François RIGAUD

OBSERVATOIRE DE
PARIS, GEPI ,
11, Rue Marcellin Berthelot
92195 Meudon (France)

Meudon, le 02/03/11

Test du convertisseur ADC de la plateforme MBED

Logiciel :

Compilateur MBED:

```
#include "mbed.h"
#define nb_data 14000  // unsigned short : 16 bits  LPC1768 : 32 K.Octets
#define periode_acquis 100  // periode entre 2 aquisitions en µs
LocalFileSystem local("local");
Serial pc(USBTX, USBRX);
AnalogIn adc(p15);
unsigned short buffer[nb_data];
int main() {
   pc.printf("Depart acquisition!\n");
   wait(0.5);
   for (int i=0; i<nb_data; i++)
      {
      //wait_us(periode_acquis);
      buffer[i]=adc.read_u16();
      }
   pc.printf("Sauvegarde, Opening File...\n"); // Drive should be marked as removed
   FILE *fp = fopen("/local/adc.csv", "w");
   if(!fp) {
      pc.printf("File /local/adc.csv could not be opened!\n");
      exit(1);
   }
   for (int i=0; i<nb_data; i++)
      {
      fprintf(fp,"%i\n",buffer[i]);
      }
   pc.printf("Closing File..., fini!\n");
   fclose(fp);
}
```

Sous windows :

Excel est associé au .CSV donc le fichier batch suivant :

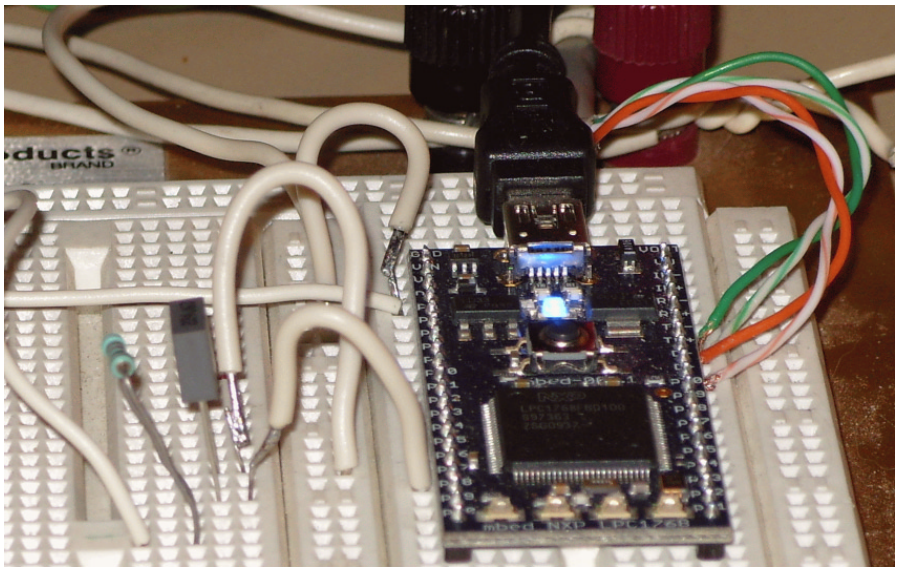
```
Rem ------------------------------
start f:\adc.csv
start D:\document\travail\20-podet\logiciel_emb\pc_graph_adc\graph_f_adc_csv.xls
rem --------------------------------
```

permet

- l'ouverture du fichier stocké sur le clef USB, mappé disque « f : » sous windows
- l'ouverture d'un 2e fichier excel, « graph_f_adc_csv.xls » affichant un graphique. Le graphique pointe sa plage de donnée sur le fichier ouvert, 1ere colonne :

« =ADC.CSV!$A$1:$A$14001 »
De plus, une case excel réalise la moyenne des data et une 2e fait l'écart type

