

TERMINALE S



Informatique et Sciences du numérique

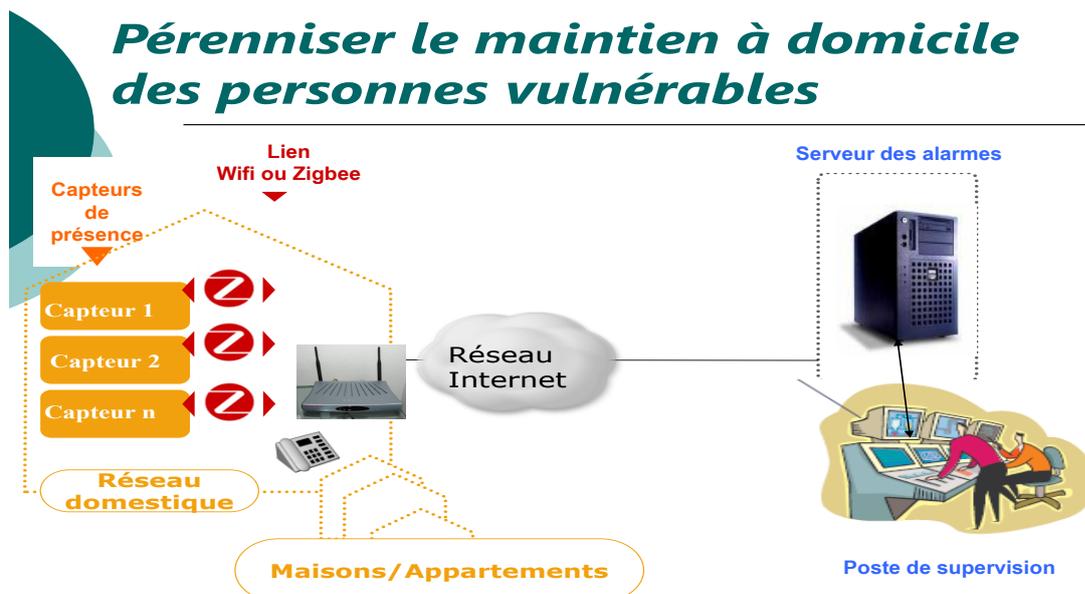
MINI-PROJET ISN

Pérenniser le maintien à domicile des personnes vulnérables

1. Contexte de réalisation

Projet proposé et suivi par :	M : LAZAAR Abdelkrim professeur M : MAILLE Vincent professeur
Constitution de l'équipe de développement :	Elèves de groupe 1 et groupe 2

2. Présentation du projet



L'objectif du projet est de contribuer à pérenniser le maintien à domicile des personnes vulnérables en leur offrant un environnement sûr.

Dans le cadre d'un service type téléalarme pour personne âgée seule à domicile, l'objectif principal est de déterminer si elle est en état de détresse.

Des capteurs de présence en Zigbee seront installés au salon, chambres, couloir, salle de bain, et cuisine.

En fonctionnement nominal, une personne qui se trouve dans une des pièces "surveillées", est considérée par défaut en état "normal". Si aucun des capteurs ne détecte la personne, après un **certain temps**, on envoie une alarme au poste de supervision. Ce dernier appelle la personne au téléphone pour vérifier si elle est dans un état "normal" ou dans un état de détresse (une chute dans un couloir par exemple).

Le logiciel devra utiliser les informations de détection ou non détection des capteurs, associées à une gestion du temps pour affiner son analyse avant d'envoyer une alarme.

3. Documents et moyens technologiques mis à disposition

- Documentation sur les capteurs Zigbee : <http://www.zigbee.org/>
- Documentation sur le protocole utilisé par les capteurs : voir annexe1
- **Décès de personnes âgées résidant à domicile pendant la vague de chaleur d'août 2003 : une étude de l'InVS identifie les facteurs de risque.**
http://www.invs.sante.fr/presse/2004/communiques/chaleur2003_170904/index.html
- L'isolement des personnes âgées ne cesse d'augmenter dans la région : Lire les articles de presse sur Internet.
- Les problématiques liées aux questions de nature sociétale :
- Apport de l'Informatique et Sciences numérique dans l'aide aux personnes âgées (vague de chaleur d'août 2003, l'étude de l'Institut de veille sanitaire sur les facteurs de risque)

4. Répartition des tâches

N.B : Le poste du professeur sera équipé d'un simulateur logiciel qui envoie les trames compatibles avec celles des capteurs Zigbee.

