

Hemisson



HEMLCD



VERSION 1.1
AVRIL 2004

Auteur de la documentation

Alexandre Colot, K-Team S.A.
Ch. de Vuasset, CP 111
1028 Préverenges
Suisse

email : info@hemisson.com

Url : www.hemisson.com

MARQUE DÉPOSÉE :
Hemisson : K-TEAM SA.

INDICATIONS :

- Le contenu de ce manuel est susceptible de changer sans préavis.
- Tous les efforts ont été fait afin d'assurer l'exactitude de ce manuel.
Si toutefois vous trouvez une erreur, merci d'en informer K-Team S.A.
- Malgré les commentaires ci-dessus, K-Team S.A. ne pourra être tenu pour responsable des conséquences liées à une erreur dans ce manuel.



1	Introduction	4
1.1	Comment utiliser ce manuel	4
1.2	Précautions d'emploi	4
1.3	Recyclage	4
2	Connexion	5
2.1	Assemblage	5
2.2	Désassemblage	5
3	Le module HemLCD	6
3.1	Vue d'ensemble	6
3.2	Adresse I2C	7
3.3	Adresses des registres	7
3.3.1	Registre firmware	7
3.3.2	Registre interrupteurs	7
3.3.3	Registre ligne 1 justifiée à gauche	7
3.3.4	Registre effacer ligne 1	8
3.3.5	Registre ligne 1 centrée	8
3.3.6	Registre ligne 1 justifiée à droite	8
3.3.7	Registre ligne 2 justifiée à gauche	8
3.3.8	Registre effacer ligne 2	8
3.3.9	Registre ligne 2 centrée	8
3.3.10	Registre ligne 2 justifiée à droite	8
3.3.11	Registre effacer ecran	9
3.3.12	Registre backlight	9
3.3.13	Registre contraste	9
3.3.14	Registre changement d'adresse I2C	9
4	Utilisation	10
4.1	Programmation en C	10
4.1.1	void HemLCD_Init(void)	10
4.1.2	char HemLCD_Read_Version(void)	10
4.1.3	void HemLCD_Set_Backlight(unsigned char)	10
4.1.4	void HemLCD_Set_Contrast(unsigned char)	11
4.1.5	void HemLCD_Clear_Screen(void)	11
4.1.6	void HemLCD_Clear_Line1(void)	11
4.1.7	void HemLCD_Clear_Line2(void)	11
4.1.8	void HemLCD_Read_Interruptors(void)	11

4.1.9	void HemLCD_Line1_Left(char* Data)	11
4.1.10	void HemLCD_Line1_Centered(char* Data)	11
4.1.11	void HemLCD_Line1_Right(char* Data)	12
4.1.12	void HemLCD_Line2_Left(char* Data)	12
4.1.13	void HemLCD_Line2_Centered(char* Data)	12
4.1.14	void HemLCD_Line2_Right(char* Data)	12
4.2	Contrôle par port série	12
5	Example	13
A	Spécifications techniques	14
B	Connecteurs	15
C	Table ASCII	16



Le module HemLcd vous permet de gérer simplement et efficacement une interface utilisateur sans passer par l'ordinateur. En effet, vous pourrez afficher différents messages directement sur Hemisson mais aussi interroger l'utilisateur qui pourra répondre par le biais du clavier à 3 touches.

1.1 Comment utiliser ce manuel

Ce manuel introduit le module HemLcd destiné au robot Hemisson. Si ce manuel ne répond pas à l'un des problèmes auquel vous êtes confronté, consultez le site web Hemisson (www.hemisson.com) et plus particulièrement le Forum ou la section FAQs¹.

1.2 Précautions d'emploi

Voici quelques recommandations pour faire bon usage du module HemLcd :

- **Eloignez-le des endroits humides.**
Un contact avec de l'eau pourrait provoquer un court-circuit.
- **Entreposez-le de façon stable.**
Cela évitera les risques de chutes qui pourraient le casser ou causer des dommages à un tiers.
- **Ne branchez pas le module pendant que le robot est allumé.**
Pour éviter tout dommage, effectuez toutes les connexions lorsque le robot est éteint.
- **Ne laissez jamais Hemisson allumé lorsqu'il est inutilisé.**
Lorsque vous avez fini de travailler avec Hemisson, éteignez le.

1.3 Recyclage

Pensez à la fin de vie de ce matériel!

Certaines parties peuvent être recyclées, par exemple ne jetez pas votre pile à la poubelle mais portez-la plutôt dans des containers adaptés. En la recyclant, vous contribuez à offrir un environnement plus propre et plus sain pour les générations futures. Pour toutes ces raisons, faites attention à la fin de vie de ce matériel, en le retournant par exemple au fabricant ou à votre revendeur local.

¹Foire Aux Questions



L'assemblage et le désassemblage du module d'extension est une opération délicate. S'il vous plaît lisez les instructions ci-dessous afin d'éviter de faire subir des dommages à celui-ci. K-TEAM n'assumera aucune responsabilité pour les dommages causés par une mauvaise manipulation.

