

```

1  /*  ChipTemp.cpp
2  *   Cette classe a été conçue par Albert van Daln
3  *   pour réaliser la lecture de la température
4  *   interne du microcontrôleur ATmega328P.
5  *
6  *   Code modifié par: T. Wong
7  *   08-2018
8  *   08-2020
9  *
10 *   Le notice originale:
11 *       ATmega328 temperature sensor interface
12 *       Rev 1.0 Albert van Dalen (www.avdweb.nl)
13 *       Based on "InternalTemp"
14 *       Requires 166 ... 204 bytes program memory
15 *       Resolution 0.1 degree
16 */
17 #include <Arduino.h>
18 #include "ChipTemp.h"
19
20
21 /* -----
22  *   Cette fonction membre initialise l'ADC du
23  *   microcontrôleur.
24  * ----- */
25 inline void ChipTemp::initialize()
26 {
27     ADMUX = 0xC8; // select reference, select temp sensor
28     delay(10); // wait for the analog reference to stabilize
29     readAdc(); // discard first sample (never hurts to be safe)
30 }
31
32 /* -----
33  *   Lire la valeur de la tension aux bornes de la
34  *   diode interne.
35  * ----- */
36 inline int16_t ChipTemp::readAdc()
37 {
38     ADCSRA |= _BV(ADSC); // start the conversion
39     while (bit_is_set(ADCSRA, ADSC)); // ADSC is cleared when the conversion finishes
40     return (ADCL | (ADCH << 8)); // combine bytes
41 }
42
43 /* -----
44  *   Convertir la tension lue en deci-degrés Celsius.
45  *   Les autres unités de température sont dérivées
46  *   du deci-degrés Celsius.
47  * ----- */
48 float ChipTemp::deciCelsius()
49 {
50     int32_t averageTemp = 0;
51     // Une bonne pratique d'initialiser à chaque fois car
52     // l'ADC peut être reprogrammé entre temps par d'autres
53     // routines d'un programme.
54     initialize();
55     for (int i = 0; i < samples; i++) averageTemp += readAdc();
56     averageTemp -= offsetFactor;
57     return averageTemp / divideFactor; // return deci degree Celsius
58 }
59

```

```
60 float ChipTemp::celsius()
61 {
62     return deciCelsius() / 10;
63 }
64
65 float ChipTemp::deciFahrenheit()
66 {
67     return (9 * deciCelsius() + 1600) / 5;
68 }
69
70 float ChipTemp::fahrenheit()
71 {
72     return (9 * deciCelsius() + 1600) / 50; // do not use deciFahrenheit()/10;
73 }
74
```