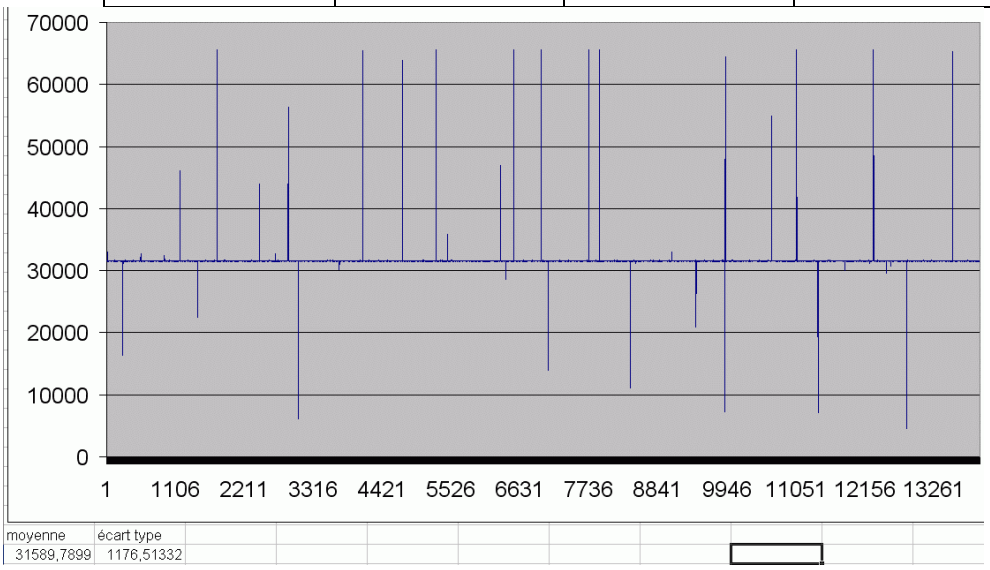
Mesures :



Pour certaine acquisition, un filtre BP type RC est utilisé. F=1/(2.PI.R.C)
Note : le LPC1768 est limité en acquisition à 200 KHz, donc la période mini est de 5 µ.secondes !

Acquisition 1 :

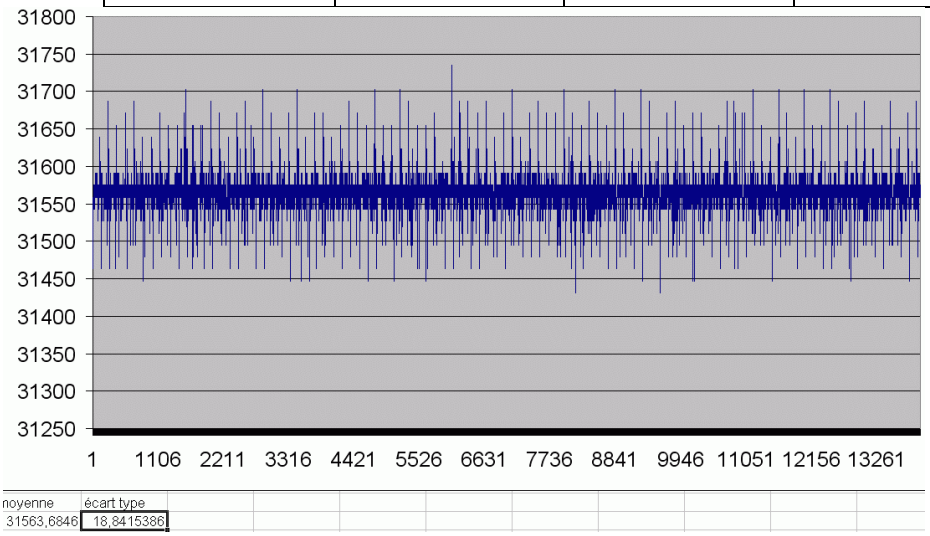
Alim. MBED USB/Alim labo	BP Filtrage, Hz	Période d'acquisition µs	Tension injectée, Volt
USB	non	5	1.594



Conclusion: inexploitable

Acquisition 2:

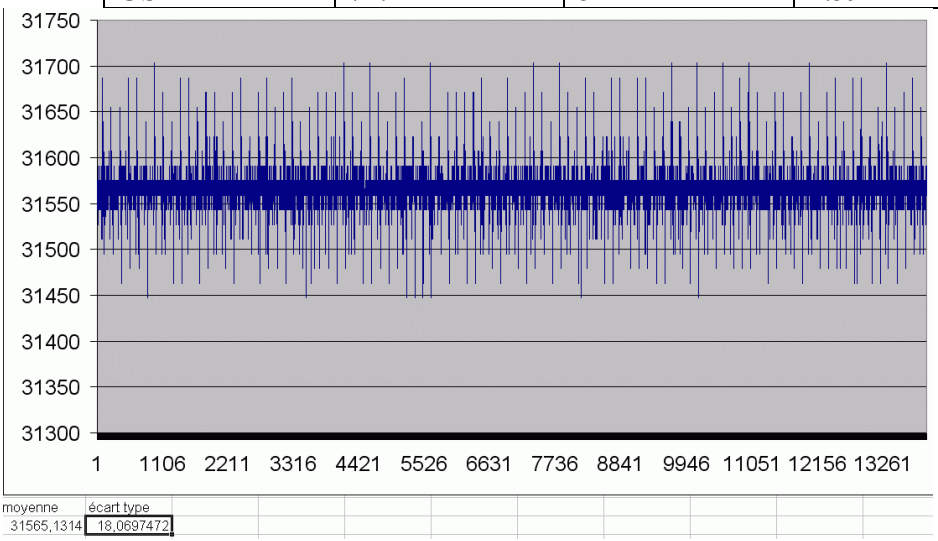
Alim. MBED USB/Alim labo	BP Filtrage, Hz	Période d'acquisition µs	Tension injectée, Volt
USB	724	5	1.594



Extrait de data:
31463,31543,31543,31543,31543,31543,31559,31543,31559,31575,31543,31559,31559,
31575,31559,31559,31559,31575,31559,31559

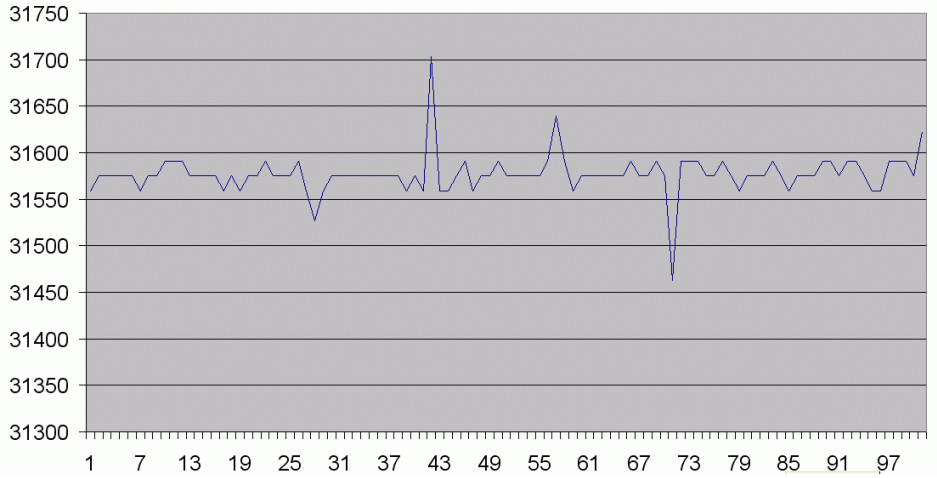
Aquisition 3:

Alim. MBED USB/Alim labo	BP Filtrage, Hz	Période d'acquisition µs	Tension injectée, Volt
USB	72.4	5	1.594



Conclusion: un filtrage plus efficace ne change rien!

Graph sur un échantillon de 100:



Acquisition 4 :

Alim. MBED USB/Alim labo	BP Filtrage, Hz	Période d'acquisition µs	Tension injectée, Volt
Alim labo	72.4	100	1.594

Graphique identique

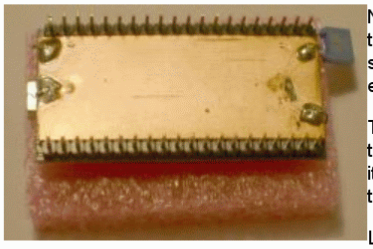
En conclusion préliminaire:

La lecture de l'article http://www.nxp.com/documents/application_note/AN10974.pdf permet d'envisager que la plateforme MBED n'est pas très bien conçue pour l'utilisation des convertisseurs ADC.
Une solution trouvée par un utilisateur de la plateforme MBED :
http://mbed.org/forum/mbed/topic/131/?page=1#comment-1841

