ISEN ALL IS DIGITAL! OUEST | yncréa

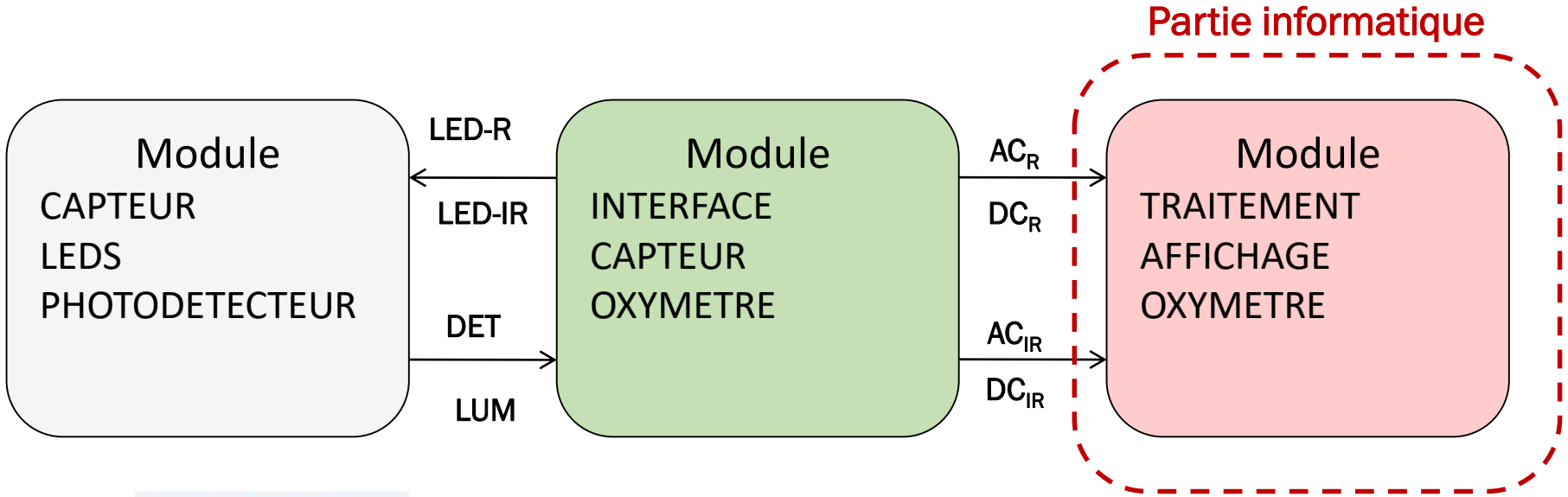
Oxymètre de pouls

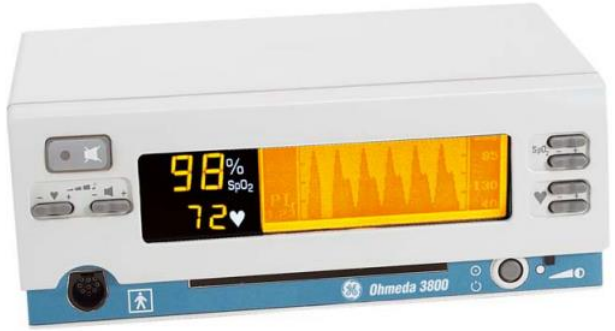
Partie informatique

Pierre-Jean BOUVET

Présentation

Description générale





Description de la partie informatique

- Récupération des valeurs mesurées de la lumière rouge (ACR et DCR) et de l'infra-rouge (ACIR et DCIR) via liaison USB
- Traitement des informations reçues afin d'éliminer les signaux parasites
- Calcul de l'oxymétrie et de fréquence cardiaque
- Visualisation des résultats via une interface utilisateur

Synoptique de la partie informatique

Carte électronique (fournie)

USB

21o

Lecture

AC_R
DC_R
AC_{IR}
DC_{IR}

FIR

AC_R
DC_R
AC_{IR}
DC_{IR}

IIR

AC_R
DC_R
AC_{IR}
DC_{IR}

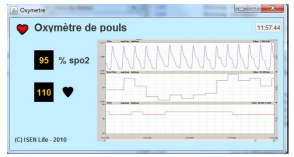
Mesures

SPO_2

Pouls

Affichage

Interface JAVA (fournie)



4o	1o	4o	1o	4o	1o	4o	1o	1o
AC_R	,	DC_R	,	AC_{IR}	,	DC_{IR}	LF	CR

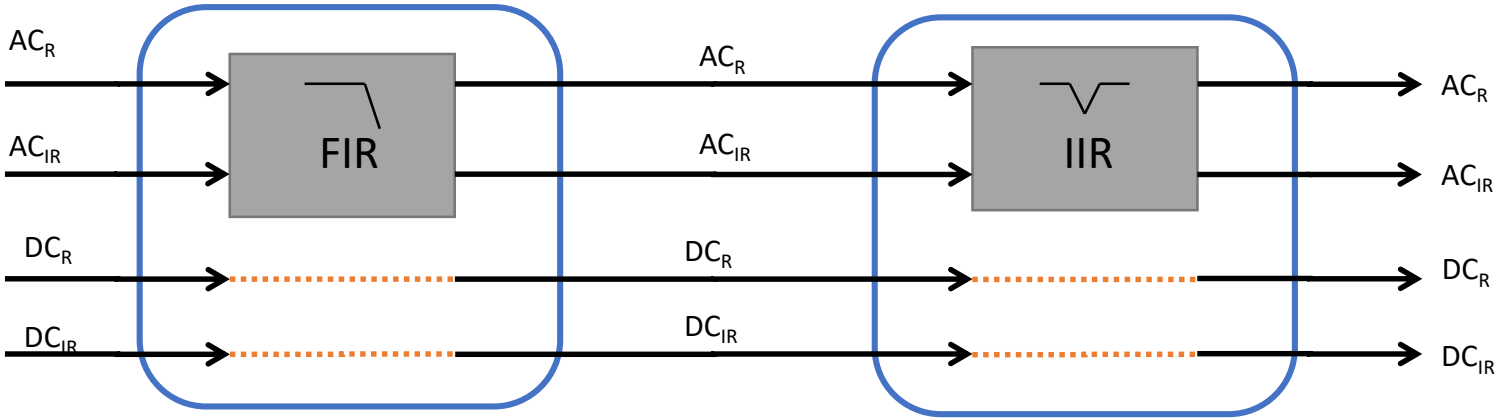
Fichiers test (fournis)

xx Module à développer pour tous

xx Module à développer pour CIR3 (Bonus pour CGSI3 / BIOST3 / CENT3)