Réf.	Fonction	Description
F1	Acquisitions des états des capteurs Zigbee (ou Wifi)	Effectuer la récupération des trames des capteurs Zigbee. Calculer le checksum pour vérifier l'intégrité de la trame avant de la traiter.
F2	Exploitation des grandeurs physiques	Extraire et exploiter les informations des capteurs Zigbee . Développer une stratégie pour déterminer si la personne est en détresse

5. Planification temporelle prévisionnelle

Repère de la tâche	Description de la tâche	1	2	3	4	5	6
T1	Représentation de l'information : <ul style="list-style-type: none"> Le codage numérique du texte : la table ASCII l'Unicode l'UTF-8 Opérations booléennes <ul style="list-style-type: none"> Calculer le checksum pour vérifier l'intégrité de la trame avant de la traiter. 						
T2	<ul style="list-style-type: none"> Effectuer la récupération des trames des capteurs Zigbee. Calculer le checksum pour vérifier l'intégrité de la trame avant de la traiter. 						
T3	<ul style="list-style-type: none"> Extraire et exploiter les informations des capteurs Zigbee. Développer une stratégie pour déterminer si la personne est en détresse 						

6. Evaluation pour l'épreuve du PROJET

Atteintes des objectifs du point de vue client : A la fin du projet, on observera différents critères qui témoigneront de l'atteinte des objectifs fixés du point de vue du client :

- 1) Réception de l'état des capteurs par le poste de gestion des alarmes.
- 2) Détection d'un état de détresse selon les divers scénarios proposés

21/03 : SEANCES 1 ET 2 (4 HEURES) : LE CODAGE NUMERIQUE DU TEXTE

1 Utilisation de la table ASCII

1.1) À l'aide de la table ASCII, coder en binaire la phrase suivante : « L'an qui vient ! ».

1.2) Voici maintenant une exclamation codée en binaire :

01000010 01110010 01100001 01110110 01101111 00101100

Retrouver cette exclamation !

1.3) Peut-on coder en binaire la phrase suivante à l'aide de la table ASCII ? (Justifier la réponse)

« Un âne est-il passé par là? »

2) Conversion

On commence par se connecter au site suivant :

https://wiki.inria.fr/sciencinfolycee/Convertisseur_texte/binaire/hexa_en_ligne

Voici le code binaire d'un texte : (téléchargeable sur le site ISN : texte.txt) :

```
01000010 01110010 01100001 01110110 01101111 00101100 00100000 01110100 01110101 00100000
01100001 01110011 00100000 01110000 01110010 01100101 01110011 01110001 01110101 01100101
00100000 01110100 01101111 01110101 01110100 00100000 01110100 01110010 01101111 01110101
01110110 11101001 00101110 00101110 00101110
```

À l'aide du logiciel fourni sur le site, retrouver le texte contenu dans le code.

3) PROGRAMMATION

Le langage Php propose plusieurs fonctions de conversion :

- [dechex\(\)](#) - Convertit de décimal en hexadécimal
- [hexdec\(\)](#) — Convertit de hexadécimal en décimal
- [bindec\(\)](#) - Convertit de binaire en décimal
- [octdec\(\)](#) - Conversion d'octal en décimal
- [base_convert\(\)](#) - Convertit un nombre entre des bases arbitraires

Ecrire un programme qui offre à l'utilisateur la possibilité de saisir un nombre (ou une Lettre) et d'afficher sa valeur dans les autres bases.

