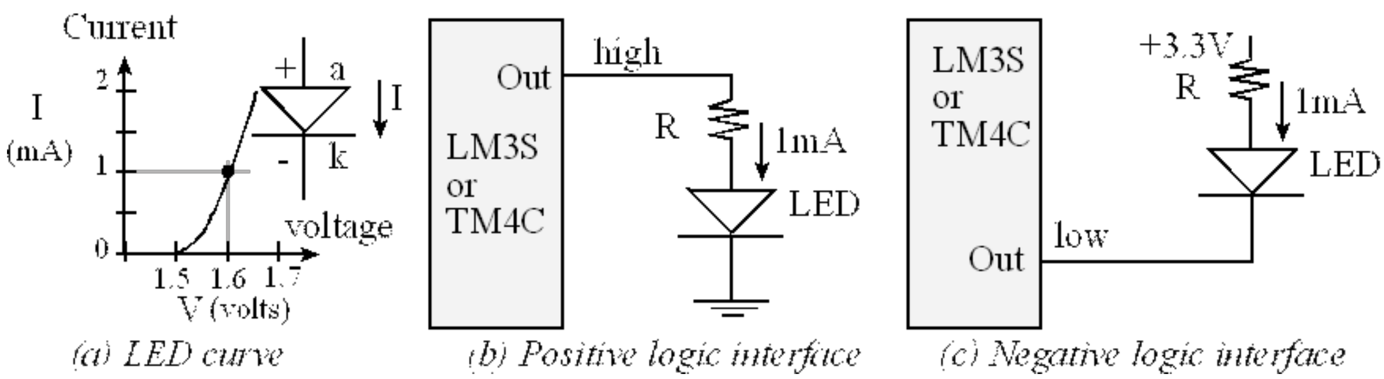

Figure 8.6. Low current LED interface (Agilent HLMP-D150).

Help

When the LED current is less than 8 mA, we can interface it directly to an output pin without using a driver. The LED shown in Figure 8.6a has an operating point of 1.6 V and 1 mA. For the positive logic interface (Figure 8.6b) we calculate the resistor value based on the desired LED voltage and current

$$R = \frac{V_{OH} - V_d}{I_d} = \frac{2.4 - 1.6}{0.001} = 800\,\Omega$$

where V_{OH} is the output high voltage of the microcontroller output pin. Since V_{OH} can vary from 2.4 to 3.3 V, it makes sense to choose a resistor from a measured value of V_{OH}, rather than the minimum value of 2.4 V. Negative logic means the LED is activated when the software outputs a zero. For the negative logic interface (Figure 8.6c) we use a similar equation to determine the resistor value

$$R = \frac{3.3 - V_d - V_{OL}}{I_d} = \frac{3.3 - 1.6 - 0.4}{0.001} = 1.3\,k\Omega$$

where V_{OL} is the output low voltage of the microcontroller output pin.

If we use a 1.2 kΩ in place of the 1.3 kΩ, then the current will be (3.3-1.6-0.4V)/1.2kΩ, which is about 1.08 mA. This slightly higher current is usually acceptable. If we use a standard resistor value of 1.5 kΩ in place of the 1.3 kΩ, then the current will be (3.3-1.6-0.4V)/1.5kΩ, which is about 0.87 mA. This slightly lower current is usually acceptable.

The software in Program 8.2 is called a **driver**, and it includes an initialization, which is called once, and two functions that can be called to turn on and off the LED. Writing software this way is called an **abstraction**, because it separates what the LED does (Init, On, Off) from how it works (PortA, bit 2, TM4C123).

CHECKPOINT 8.2

What resistor value in of Figure 8.6 is needed if the desired LED operating point is 1.7V and 2 mA? Use the negative logic interface and, VOL of 0.4V.

Hide Answer

We represent voltages in V, current in A, and resistances in ohms. For this LED, we have 3.3 - 1.7 - 0.002*R - 0.4 = 0 => R = 600Ω

```
void LED_Init(void){ volatile unsigned long delay;
  SYSCTL_RCGC2_R |= 0x01;          // 1) activate clock for Port A
  delay = SYSCTL_RCGC2_R;          // allow time for clock to start
                                   // 2) no need to unlock PA2
  GPIO_PORTA_PCTL_R &= ~0x00000F00; // 3) regular GPIO
  GPIO_PORTA_AMSEL_R &= ~0x04;     // 4) disable analog function on PA2
  GPIO_PORTA_DIR_R |= 0x04;        // 5) set direction to output
  GPIO_PORTA_AFSEL_R &= ~0x04;     // 6) regular port function
  GPIO_PORTA_DEN_R |= 0x04;        // 7) enable digital port
}
// Make PA2 high
void LED_On(void){
  GPIO_PORTA_DATA_R |= 0x04;
}
// Make PA2 low
void LED_Off(void){
  GPIO_PORTA_DATA_R &= ~0x04;
}
```

Program 8.2. Software interface for an LED on PF2 (C8_LED). Hardware interface shown in Figure 8.6b.

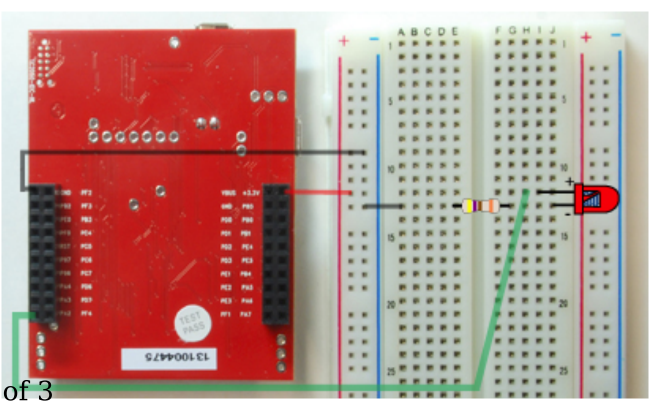

Figure 8.7. Construction of the interface of an LED to a microcomputer output (Figure 8.6b). The yellow-purple-brown resistor is 470ohm. It doesn't matter what color the wires are, but in this figure the wires are black, red and green. The two black wires are ground, the red wire is +3.3V, and the green wire is the signal ***Out****, which connects PA2 of the microcontroller to the positive side of the LED.*