Les différents blocs

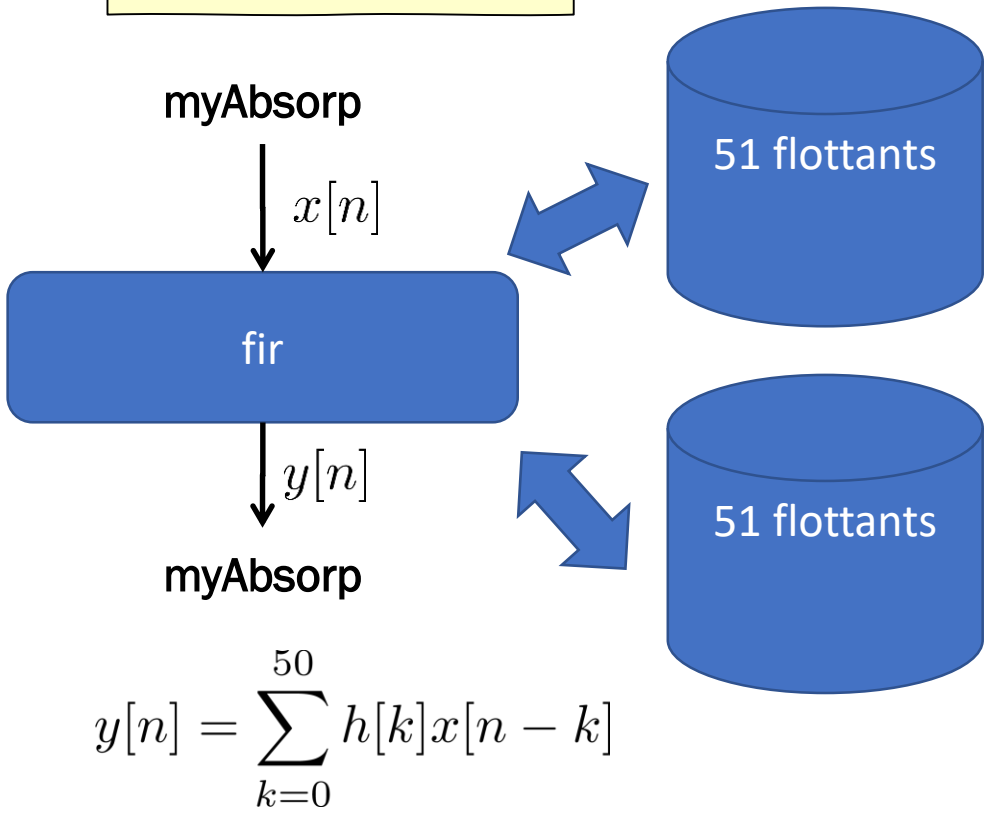
Filtrage (deux blocs : FIR et IIR)



- Points essentiels
  - Filtrage passe-bas (FIR) et passe-haut (IIR) des composantes AC_R et AC_{IR}
- Points critiques
  - Implémentation des filtres

Filtrage FIR

```
typedef struct{
          float acr;
          float dcr;
          float acir;
          float dcir;
} absorp;
```

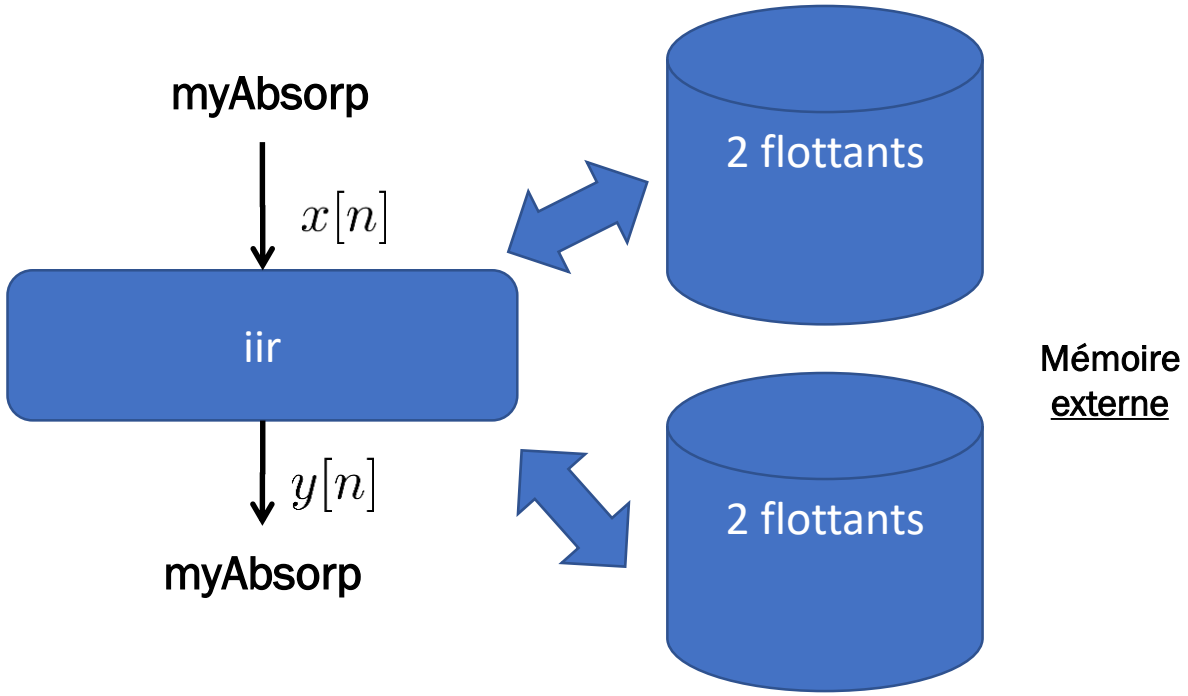


Mémoire externe

$$y[n] = \sum_{k=0}^{50} h[k]x[n-k]$$

k	$h[k]$	k	$h[k]$	k	$h[k]$
0	1.4774946e-019	17	3.1294938e-002	34	2.7892178e-002
1	1.6465231e-004	18	3.4578348e-002	35	2.4459630e-002
2	3.8503956e-004	19	3.7651889e-002	36	2.1082435e-002
3	7.0848037e-004	20	4.0427695e-002	37	1.7838135e-002
4	1.1840522e-003	21	4.2824111e-002	38	1.4793934e-002
5	1.8598621e-003	22	4.4769071e-002	39	1.2004510e-002
6	2.7802151e-003	23	4.6203098e-002	40	9.5104679e-003
7	3.9828263e-003	24	4.7081811e-002	41	7.3374938e-003
8	5.4962252e-003	25	4.7377805e-002	42	5.4962252e-003
9	7.3374938e-003	26	4.7081811e-002	43	3.9828263e-003
10	9.5104679e-003	27	4.6203098e-002	44	2.7802151e-003
11	1.2004510e-002	28	4.4769071e-002	45	1.8598621e-003
12	1.4793934e-002	29	4.2824111e-002	46	1.1840522e-003
13	1.7838135e-002	30	4.0427695e-002	47	7.0848037e-004
14	2.1082435e-002	31	3.7651889e-002	48	3.8503956e-004
15	2.4459630e-002	32	3.4578348e-002	49	1.6465231e-004
16	2.7892178e-002	33	3.1294938e-002	50	1.4774946e-019

Filtrage IIR

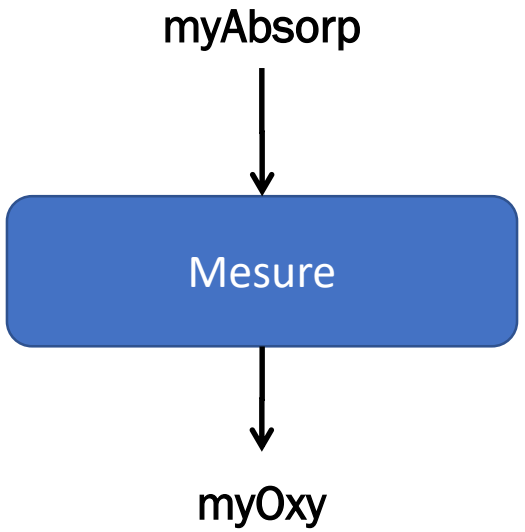


$$y[n] = x[n] - x[n-1] + \alpha y[n-1] \quad \text{avec} \quad \alpha = 0.992$$

Mesure

- Prend en entrée les valeurs mesurées de la lumière rouge (AC_R et DC_R) et de l'infra-rouge (AC_{IR} et DC_{IR})

- Extraction à partir des différentes composantes des quantités suivantes :
  - Taux de saturation d'oxygène dans le sang (SpO_2)
  - Fréquence cardiaque (pouls)