2.1 Assemblage

Commencez par vous assurer que le robot est éteint (position Off) puis insérez le plus verticalement possible et sans forcer le module HemLCD comme suit :

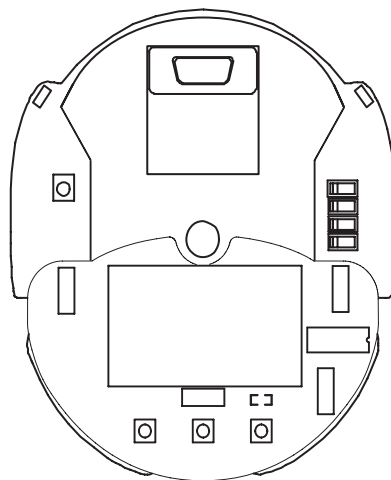


FIG. 2.1 – Positionnement du module HemLCD sur Hemissoon

Ne tentez surtout pas d'insérer HemLCD d'une autre façon ; cela pourrait causer de graves dommages à votre robot Hemissoon ainsi qu'au module d'extension.

2.2 Désassemblage

Pour le désassemblage, veuillez tout d'abord à ce que votre robot éteint (position Off), puis en tenant le robot d'une main, tirez verticalement le module HemLCD avec l'autre main.

3 LE MODULE HEMLCD



3.1 Vue d'ensemble

Le module HemLCD se compose comme suit :

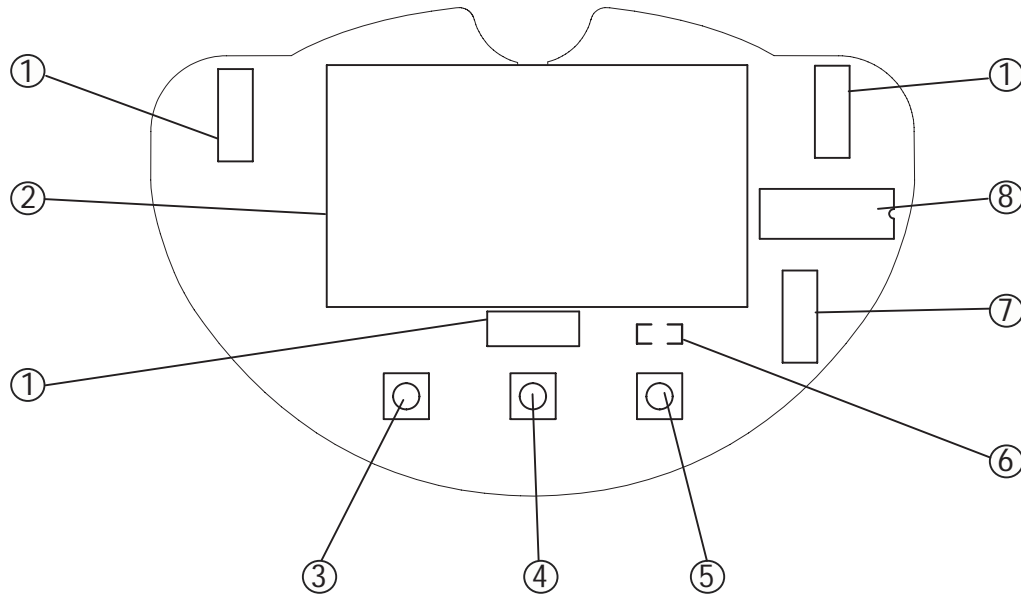


FIG. 3.1 – Détails du module HemLCD

1. Connecteurs d'extension Hemiison
2. Module LCD 2 lignes de 12 caractères
3. Interrupteur 1
4. Interrupteur 2
5. Interrupteur 3
6. LED
7. Connecteur de reprogrammation du module
8. Processeur PIC16F876 @ 4 MHz

3.2 Adresse I2C

Le robot Hemisson accède au module HemLCD par le biais d'une communication I2C. L'adresse de ce module est définie sur 7 bits + 1 bit pour la sélection du mode "lecture" ou "écriture".



FIG. 3.2 – Adresse I2C du module

Ainsi, l'adresse en mode "lecture" est **0xA1** ; et **0xA0** en mode "écriture".

3.3 Adresses des registres

Pour lire la valeur des interrupteurs ou encore envoyer du texte à afficher, il s'agit d'accès "lecture" et "écriture" à des registres dont la liste se trouve ci-dessous

Tout au long de la section suivante nous utiliserons la notation :

- *R* : pour un registre accessible uniquement en mode "Lecture".
- *W* : pour un registre accessible uniquement en mode "Ecriture".
- *R&W* : pour un registre accessible en mode "Lecture" et "Ecriture".

3.3.1 Registre firmware

Description : lire la version du firmware du module HemLCD.

Adresse : 0 (0x00)

Accès : R

3.3.2 Registre interrupteurs

Description : lire la valeur binaire (0 ou 1) de chaque interrupteur.

Adresse : 16 (0x10)

Accès : R

3.3.3 Registre ligne 1 justifiée à gauche

Description : envoyer les 12 caractères ASCII de la première ligne qui seront affichés de manière justifié à gauche.