Dave Malham

04 Jan 2010 . Edited: 04 Jan 2010

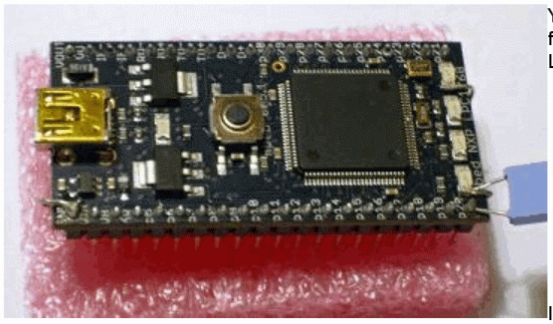
Okay, some reuslts. Firstly, I have added an extra groundplane to the Mbed using some thin (0.75mm) pcb I had, cut to size (19mm wide x 52 mm long), mounted under the mbed between the mounting strip of the pins.



Notice the solder blobs. The midle one on the left connects the top and bottom of the pcb together - it does not connect to anywhere else - and certainly not to the shield of the USB connector, which must NOT be connected to ground at the Mbed end.

The other blobs are wires going to the topside of the Mbed - top left goes to pin 1, the two on the right go to the grounded end of two of the LED's. Note that, although it should not be possible to short out to any of the components on the underside of the Mbed, I added some insulation, just in case.

Looking at the top of the modified Mbed, you can see the connections to pin 1 and the LED's.



You can also see at the bottm right, a (blue) 100nF capacitor from the AdcInput on pin 20 and the ground connection on LED1 (which also connects to the new groundplane).

I modified the code that Simon posted, so that the printf statement read;

En 2e conclusion, il semble nécessaire pour obtenir de meilleurs résultats sur la plateforme MBED, d'implémenter un algorithme de moyenne flottante avec réjection des « fausses » mesures.

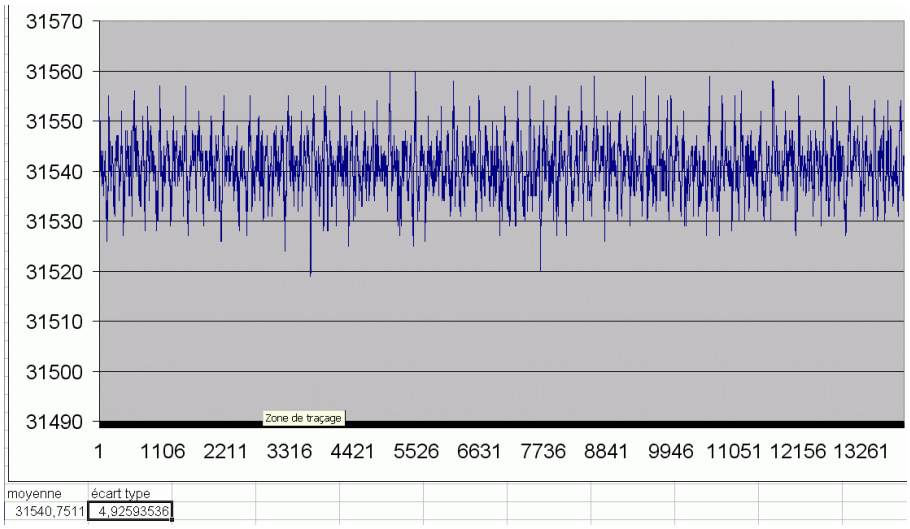
Code de test d'une moyenne flottante sur 16 acquisitions :

```
#include "mbed.h"
#define nb_data 14000  // unsigned short : 16 bits  LPC1768 : 32 K.Octets
#define periode_acquis 95  // periode entre 2 aquisitions en µs
LocalFileSystem local("local");
Serial pc(USBTX, USBRX);
AnalogIn adc(p15);
unsigned short buffer[nb_data];
unsigned short buf_moy[16]; // buffer tournant
unsigned int somme;
char j;

int main() {
   pc.printf("Depart acquisition!\n");
   wait(0.5);
   // remplissage tableau à moyenner
   somme=0;
   for (j=0; j<16; j++)
      {
      buf_moy[j]=adc.read_u16();
      somme+=buf_moy[j];
      }
   j=0;
```

```
    // acquisition
    for (int i=0; i<nb_data; i++)
        {
        somme-=buf_moy[j];
        buf_moy[j]=adc.read_u16();
        somme+=buf_moy[j];
        j++;
        if (j==16) {j=0;}
        buffer[i]= (somme / 16);
        }
    pc.printf("Sauvegarde, Opening File...\n"); // Drive should be marked as removed
    FILE *fp = fopen("/local/adc.csv", "w");
    if(!fp) {
        pc.printf("File /local/adc.csv could not be opened!\n");
        exit(1);
    }
    for (int i=0; i<nb_data; i++)
        {
        fprintf(fp,"%i\n",buffer[i]);
        }
    pc.printf("Closing File..., fini!\n");
    fclose(fp);
}
```