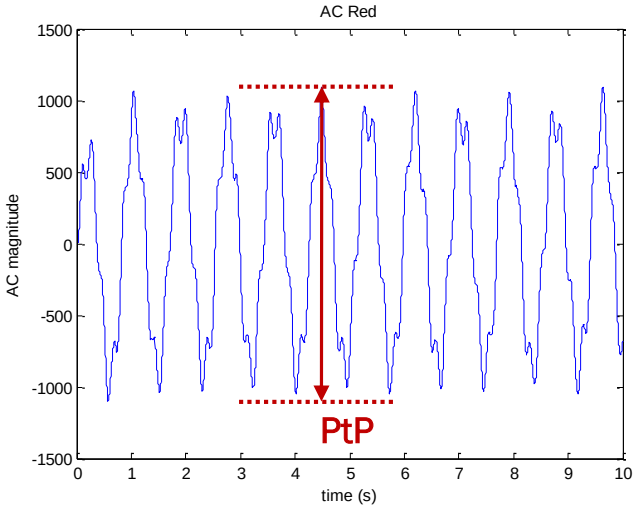


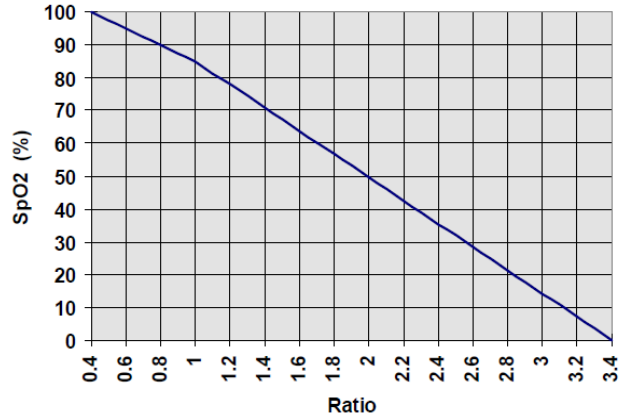

```
typedef struct{
          int spo2;  /*!< SPO2 */
          int pouls;  /*!< Pouls */
} oxy;
```

Mesure $Sp0_2$

- Points essentiels
  - Calcul de la valeur crête de chaque composante AC
  - Calcul de RsIR puis $Sp0_2$ via une table de correspondance
- Points critiques
  - Calcul de $Sp0_2$ au "fil de l'eau"

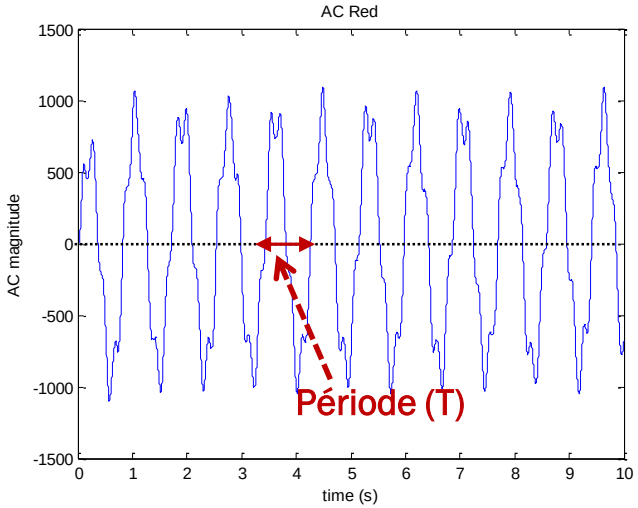
$$RsIR = \frac{\frac{\mathrm{PtP}(AC_R)}{DC_R}}{\frac{\mathrm{PtP}(AC_{IR})}{DC_{IR}}}$$





Mesure de la fréquence cardiaque

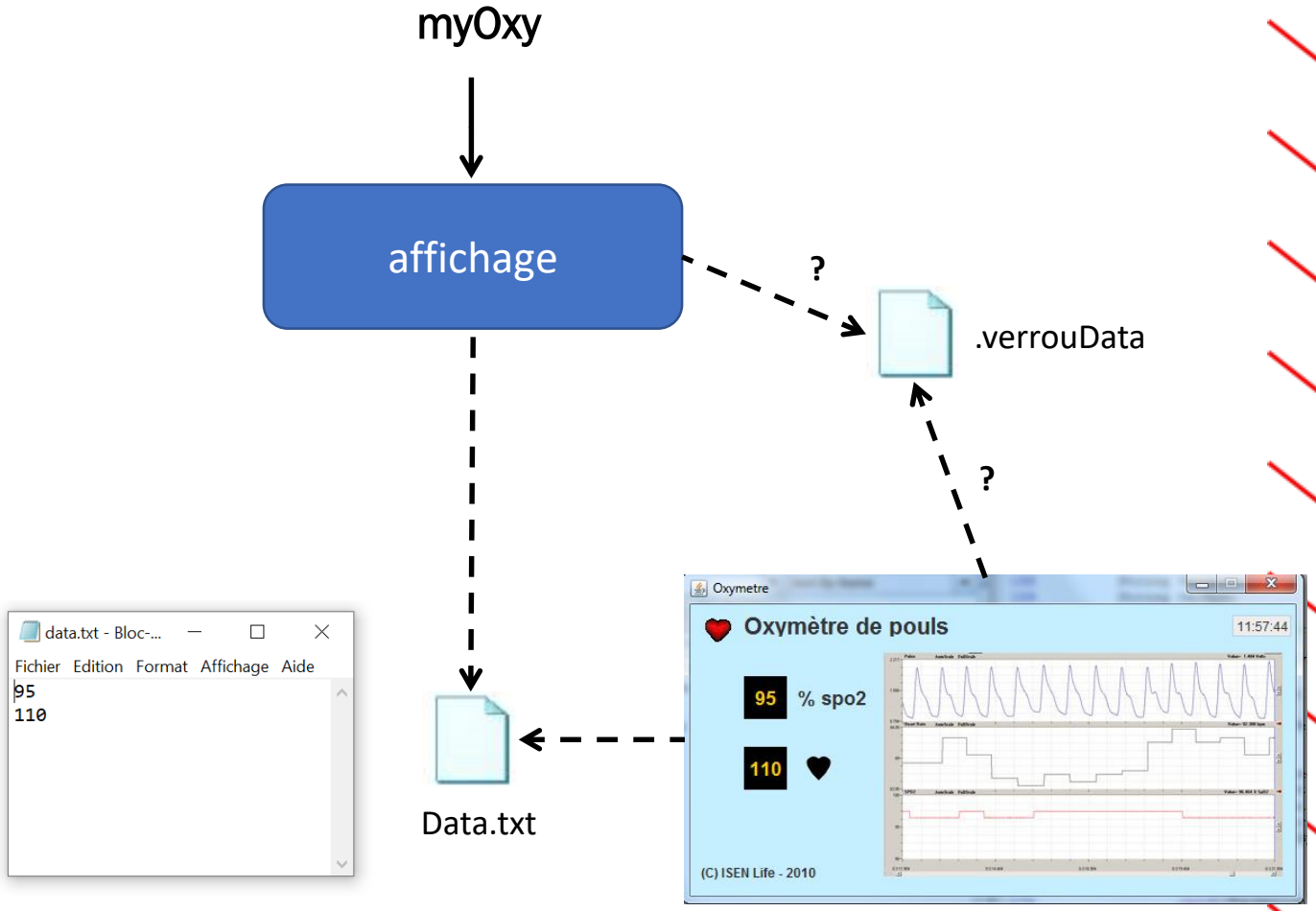
- Points essentiels
  - Calcul de la fréquence de l'onde de pouls (sur AC_R et AC_{IR})
- Points critiques
  - Calcul de la fréquence cardiaque au "fil de l'eau"



$$F = \frac{1}{T}$$

Affichage

- Points essentiels
  - Ecriture des informations dans le fichier Data.txt
- Points critiques
  - Exclusion mutuelle à l'aide d'un verrou



Affichage

- Test existence d'un fichier

```
#include <unistd.h>

if( access( ".verrouData", F_OK ) != -1 )
{
        // Fichier existe
}else{
        // Fichier n'existe pas
}
```

- Effacer un fichier

```
remove(".verrouData")
```

Lecture

- Points essentiels
  - Lire chacun des champs
  - Recentrer les composantes AC autour de 0
- Points critiques
  - Etablir la synchronisation afin de récupérer les valeurs
  - Driver FTDI

Du fait du confinement due à l'épidémie COVID-19, l'accès à la carte électronique est rendu impossible. La lecture sur le port USB sera simulée. A cet effet nous avons créé un fichier nommé `record1_bin.dat` contenant les données à lire sous format binaire.