REMARQUE : Les mots [[TEXTE](#)] [[ASCII Décimal](#)] [[Binaire](#)] [[Hexadécimal](#)] sont des liens vers les pages suivantes :

- [[TEXTE](#)] → <http://fr.wikipedia.org/wiki/Ascii>
- [[ASCII Décimal](#)] → <http://fr.wikipedia.org/wiki/ASCII>
- [[Binaire](#)] → [http://fr.wikipedia.org/wiki/Code binaire](http://fr.wikipedia.org/wiki/Code_binaire)
- [[Hexadécimal](#)] → <http://fr.wikipedia.org/wiki/Hexad%C3%A9cimal>

Exemple d'IHM

[TEXTE]	[ASCII Décimal]	[Binaire]	[Hexadécimal]
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="button" value="Décoder"/>			

Version 1 : On utilise, deux pages : une pour la saisie, une pour l'affichage du résultat

Version 2 : On utilise une seule page, le résultat apparaîtra alors dans les champs de saisie non renseignés (on pourra utiliser l'instruction [isset\(\)](#) permettant de savoir si une variable existe ou non.

4) Révisions sur les tableaux

⇒ Les tableaux : le php propose une structure de tableau. Il s'agit d'un type d'objet permettant de placer dans une seule variable (presque) autant d'objets que l'on souhaite.

```
<?php
    $valeur = array(1,8,7,9,10);

    echo $valeur[3];
?>
```

⇒ Exemple : l'exemple suivant enregistre dans le tableau \$valeur la série de valeurs 1, 8, 7, 9, 10 puis affiche la valeur d'indice 3, c'est à dire 9 (car la numérotation commence à 0).

⇒ Remarque : dans un tableau, on peut stocker tout type d'objets, l'exemple ci-contre affiche les 4 premiers jours de la semaine.

```
<?php
    $jours = array("Lundi", "Mardi", "Mercredi", "Jeudi", "Vendredi", "Samedi", "Dimanche");

    for ($j=0;$j<4;$j++){
        echo $jours[$j];
        echo "<BR>";
    }
?>
```

Exercice 4.1 : Dans cet exercice, on considère un tableau contenant un relevé journalier de la température d'une pièce de l'appartement de la personne âgée. Il y a un relevé par heure, soit 24 valeurs (de 0h00 à 23h00)

17.1	17.1	17.3	17	17.4	17.4	17.4	17.4	18	18.5	18.8	19.3
19.5	19.5	19.3	19.4	19.3	19.5	19.5	19.2	18.7	18.5	18.1	17.7

1. Afficher les 24 températures et l'heure du relevé
2. Afficher la température journalière moyenne
3. Afficher la température la plus élevée, la plus basse et l'étendue de la série de valeurs.

Exercice 4.2 : Afficher parmi les 24 températures, celles supérieures à 18°C

Exercice 4.3 : Afficher parmi les 24 températures le nombre de fois où la température augmenté d'au moins un demi degré en une heure.

⇒ Quelques fonctions sur les tableaux en php :

sort(\$tableau) : Renvoie un tableau ordonné (ne modifie pas \$tableau) sizeof(\$tableau) ou count(\$tableau) : renvoie le nombre d'éléments du tableau
--

5) Opérations booléennes

Objectif : Calculer le checksum pour vérifier l'intégrité de la trame avant de la traiter.

Les capteurs: Nous avons à notre disposition différents capteurs (Zigbee) :