Le résultat est :

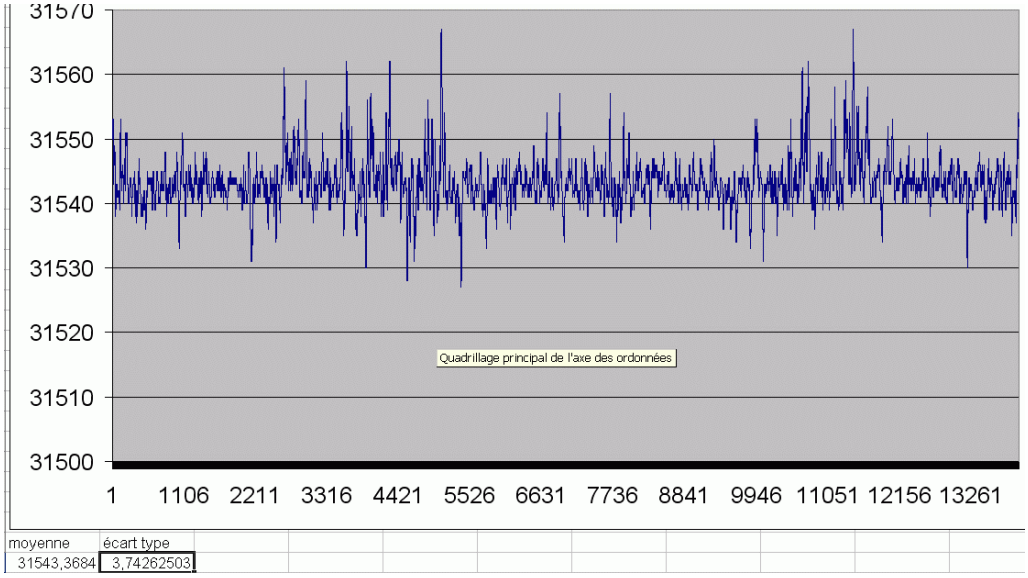


L'écart type passe de 18 à 5 !

La partie de code :