Lecture

- Détail de la trame

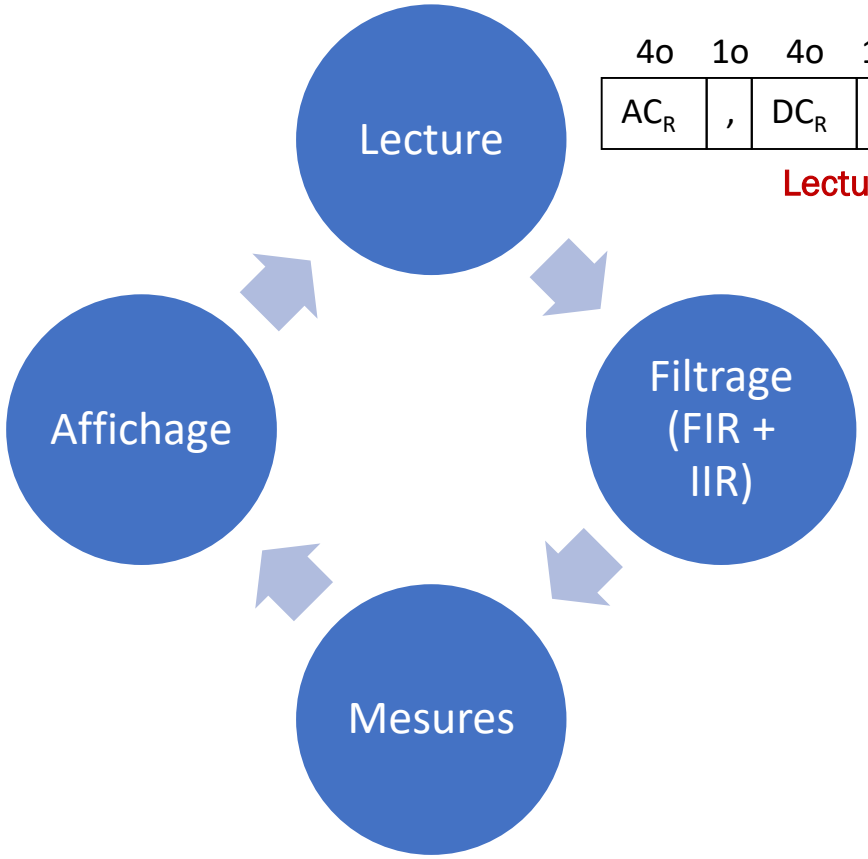
21 octets (rafraichissement toutes 2 ms)

Taille (octets)	4	1	4	1	4	1	4	1	1
Champ	AC_R	,	DC_R	,	AC_{IR}	,	DC_{IR}	LF	CR

Architecture

Organisation du programme

Le traitement se fait au fil de l'eau !

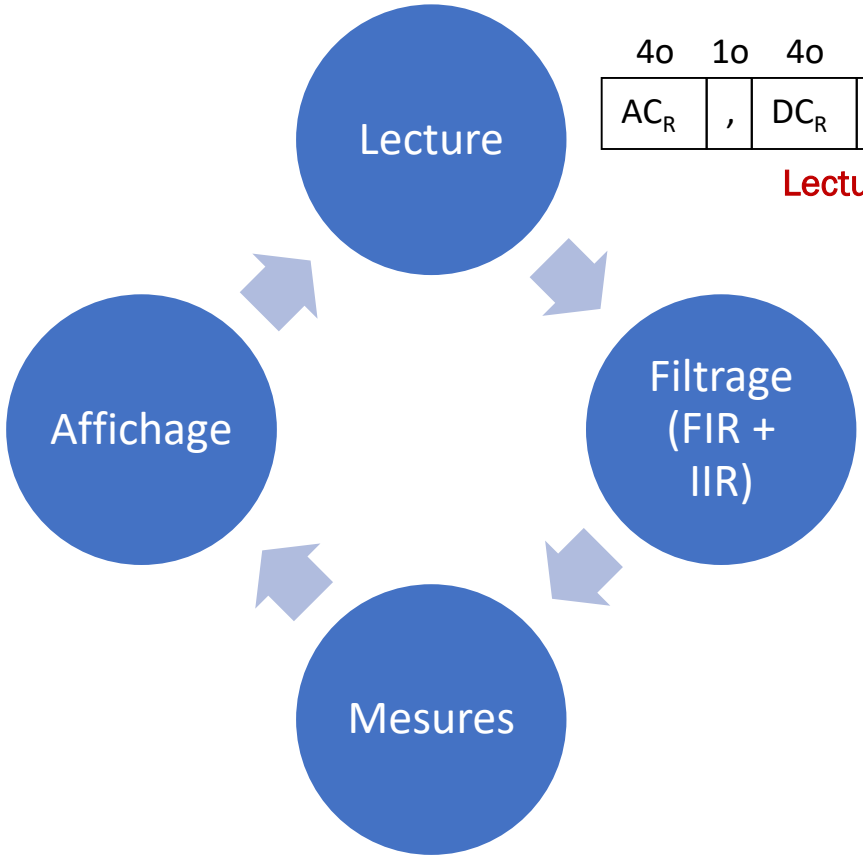


4o	1o	4o	1o	4o	1o	4o	1o	1o
AC_R	,	DC_R	,	AC_{IR}	,	DC_{IR}	LF	CR

Lecture d'une mesure

Organisation du programme

Le traitement se fait au fil de l'eau !



4o	1o	4o	1o	4o	1o	4o	1o	1o
AC_R	,	DC_R	,	AC_{IR}	,	DC_{IR}	LF	CR

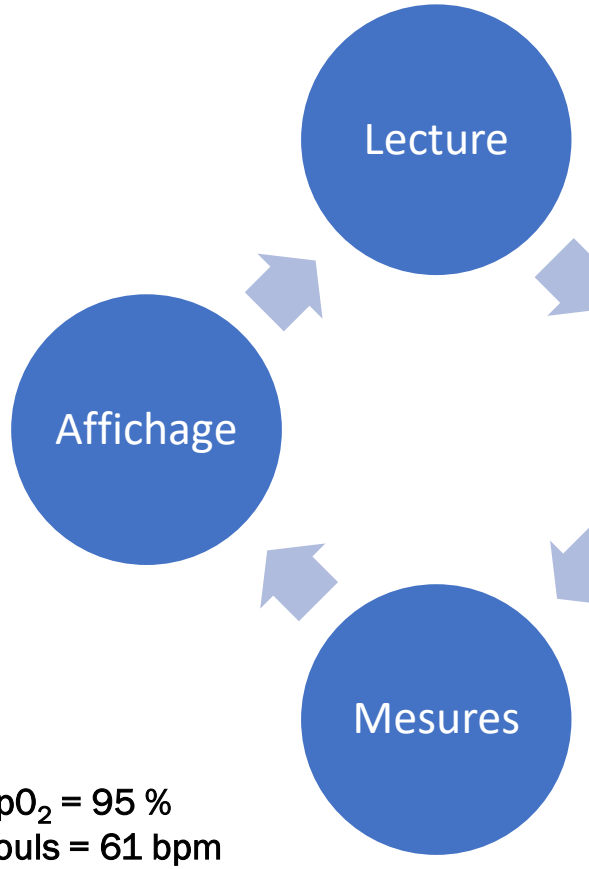
Lecture d'une mesure

$AC_R = 10$
$DC_R = 2030$
$AC_{IR} = 30$
$DC_{IR} = 2030$

Filtrage d'une mesure

Organisation du programme

Le traitement se fait au fil de l'eau !



