

```
/*  
 * File:      readTemperature.ino  
 * Author:   Andy Gelme (@geekscape) & Angus Gratton (angus at  
freetronics.com)  
 * License:  GPLv3  
 *  
 * For more information see www.freetronics.com/irtemp  
 *  
 */
```

```
#include "IRTemp.h"
```

```
int sensorpins[3][3] = {  
  {2, 3, 4},  
  {5, 6, 7},  
  {8, 9, 10}  
};
```

```
int i = 0;
```

```
byte PIN_ACQUIRE;
```

```
byte PIN_CLOCK ;
```

```
byte PIN_DATA ;
```

```
static const TempUnit SCALE=FAHRENHEIT;
```

```
void setup(void) {  
  Serial.begin(9600);  
}
```

```
void loop(void) {  
  for(i=0;i<2;i=i+1){  
    PIN_DATA = sensorpins[i][0];  
    PIN_CLOCK = sensorpins[i][1];  
    PIN_ACQUIRE = sensorpins[i][2];
```

```
IRTemp irTemp(PIN_ACQUIRE, PIN_CLOCK, PIN_DATA);
```

```
float irTemperature = irTemp.getIRTemperature(SCALE);  
printTemperature("IR", irTemperature,i);
```

```
delay(500);  
float ambientTemperature = irTemp.getAmbientTemperature(SCALE);  
printTemperature("Ambient", ambientTemperature, i);  
  
delay(1000);  
}  
}  
  
void printTemperature(  
char *type,  
float temperature, int i) {  
  
Serial.print(type);  
Serial.print(",");  
Serial.print(i);  
Serial.print(",");  
  
if (isnan(temperature)) {  
    Serial.println("Failed");  
}  
else {  
    Serial.println(temperature);  
}  
}
```