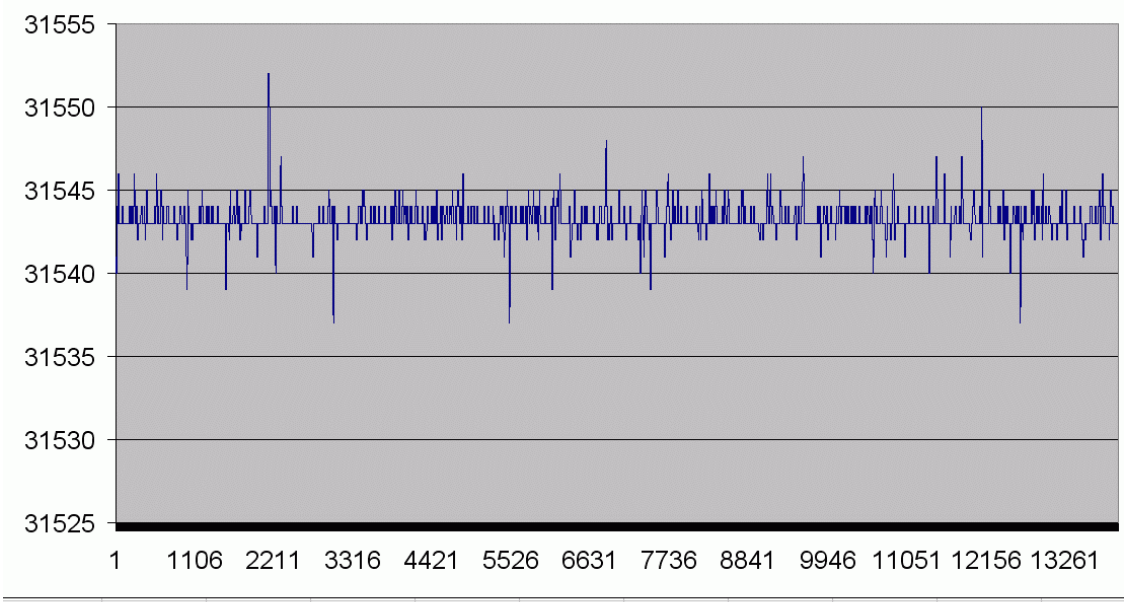
```
// acquisition
   for (int i=0; i<nb_data; i++)
      {
      somme-=buf_moy[j];
      buf_moy[j]=adc.read_u16();
      // si parasite: rejet
      if (abs(buf_moy[j]-buffer[i-1])>5) { buf_moy[j]=adc.read_u16(); }
      somme+=buf_moy[j];
      j++;
```

Donne ceci :



Le code suivant fait encore plus fort avec de multiple rejections et un délai entre les périodes agitées :

```
   for (int i=0; i<nb_data; i++)
      {
      somme-=buf_moy[j];
      buf_moy[j]=adc.read_u16();
      // si parasite: rejet
      if (abs(buf_moy[j]-buffer[i-1])>5)
         {
         wait_us(500);
         buf_moy[j]=adc.read_u16();
         }
      if (abs(buf_moy[j]-buffer[i-1])>5)
         {
         wait_us(500);
         buf_moy[j]=adc.read_u16();
         }
      if (abs(buf_moy[j]-buffer[i-1])>5)
         {
         wait_us(500);
         buf_moy[j]=adc.read_u16();
         }
      somme+=buf_moy[j];
      j++;
```