4o	1o	4o	1o	4o	1o	4o	1o	1o
AC_R	,	DC_R	,	AC_{IR}	,	DC_{IR}	LF	CR

Lecture d'une mesure

$AC_R = 10$
$DC_R = 2030$
$AC_{IR} = 30$
$DC_{IR} = 2030$

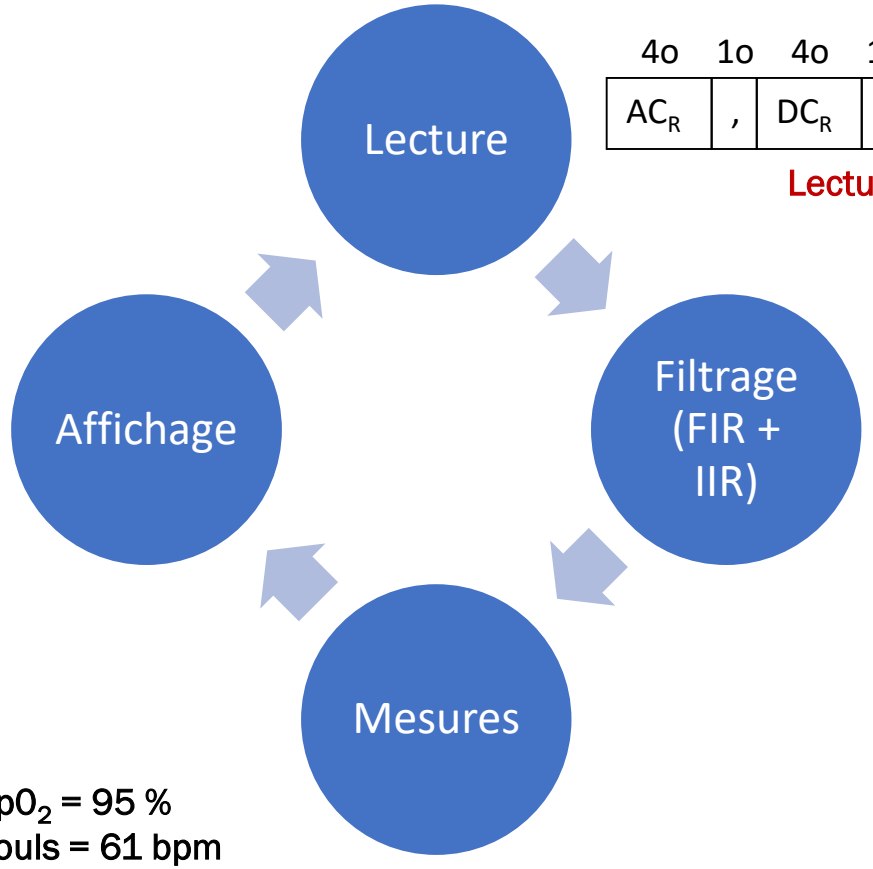
Filtrage d'une mesure

$SpO_2 = 95\ \%$
Pouls = 61 bpm

Calcul de Sp02 et Pouls

Organisation du programme

Le traitement se fait au fil de l'eau !



4o	1o	4o	1o	4o	1o	4o	1o	1o
AC_R	,	DC_R	,	AC_{IR}	,	DC_{IR}	LF	CR

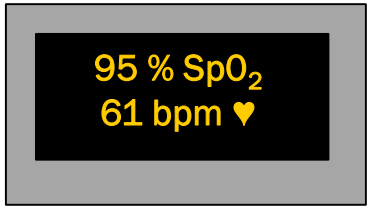
Lecture d'une mesure

AC_R = 10
DC_R = 2030
AC_{IR} = 30
DC_{IR} = 2030

Filtrage d'une mesure

SpO_2 = 95 %
Pouls = 61 bpm

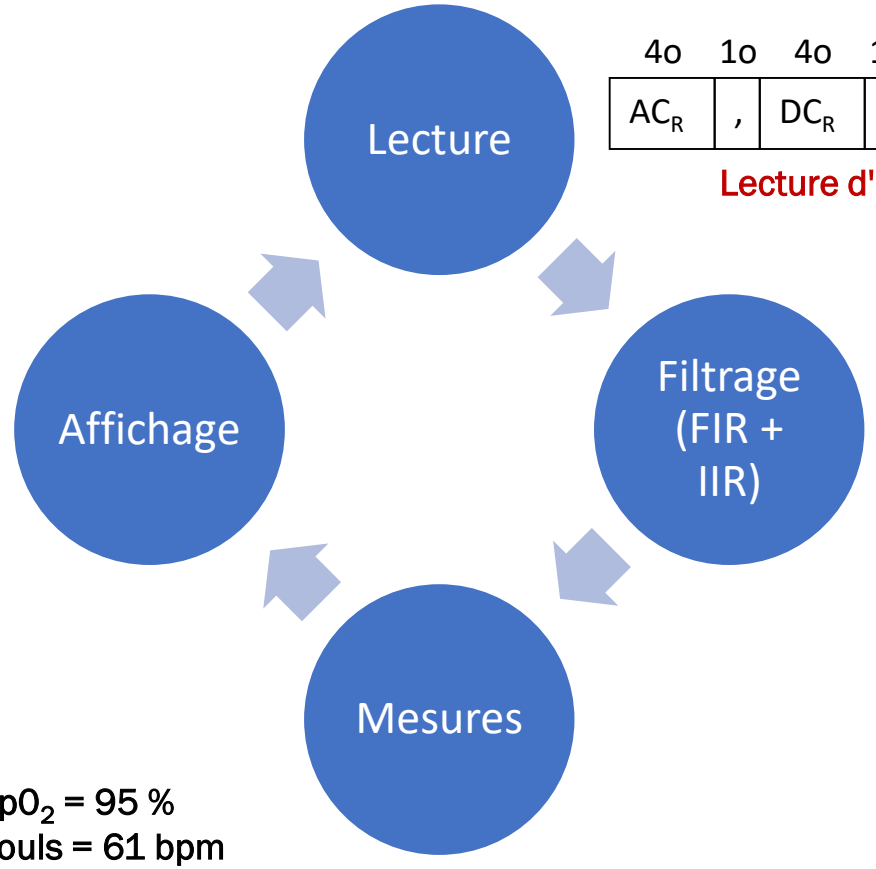
Calcul de Sp02 et Pouls



Affichage du résultat

Organisation du programme

Le traitement se fait au fil de l'eau !