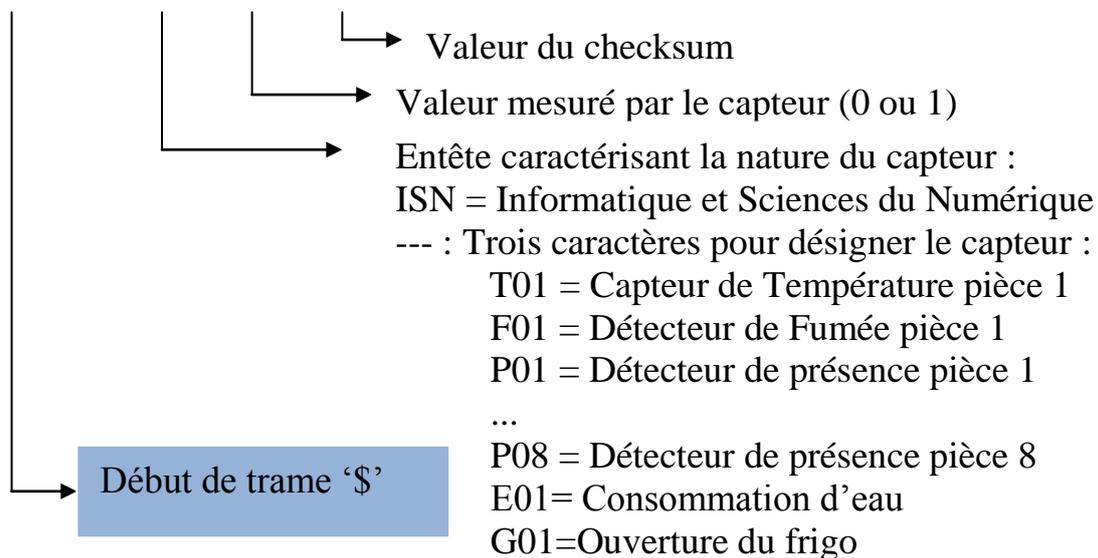
- Des capteurs de température, des détecteurs de fumée, détecteurs de présence, etc....
- Chaque capteur envoie une trame d'information en continue.

Poste de gestion des alarmes



Une trame d'un capteur est composée ainsi :

\$ISN--- , x, *00<CR> <LF>



Pour s'assurer de l'intégrité de la trame reçue une clef est envoyée à la fin de la chaîne de caractère (comme pour un numéro de sécurité sociale, de n° de compte de banque, un code barre sur une livre,)

Méthode de calcul de checksum : Description of approved Sentences

“\$”	HEX24 – start of sentence
<Adresse field>	TALKER identifier and sentence formatter
[“,”<data field>]	Zero or more data fields
⋮	
[“,”<data field>,”]	
“*”<checksum field>	Checksum field
<CR><LF>	HEX 0D 0A- End of sentence

Checksum field: The absolute value calculates by exclusive-OR'ing the 8 data bits of each character in the sentence, between, but excluding “\$” and “*”. The hexadecimal value of the most significant and least significant 4 bits of the result are converted to two ASCII characters (0-9,A-F (upper case)) for transmission. The most significant character is transmitted first. The checksum field is required in all transmitted sentences.

Exemples des trames :

"\$ISNP01,0,*35\r\n"	"\$ISNE01,0,*20\r\n"
"\$ISNG01,0,*22\r\n"	"\$ISNT01,1,*FD\r\n"
"\$ISNF01,1,*22\r\n"	"\$ISNT01,1,*30\r\n"
"\$ISNP01,1,*34\r\n"	"\$ISNE01,0,*21\r\n"
"\$ISNP01,1,*3E\r\n"	...

Exercice 5.1 Ecrire un programme qui calcule le Checksum d'une trame saisie au clavier (ex : "\$ISNG01,0,*22\r\n") et le compare avec celui de la trame saisie.

Dans notre cas, un système électronique récupère les trames de chaque capteur, les assemble et envoie en continue les trames bout à bout dans cet ordre :

Trame ISNT01 = Capteur de Température pièce 1
Trame ISNF01 = Détecteur de Fumée pièce 1
Trame ISNP01 = Détecteur de présence pièce 1
...
Trame ISNP08 = Détecteur de présence pièce 8
Trame ISNE01= Consommation d'eau
Trame ISNG01=Ouverture du frigo