moyenne	écart type
31543,2433	0,93755944

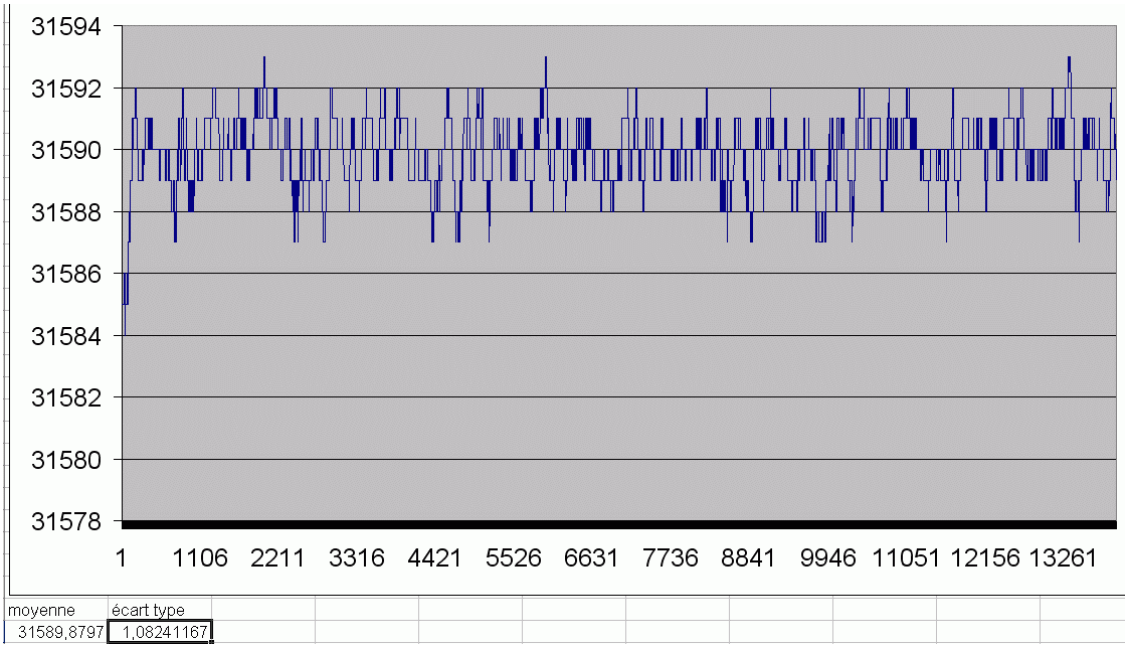
Voici enfin un code déboguer qui fait mieux :

```
#include "mbed.h"
#define nb_data 14000  // unsigned short : 16 bits  LPC1768 : 32 K.Octets
#define periode_acquis 95  // periode entre 2 aquisitions en µs
#define size_buf_moy 100    // attention à l'overflow de somme
#define delais_acquis 50     // delais en µseconde suite à une acquisition à refaire
#define delais_rejet 100     // delais en µseconde suite à une acquisition à refaire
#define seuil_refaire 2
LocalFileSystem local("local");
Serial pc(USBTX, USBRX);
AnalogIn adc(p15);
unsigned short buffer[nb_data];
unsigned short buf_moy[size_buf_moy]; // buffer tournant
unsigned long long somme;
char j;

int main() {
   pc.printf("Depart acquisition!\n");
   wait(0.5);
   // remplissage tableau à moyenner
   somme=0;
   for (j=0; j<size_buf_moy; j++)
       {
       buf_moy[j]=adc.read_u16();
       somme+=buf_moy[j];
       wait_us(delais_acquis);
       }
   j=0;
   // acquisition
   for (int i=0; i<nb_data; i++)
       {
```

```
        somme-=buf_moy[j];
        buf_moy[j]=adc.read_u16();
        // si parasite: rejet
        if (abs(buf_moy[j]-buffer[i-1])>seuil_refaire)
            {
            wait_us(delais_rejet);
            buf_moy[j]=adc.read_u16();
            //pc.printf("1ere pass \n");
            }
        else if (abs(buf_moy[j]-buffer[i-1])>seuil_refaire)
            {
            wait_us(delais_rejet);
            buf_moy[j]=adc.read_u16();
            pc.printf("2e pass \n");
            }
        else if (abs(buf_moy[j]-buffer[i-1])>seuil_refaire)
            {
            wait_us(delais_rejet);
            buf_moy[j]=adc.read_u16();
            pc.printf("3e pass \n");
            }
        somme+=buf_moy[j];
        j++;
        if (j==size_buf_moy) j=0;
        buffer[i]= (somme / size_buf_moy);
        wait_us(delais_acquis);
        }
    pc.printf("Sauvegarde, Opening File...\n"); // Drive should be marked as removed
    FILE *fp = fopen("/local/adc.csv", "w");
    if(!fp) {
        pc.printf("File /local/adc.csv could not be opened!\n");
        exit(1);
    }
    for (int i=0; i<nb_data; i++)
        {
        fprintf(fp,"%i\n",buffer[i]);
        }
    pc.printf("Closing File..., fini!\n");
    fclose(fp);
}
/*
Sous windows :
Excel est associé au .CSV donc le fichier batch suivant :
Rem -------------------------------
start f:\adc.csv
start D:\document\travail\20-podet\logiciel_emb\pc_graph_adc\graph_f_adc_csv.xls
rem ---------------------------------
permet
-           l'ouverture du fichier stocké sur le clef USB, mappé disque « f : » sous windows
-           l'ouverture d'un 2e fichier excel, « graph_f_adc_csv.xls » affichant un graphique.
        Le graphique pointe sa plage de donnée sur le fichier ouvert,
        1ere colonne : « =ADC.CSV!$A$1:$A$14001 »
De plus, une case excel réalise la moyenne des data: =MOYENNE(ADC.CSV!$A:$A)
et une 2e fait l'écart type: =ECARTYPE(ADC.CSV!$A:$A)

*/
```



Pour faire mieux, le code serait probablement plus sophistiqué, et beaucoup plus gourmand en ressources !
Je suis ouvert à toutes les propositions ! Better sugest ?

Cordialement
François