4o	1o	4o	1o	4o	1o	4o	1o	1o
AC_R	,	DC_R	,	AC_{IR}	,	DC_{IR}	LF	CR

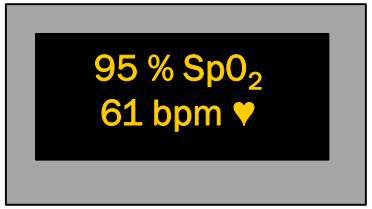
Lecture d'une nouvelle mesure

$AC_R = 10$
$DC_R = 2030$
$AC_{IR} = 30$
$DC_{IR} = 2030$

Filtrage d'une mesure

$SpO_2 = 95\ \%$
Pouls = 61 bpm

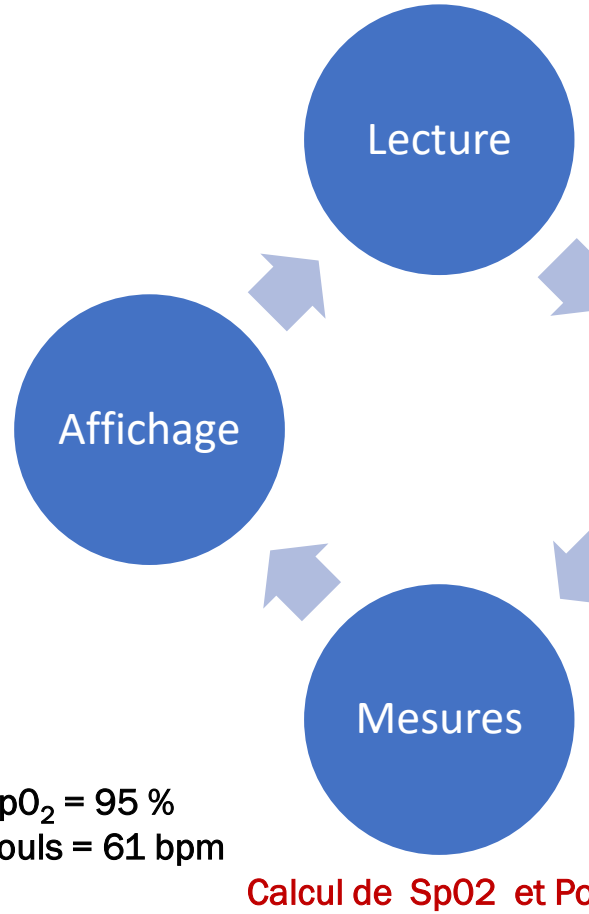
Calcul de Sp02 et Pouls



Affichage du résultat

Organisation du programme

Le traitement se fait au fil de l'eau !



4o	1o	4o	1o	4o	1o	4o	1o	1o
AC_R	,	DC_R	,	AC_{IR}	,	DC_{IR}	LF	CR

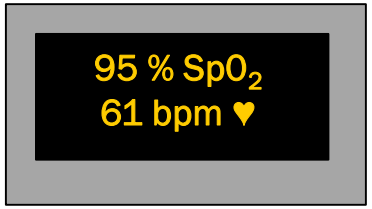
Lecture d'une nouvelle mesure

AC_R = 12
DC_R = 2030
AC_{IR} = 45
DC_{IR} = 2030

Filtrage d'une nouvelle mesure

SpO_2 = 95 %
Pouls = 61 bpm

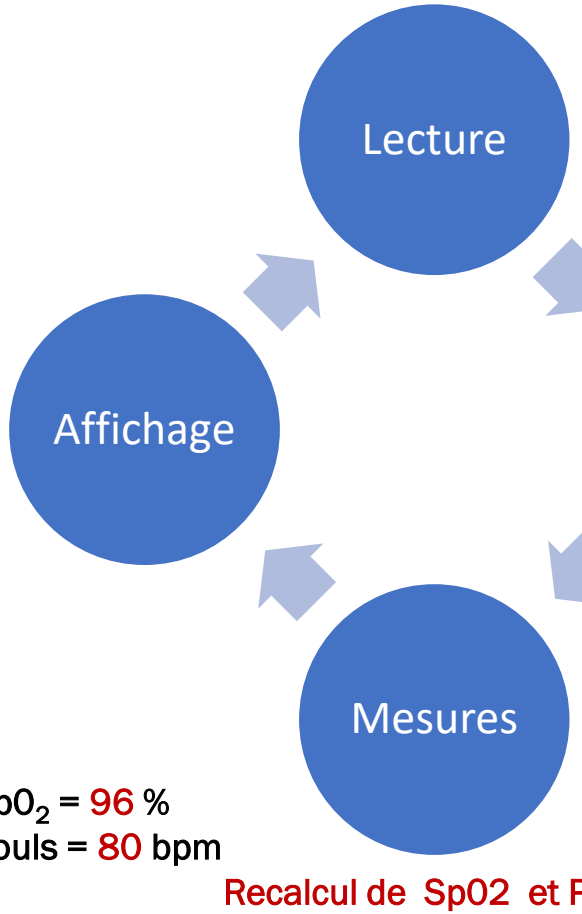
Calcul de Sp02 et Pouls



Affichage du résultat

Organisation du programme

Le traitement se fait au fil de l'eau !



4o	1o	4o	1o	4o	1o	4o	1o	1o
AC_R	,	DC_R	,	AC_{IR}	,	DC_{IR}	LF	CR

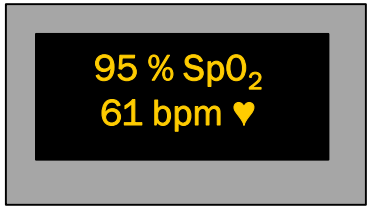
Lecture d'une nouvelle mesure

AC_R = 12
DC_R = 2030
AC_{IR} = 45
DC_{IR} = 2030

Filtrage d'une nouvelle mesure

SpO_2 = 96 %
Pouls = 80 bpm

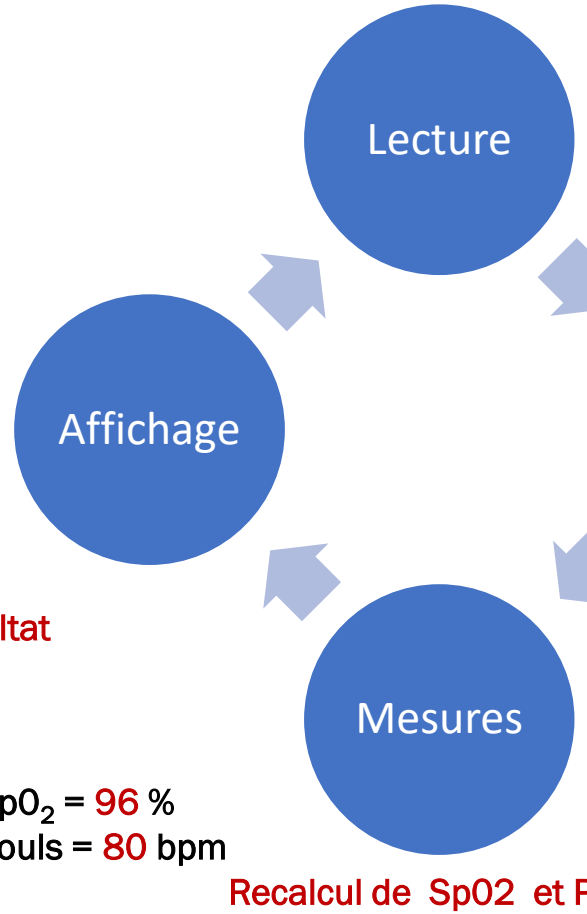
Recalcul de Sp02 et Pouls



Affichage du résultat

Organisation du programme

Le traitement se fait au fil de l'eau !