Exercice 5.2 : On a reçu cette chaîne de 600 octets (trame600.txt) : Extraire un cycle complet des 12 trames provenant des différents capteurs. Afficher alors en français les informations sur une page web.

```
F01,0,*23\r\n$ISNP01,0,*35\r\n$ISNP02,0,*36\r\n$ISNP03,0,*37\r\n$ISNP04,0,*30\r\n$ISNP05,0,*31\r\n$ISNP06,0,*32\r\n$ISNP07,0,*33\r\n$ISNP08,0,*3C\r\n$ISNE01,0,*20\r\n$ISNG01,0,*22\r\n$ISNT01,25,*06\r\n$ISNF01,0,*23\r\n$ISNP01,0,*35\r\n$ISNP02,0,*36\r\n$ISNP03,0,*37\r\n$ISNP04,0,*30\r\n$ISNP05,0,*31\r\n$ISNP06,0,*32\r\n$ISNP07,0,*33\r\n$ISNP08,0,*3C\r\n$ISNE01,0,*20\r\n$ISNG01,0,*22\r\n$ISNT01,25,*06\r\n$ISNF01,0,*23\r\n$ISNP01,0,*35\r\n$ISNP02,0,*36\r\n$ISNP03,0,*37\r\n$ISNP04,0,*30\r\n$ISNP05,0,*31\r\n$ISNP06,0,*32\r\n$ISNP07,0,*33\r\n$ISNP08,0,*3C\r\n$ISNE01,0,*20\r\n$ISNG01,0,*22\r\n$ISNT01,25,*06\r\n$ISNF01,0,*23\r\n$ISNP01,0,*35\r\n$ISNP02,0,*36\r\n$ISNP03,0,*37\r\n$
```

DM3 : LES COULEURS

à rendre pour le mercredi 28/11/2012

Consignes : Transfert de votre programme sur le serveur : A l'aide d'un logiciel de transfert de fichier (FTP), télécharger votre code source sur le serveur : Créer dans votre répertoire un répertoire DM3, et transférer vos codes. Connectez-vous au serveur et tester le bon fonctionnement de votre programme. Information pour se connecter au serveur FTP :

Hôte = 195.221.154.52

Utilisateur = ISN Mot de passe = USER

Port = 4451

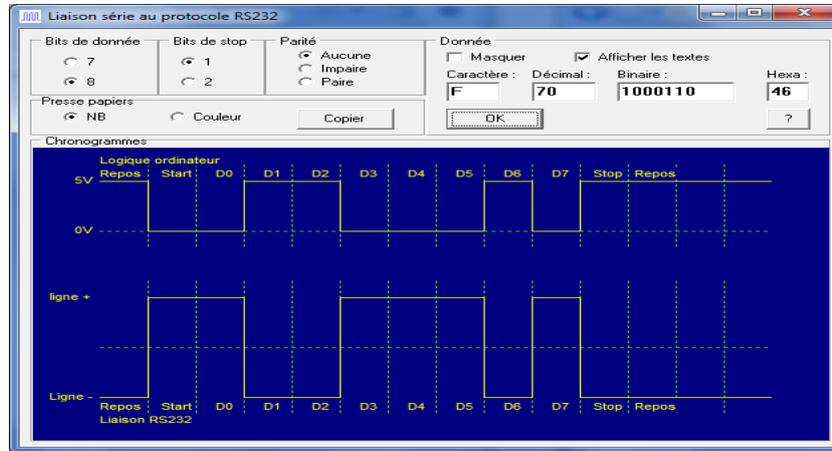
Pour tester votre programme : <http://195.221.154.52:4450/>

En cas d'impossibilité de vous connecter au serveur, il faut envoyer vos programmes et l'exposé par mail à l'adresse suivante : tsirisbranly@ac-amiens.fr (M. Lazaar) ou vincent.maille@ac-amiens.fr (M. MAILLE)

28/11 : SEANCE 3 - COMMUNICATION PAR TRANSMISSION SERIE RS232

1) Communication entre deux équipements

1. Brancher le câble série, lancer le logiciel Tera Term et établir un dialogue entre deux ordinateurs.
2. Lancer le logiciel ChronoRS232.exe



Relever les chronogrammes de la lettre F dans les cas suivants :

1 ^{er} cas : Bits de données = 8 Bit de stop = 1 Parité = Aucune	2 ^{ème} cas: Bits de données = 8 Bit de stop = 1 Parité = impaire
3 ^{ème} cas : Bits de données = 8 Bit de stop = 1 Parité = paire	4 ^{ème} cas : Bits de données = 8 Bit de stop = 2 Parité = Aucune , paire et impaire

Que constatez-vous ? Discutez en particulier l'utilité du paramètre parité.

