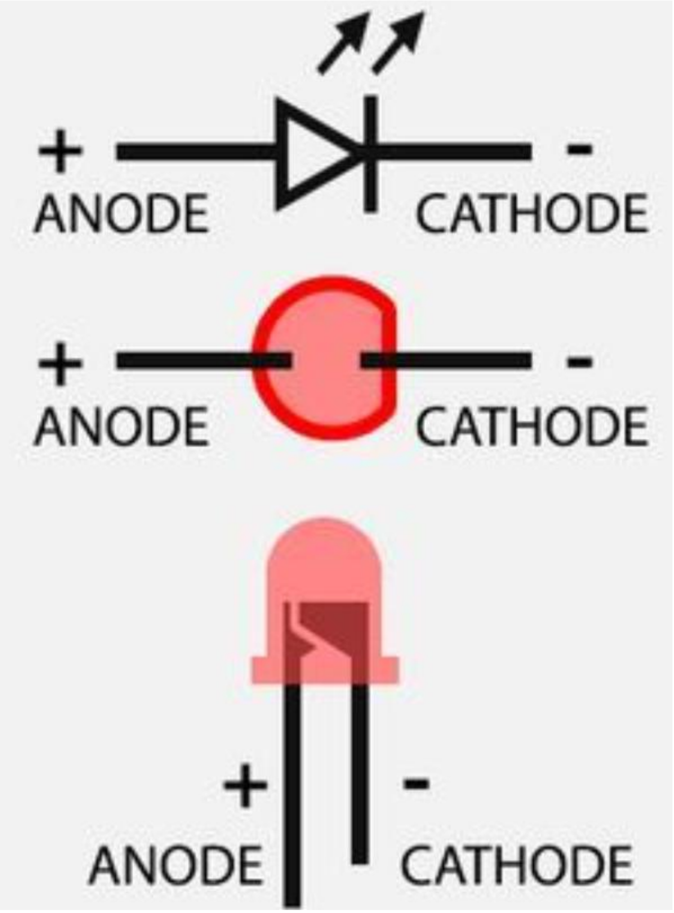
Male Connectors
Top Side

BC547 Transistor & LED

HARDWARE CONNECTIONS







https://www.componentsinfo.com/bc547-pinout-equivalent/
https://www.pinterest.com/pin/816629344903063893/

THANK YOU

Any Questions???