4o	1o	4o	1o	4o	1o	4o	1o	1o
AC_R	,	DC_R	,	AC_{IR}	,	DC_{IR}	LF	CR

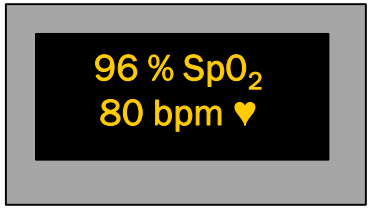
Lecture d'une nouvelle mesure

AC_R = 12
DC_R = 2030
AC_{IR} = 45
DC_{IR} = 2030

Filtrage d'une nouvelle mesure

SpO_2 = 96 %
Pouls = 80 bpm

Recalcul de Sp02 et Pouls

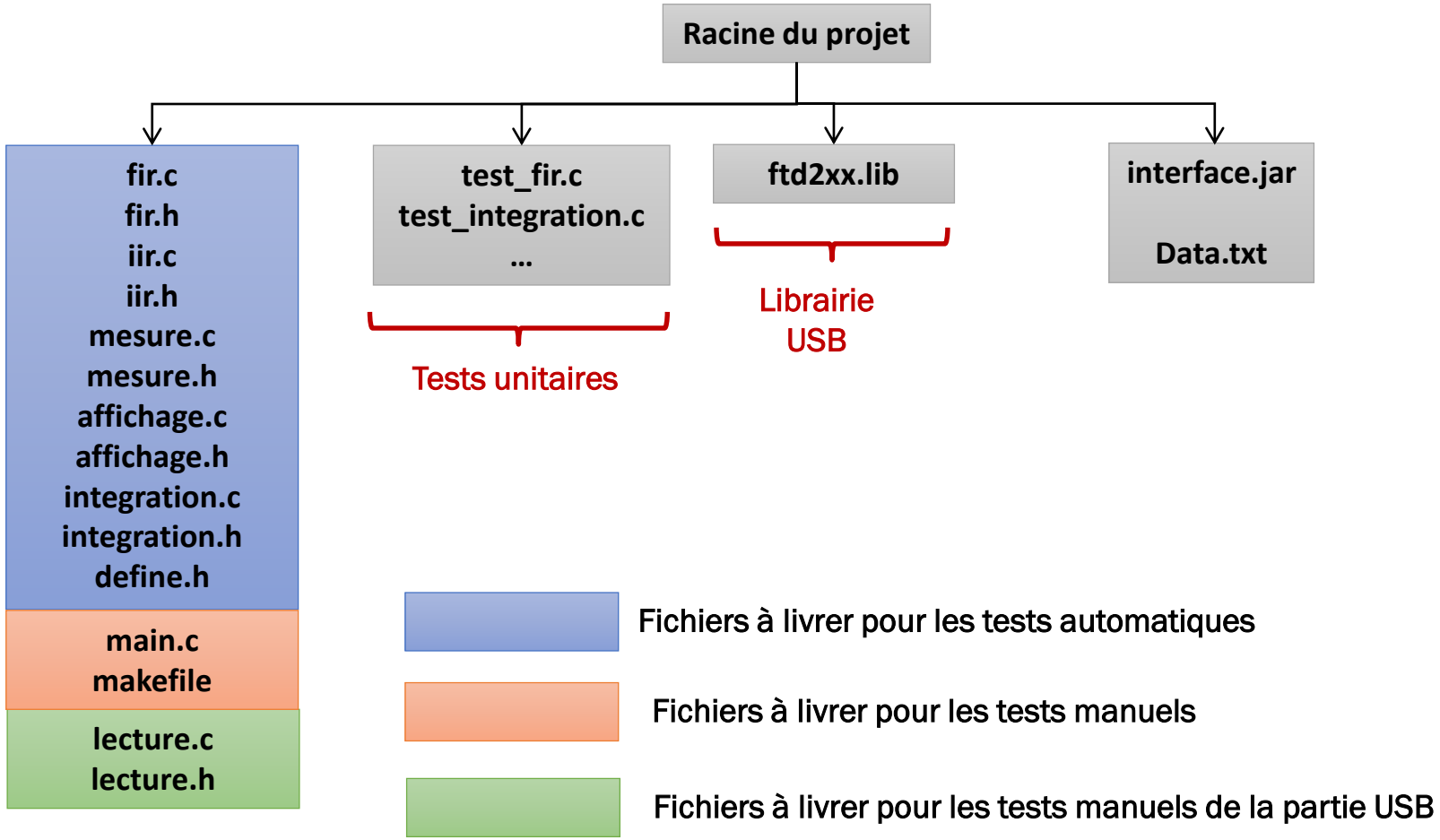


Affichage du nouveau résultat

Convention de nommage

Bloc	Nom fonction	Nom fichier source	Nom fichier header
Filtrage FIR	`fir`	`fir.c`	`fir.h`
Filtrage IIR	`iir`	`iir.c`	`iir.h`
Mesure	`mesure`	`mesure.c`	`mesure.h`
Affichage	`affichage`	`affichage.c`	`affichage.h`
Intégration	`integration`	`integration.c`	`integration.h`
Lecture	`lecture`	`lecture.c`	`lecture.h`
Définition globales			`define.h`

Arborescence



Exemple de programme

main.c

```
int main(){
        int etat=0;
        absorp myAbsorp;
        oxy myOxy;
        param_fir* myFIR = init_fir(…); // init FIR
        param_iir* myIIR = init_iir(…); // init IIR
        param_mesure* myMes = init_mesure(…) // init mesure
        FILE* myFile = initFichier("record1.dat");
        do{
                myAbsorp = lireFichier(myFile,&etat);
                myAbsorp = fir(myAbsorp,myFIR);
                myAbsorp = iir(myAbsorp,myIIR);
                myOxy = mesure(myAbsorp,myMes);
                affichage(myOxy);
        }while( etat != EOF );
        finFichier(myFile);
        fin_mesure(myMes);
        fin_iir(myIIR);
        fin_fir(myFIR) ;
        return EXIT_SUCCESS;
}
```

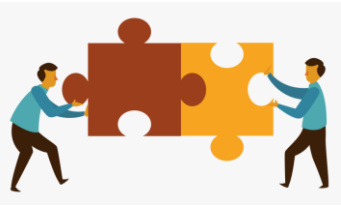
define.h

```
#ifndef DEFINE_H
#define DEFINE_H

typedef struct{
          float acr;  /*!< AC R */
          float dcr;  /*!< DC R */
          float acir;  /*!< AC IR */
          float dcir;  /*!< DC IR */
} absorp;
typedef struct{
          int spo2;  /*!< SPO2 */
          int pouls;  /*!< Pouls */
} oxy;

#endif
```

Déroulé du projet

Déroulé du projet

4 phases



Lundi matin	Lundi après-midi	Mardi matin	Mardi après-midi	Mercredi matin	Mercredi après-midi	Vendredi matin	Vendredi après-midi
Analyse	Réalisation				Intégration		Recette

- ✓Analyse de chaque bloc
- ✓Architecture du programme
- ✓Définition des Types de données

- ✓Code source de chaque bloc
- ✓Tests unitaires
- ✓Makefile

- ✓Code source programme principal
- ✓Test intégration avec le simulateur ou avec la carte
- ✓Validation avec l'interface java

- ✓QCM
- ✓Validation automatique sur l'intranet
- ✓Validation manuelle par l'enseignant sur PC enseignant
- ✓Qualimétrie

Au jour le jour

- Travail en binôme
  - Chaque étudiant connait l'ensemble du projet
  - Attention à bien se répartir le travail