2) Acquisition et Schéma de câblage RS232

1. Rechercher sur Internet, des ressources sur le schéma de câblage RS232 et la configuration d'une liaison série RS232.
2. Lancer le logiciel Tera Term : Effectuer la récupération des trames des capteurs Zigbee.

3) Programmation

Pour lire les trames, envoyées par les capteurs, on vous propose d'utiliser le code "lectureliaisonserie.php", ce programme met dans la variable « \$trameValide », le résultat de la lecture.

Ecrire un programme qui

- lit et affiche les trames envoyées par le simulateur.
- Calcul de Checksum et le compare avec celui de la trame reçue.

Extraire et exploiter les informations des capteurs Zigbee . Développer une stratégie pour déterminer si la personne est en détresse.

SEANCE 4, 5 ET 6 : DETECTION D'UN ETAT DE DETRESSE

Mise en pratique (voir le CAHIER des charges)

Barème et notation

Evaluation		Note
Communication écrite et orale	Un dossier du projet de 5 pages maximum (voir Guide pour la rédaction) Respect du temps imparti pour la présentation (10 min) Décrire son projet; Expliquer à l'aide de schémas synoptiques, photos ou vidéos son organisation et son fonctionnement;	/10
Algorithme programmation	Capacité à traduire un algorithme en langage PHP	/20
	Capacité à écrire un algorithme	
	Capacité à coder en PHP (Déclaration, Entrées, Traitements, Sorties et les commentaires)	
	Capacité à être autonome dans l'exécution des tâches de réalisation, de codage, de test, d'installation. Dégré d'autonomie dans la recherche des fonctions nécessaires au code.	
La réalisation	Etat et qualité de la réalisation (IHM)	/10
	Capacité à transférer les fichiers sur le serveur ISN	
Total		/40

Annexe : Table ASCII des 128 premiers caractères

Décimal	Hexadécimal	Caractère									
0	00		32	20		64	40	@	96	60	`
1	01	␣	33	21	!	65	41	A	97	61	a
2	02	␣	34	22	"	66	42	B	98	62	b
3	03	♥	35	23	#	67	43	C	99	63	c
4	04	♦	36	24	\$	68	44	D	100	64	d
5	05	♠	37	25	%	69	45	E	101	65	e
6	06	♣	38	26	&	70	46	F	102	66	f
7	07	·	39	27	'	71	47	G	103	67	g
8	08	█	40	28	(72	48	H	104	68	h
9	09	○	41	29)	73	49	I	105	69	i
10	0A	█	42	2A	*	74	4A	J	106	6A	j
11	0B	♂	43	2B	+	75	4B	K	107	6B	k
12	0C	♀	44	2C	,	76	4C	L	108	6C	l
13	0D	♪	45	2D	-	77	4D	M	109	6D	m
14	0E	♫	46	2E	.	78	4E	N	110	6E	n
15	0F	*	47	2F	/	79	4F	O	111	6F	o
16	10	▶	48	30	0	80	50	P	112	70	p
17	11	◀	49	31	1	81	51	Q	113	71	q
18	12	↕	50	32	2	82	52	R	114	72	r
19	13	!!	51	33	3	83	53	S	115	73	s
20	14	∏	52	34	4	84	54	T	116	74	t
21	15	§	53	35	5	85	55	U	117	75	u
22	16	-	54	36	6	86	56	V	118	76	v
23	17	±	55	37	7	87	57	W	119	77	w
24	18	↑	56	38	8	88	58	X	120	78	x
25	19	↓	57	39	9	89	59	Y	121	79	y
26	1A	→	58	3A	:	90	5A	Z	122	7A	z
27	1B	←	59	3B	;	91	5B	[123	7B	{
28	1C	⌊	60	3C	<	92	5C	\	124	7C	
29	1D	⊕	61	3D	=	93	5D]	125	7D	}
30	1E	▲	62	3E	>	94	5E	^	126	7E	~
31	1F	▼	63	3F	?	95	5F	_	127	7F	Δ