

```
/*  
 * File:  IRTemp.h  
 * Version: 1.0  
 * Author: Andy Gelme (@geekscape)  
 * License: GPLv3  
 *  
 * See www.freetronics.com/irtemp for more information.  
 */
```

```
#ifndef IRTEMP_h  
#define IRTEMP_h
```

```
#if ARDUINO >= 100  
  #include "Arduino.h"  
#else  
  #include "WProgram.h"  
#endif
```

```
typedef enum {  
  CELSIUS,  
  FAHRENHEIT  
} TempUnit;
```

```
class IRTemp {  
private:  
  byte _pinAcquire;  
  byte _pinClock;  
  byte _pinData;
```

```
float convertFahrenheit(float celsius);
float getTemperature(TempUnit scale, byte dataType);
float decodeTemperature(volatile byte data[]);
void sensorEnable(bool state);
bool validData(byte data[]);

public:
  IRTemp(byte pinAcquire, byte pinClock, byte pinData);
  float getAmbientTemperature(TempUnit scale=CELSIUS);
  float getIRTemperature(TempUnit scale=CELSIUS);

  // Deprecated calls:
  // for compatibility with old API which used boolean for scale,
  // recommended to use TempUnit instead
  inline float getAmbientTemperature(bool scale) {
    return getAmbientTemperature(scale ? FAHRENHEIT : CELSIUS);
  }
  inline float getIRTemperature(bool scale) {
    return getIRTemperature(scale ? FAHRENHEIT : CELSIUS);
  }
};
#endif
```