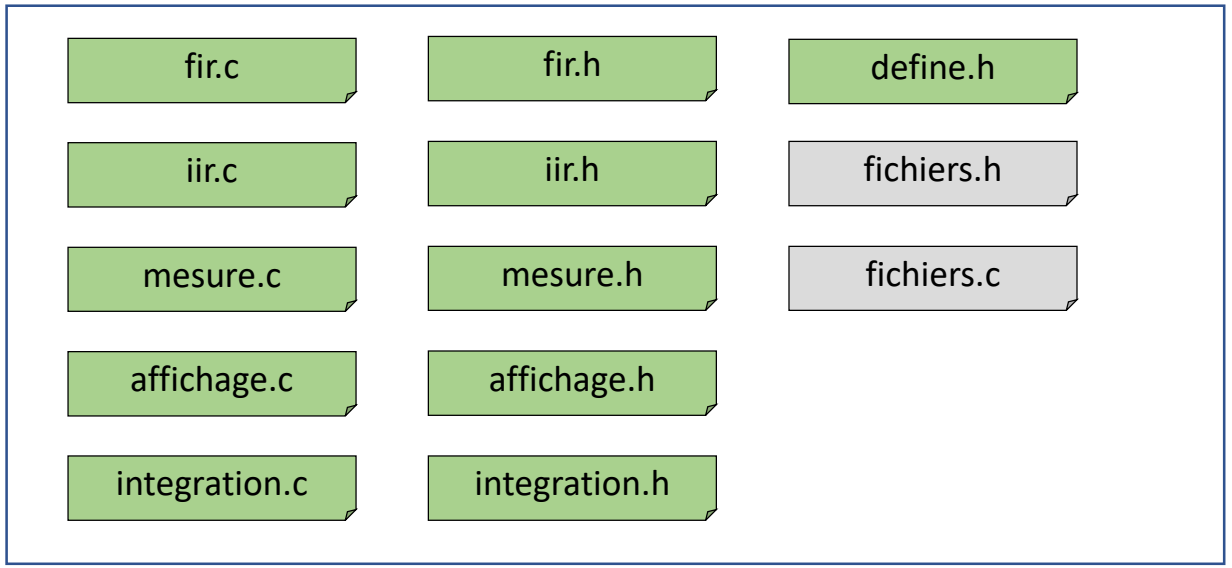
- Ressources externes
  - Tous les documents sont autorisés…
    - … mais les informations nécessaires au projet sont fournies !!!
  - Attention à utiliser avec une grande précaution tous documents venant du web (site de vulgarisation, forum, code d'autrui,…)

- Livraison de code ou de document
  - Ne pas attendre la dernière minute pour poster le livrable
    - Préparer des livrables intermédiaires
    - Sauvegarder régulièrement vos données

Recette

Ce qu'il faut rendre (sauf CIR3)

- Sources pour tests automatiques



tests_auto_xx_yy.zip

Pour vous aider, une coquille vide comprenant l'architecture imposée est disponible sur l'ENT

Fichiers à modifier

Fichiers à ne pas modifier

- Sources pour tests manuels
  - Fichiers précédents
  - main.c
  - Makefile
  - Si USB: lecture.c et lecture.h

tests_manuels_xx_yy.zip

Ce qu'il faut rendre (CIR3)

- Sources pour tests automatiques

fir.c	fir.h	define.h
iir.c	iir.h	fichiers.h
mesure.c	mesure.h	fichiers.c
affichage.c	affichage.h	lecture.c
integration.c	integration.h	lecture.h

tests_auto_xx_yy.zip

Pour vous aider, une coquille vide comprenant l'architecture imposée est disponible sur l'ENT

Fichiers à modifier

Fichiers à ne pas modifier

- Sources pour tests manuels
  - Fichiers précédents
  - main.c
  - Makefile

tests_manuels_xx_yy.zip

Le code source

- Chaque bloc :
  - Un fichier
  - Une fonction

Les interfaces de la fonction principale de chaque bloc sont imposées afin de permettre un validation automatique lors de la recette. Les prototypes vous seront fournies en début de réalisation !

- Programme principal :
  - Permet de tester chaque bloc de façon indépendante
  - Permet de tester le programme complet
    - Avec le simulateur
    - Avec la carte électronique
- Makefile
  - Permet de compiler automatiquement votre code source
  - Obligatoire pour la partie USB (sinon pénalités)
- Revue de code
  - Lors de la recette, il vous sera demandé d'expliquer une ou plusieurs portions de votre code

Barème final (indicatif)

	Coefficient
Recette programme	6
Qualimétrie code	2
QCM	2
Total	**10**

Barème recette (indicatif)

#	Nom d test	Type	Barème A3 hors CI3	Barème CIR3
1	FIR	Automatique	3,5	3
2	IIR	Automatique	3	2,5
3	Mesure SPO2	Automatique	2,5	2
4	Mesure Pouls	Automatique	2,5	2
5	Affichage	Automatique	3	2,5
6	Intégration	Automatique	3,5	3
7	Programme global simulation	Manuel	2	2
8	Programme global USB	Manuel	Bonus	3
Total			20	20

Tests automatiques

General informations

Project: Oxymetrie
Student: eleven3 Eleve

Download Src

Details

Function	Tests
affichage	0 / 2
fir	0 / 1
iir	0 / 1
intergration	0 / 2
lecture	3 / 3
mesure	0 / 2

Les ressources

Le materiel

- Votre PC portable
  - Analyse
  - Réalisation
  - Intégration

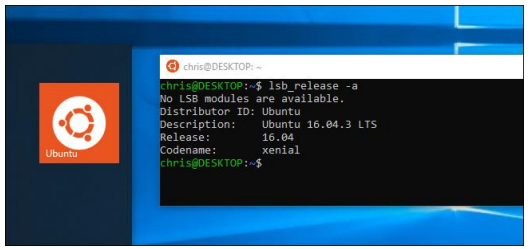
Casque / écouteur audio déconseillés (consignes non appliqués = pénalité) !

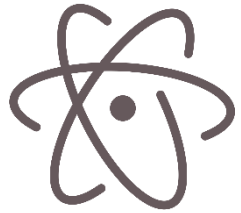
Boissons et nourriture interdites !

- Carte électronique
  - Validation de la partie USB

Outils de développement

- Noyau de compilation
  - Gcc sous MinGW (minimalist GNU for Windows)
  - Bash Windows
  - Linux Virtual Box
  - Jupyterhub
  - Linux natif
- Editeur
  - Notepad++
  - Sublime Text
  - Atom
  - Visual studio code
  - …

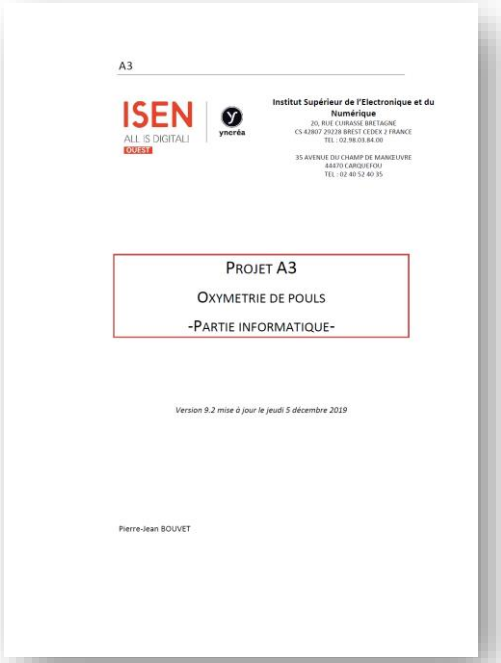




Documentation

- Support de présentation

- Description détaillée



- Guide de programmation FTDI

ISEN
ALL IS DIGITAL!

yncréa

MERCI

Des questions ?