```
// this constant won't change:
const int buttonPin1 = 15;     //pushbutton1 pin #
const int buttonPin2 = 14;     //pushbutton2 pin #

const int led1R =  2;       // Red LED of RGB1 pin #
const int led1B =  3;      // Blue LED of RGB1 pin #
const int led2R =  4;      // Red LED of RGB2 pin #
const int led2B =  7;      // Blue LED of RGB2 pin #
const int led3R =  8;      // Red LED of RGB3 pin #
const int led3B =  16;      // Blue LED of RGB3 pin #

const int HeatPulse =  6;      // Heat Pulse pin #
const int ColdPulse =  5;      // Cold Pulse pin #

// Variables will change:
int buttonPushCounter = 0;   // counter for the number of button presses
int buttonState1;         // current state of the button
int buttonState2;
int lastButtonState1;     // previous state of the button
int lastButtonState2;

void setup() {
  pinMode(led1R, OUTPUT);
  pinMode(led1B, OUTPUT);
  pinMode(led2R, OUTPUT);
  pinMode(led2B, OUTPUT);
  pinMode(led3R, OUTPUT);
  pinMode(led3B, OUTPUT);

  pinMode(HeatPulse, OUTPUT);
  pinMode(ColdPulse, OUTPUT);

  // initialize the pushbutton pins as inputs:
  pinMode(buttonPin1, INPUT);
  pinMode(buttonPin2, INPUT);

  // initialize serial communication:
  Serial.begin(9600);

// Blink all three LEDs purple 3 times to indicate that the TEB is ready to use
for(int i=0; i<3; i++) {
  digitalWrite(led1R, HIGH);
  digitalWrite(led1B, HIGH);
```

```
  digitalWrite(led2R, HIGH);
  digitalWrite(led2B, HIGH);
  digitalWrite(led3R, HIGH);
  digitalWrite(led3B, HIGH);
    delay(100);
  digitalWrite(led1R, LOW);
  digitalWrite(led1B, LOW);
  digitalWrite(led2R, LOW);
  digitalWrite(led2B, LOW);
  digitalWrite(led3R, LOW);
  digitalWrite(led3B, LOW);
    delay(100);
  }
}

void loop() {
  // read the pushbutton input pin:
  buttonState1 = digitalRead(buttonPin1);
  buttonState2 = digitalRead(buttonPin2);

  // compare the buttonState to its previous state
if (buttonState1 != lastButtonState1) {
    // if the state has changed, increment the counter
    if (buttonState1 == HIGH) {
      // if the current state is HIGH then the button
      // wend from off to on:
      buttonPushCounter++;
      Serial.println("Up");
    }
else if (buttonState2 != lastButtonState2) {
    // if the state has changed, increment the counter
    if (buttonState2 == HIGH) {
      // if the current state is HIGH then the button
      // wend from off to on:
      buttonPushCounter--;
      Serial.println("Down");

      Serial.println(buttonPushCounter);
    }
else {
      // if the current state is LOW then the button
      // wend from on to off:
      Serial.println("off");
```

```
  }
 }
 // save the current state as the last state,
 //for next time through the loop
 lastButtonState1 = buttonState1;
 lastButtonState2 = buttonState2;

 // turns on the LED every four button pushes by
 // checking the modulo of the button push counter.
 // the modulo function gives you the remainder of
 // the division of two numbers:
 if (buttonPushCounter == 3) {
   digitalWrite(led3R, HIGH);
   digitalWrite(led2R, HIGH);
   digitalWrite(led1R, HIGH);
   digitalWrite(led3B, LOW);
   digitalWrite(led2B, LOW);
   digitalWrite(led1B, LOW);

   digitalWrite(HeatPulse, HIGH);
   delay(3000);
   digitalWrite(HeatPulse, LOW);
   delay(1000);
 }

 else if (buttonPushCounter == 2) {
   digitalWrite(led3R, LOW);
   digitalWrite(led2R, HIGH);
   digitalWrite(led1R, HIGH);
   digitalWrite(led3B, LOW);
   digitalWrite(led2B, LOW);
   digitalWrite(led1B, LOW);

   digitalWrite(HeatPulse, HIGH);
   delay(2000);
   digitalWrite(HeatPulse, LOW);
   delay(1000);
 }
 else if (buttonPushCounter == 1) {
   digitalWrite(led3R, LOW);
   digitalWrite(led2R, LOW);
   digitalWrite(led1R, HIGH);
   digitalWrite(led3B, LOW);
```

```
  digitalWrite(led2B, LOW);
  digitalWrite(led1B, LOW);

  digitalWrite(HeatPulse, HIGH);
  delay(1000);
  digitalWrite(HeatPulse, LOW);
  delay(1000);
 }
else if (buttonPushCounter == 0) {
  digitalWrite(led3R, LOW);
  digitalWrite(led2R, LOW);
  digitalWrite(led1R, LOW);
  digitalWrite(led3B, LOW);
  digitalWrite(led2B, LOW);
  digitalWrite(led1B, LOW);
 }
else if (buttonPushCounter == -1) {
  digitalWrite(led3R, LOW);
  digitalWrite(led2R, LOW);
  digitalWrite(led1R, LOW);
  digitalWrite(led3B, LOW);
  digitalWrite(led2B, LOW);
  digitalWrite(led1B, HIGH);

  digitalWrite(ColdPulse, HIGH);
  delay(1000);
  digitalWrite(ColdPulse, LOW);
  delay(1000);
 }
else if (buttonPushCounter == -2) {
  digitalWrite(led3R, LOW);
  digitalWrite(led2R, LOW);
  digitalWrite(led1R, LOW);
  digitalWrite(led3B, LOW);
  digitalWrite(led2B, HIGH);
  digitalWrite(led1B, HIGH);

  digitalWrite(ColdPulse, HIGH);
  delay(2000);
  digitalWrite(ColdPulse, LOW);
  delay(1000);
 }
else if (buttonPushCounter == -3) {
```

```
    digitalWrite(led3R, LOW);
    digitalWrite(led2R, LOW);
    digitalWrite(led1R, LOW);
    digitalWrite(led3B, HIGH);
    digitalWrite(led2B, HIGH);
    digitalWrite(led1B, HIGH);

    digitalWrite(ColdPulse, HIGH);
    delay(3000);
    digitalWrite(ColdPulse, LOW);
    delay(1000);
  }

  else {
    digitalWrite(led3R, LOW);
    digitalWrite(led2R, LOW);
    digitalWrite(led1R, LOW);
    digitalWrite(led3B, LOW);
    digitalWrite(led2B, LOW);
    digitalWrite(led1B, LOW);
      }
  }
}
```