Adresse : 16 (0x10)

Accès : W

3.3.4 Registre effacer ligne 1

Description : effacer le contenu de la première ligne.

Adresse : 17 (0x11)

Accès : W

3.3.5 Registre ligne 1 centrée

Description : envoyer les 12 caractères ASCII de la première ligne qui seront affichés de manière centrée.

Adresse : 18 (0x12)

Accès : W

3.3.6 Registre ligne 1 justifiée à droite

Description : envoyer les 12 caractères ASCII de la première ligne qui seront affichés de manière justifié à droite.

Adresse : 19 (0x13)

Accès : W

3.3.7 Registre ligne 2 justifiée à gauche

Description : envoyer les 12 caractères ASCII de la deuxième ligne qui seront affichés de manière justifié à gauche.

Adresse : 16 (0x10)

Accès : W

3.3.8 Registre effacer ligne 2

Description : effacer la deuxième ligne

Adresse : 33 (0x21)

Accès : W

3.3.9 Registre ligne 2 centrée

Description : envoyer les 12 caractères ASCII de la deuxième ligne qui seront affichés de manière centrée.

Adresse : 34 (0x22)

Accès : W

3.3.10 Registre ligne 2 justifiée à droite

Description : envoyer les 12 caractères ASCII de la deuxième ligne qui seront affichés de manière justifié à droite.

Adresse : 35 (0x23)

Accès : W

3.3.11 Registre effacer ecran

Description : effacer la contenu de la totalité de l'affichage.

Adresse : 48 (0x30)

Accès : W

3.3.12 Registre backlight

Description : régler le niveau du backlight (éclairage du module 0(0x00)-*i* Eteint, 255(0xFF)-*i* Allumé au maximum).

Adresse : 64 (0x40)

Accès : W

3.3.13 Registre contraste

Description : régler le niveau du contraste de l'écran (0(0x00)-*i* contraste minimum, 255(0xFF)-*i* contraste maximum). Cette valeur est automatiquement stockée en mémoire et sera utilisée par défaut lors du redémarrage du module.

Adresse : 65 (0x41)

Accès : W

3.3.14 Registre changement d'adresse I2C

Description : ce registre permet de changer l'adresse I2C du module. Cette nouvelle adresse est stockée en EEPROM ainsi il n'est pas nécessaire de lui redonner la nouvelle adresse à chaque redémarrage. Il faut lui spécifier une nouvelle adresse en mode écriture, donc une adresse paire (par exemple 0x50). Si vous entrez une adresse erronée (par exemple 0x51), le module retournera automatiquement à son adresse de base, c'est à dire 0xA0.

Adresse : 96 (0x60)

Accès : W



Il y a deux manière d'utiliser HemLCD avec HemiOs :

- Programmation en C
- Contrôle par port série

4.1 Programmation en C

Pour programmer HemiOs en C, vous devez avoir auparavant fait l'acquisition du compilateur CCS (disponible sur le site web HemiOs dans la section logiciels). Nous vous conseillons tout d'abord de lire la dernière version du manuel de l'HemiOs (<http://www.hemisson.com/French/assistance.html>), dans cette doc vous trouverez toutes les informations nécessaires sur l'HemiOs ainsi que l'installation de CCS.

- Téléchargez ensuite (<http://www.hemisson.com/French/assistance.html>) la dernière version disponible de l'HemiOs et de HemLCD.h.
- Ouvrez le projet avec CCS

La librairie hemLCD.h intègre les fonctions suivantes :

4.1.1 void HemLCD_Init(void)

But : Initialisation du module HemLCD.

Exemple :

```
HemLCD_Init();
```

4.1.2 char HemLCD_Read_Version(void)

But : Lire la version du firmware présent dans le module d'extension.

Exemple :

```
char Version;  
Version = HemLCD_Read_Version();
```

4.1.3 void HemLCD_Set_Backlight(unsigned char)

But : Définir le niveau du backlight.

Exemple :

```
HemLCD_Set_Backlight(127);
```

4.1.4 void HemLCD_Set_Contrast(unsigned char)

But : Définir le niveau du contraste.

Exemple :

```
HemLCD_Set_Contrast(0x86);
```

4.1.5 void HemLCD_Clear_Screen(void)

But : Effacer l'écran.

Exemple :

```
HemLCD_Clear_Screen();
```

4.1.6 void HemLCD_Clear_Line1(void)

But : Effacer la ligne 1.

Exemple :

```
HemLCD_Clear_Line1();
```

4.1.7 void HemLCD_Clear_Line2(void)

But : Effacer la ligne 2.

Exemple :

```
HemLCD_Clear_Line2();
```

4.1.8 void HemLCD_Read_Interruptors(void)

But : Lire les valeurs de interrupteurs. Les états sont ensuite stockés dans les variables globales SW1, SW2 et SW3.

Exemple :

```
HemLCD_Read_Interruptors();  
printf("Switches : %d %d %d\r",SW1,SW2,SW3);
```

4.1.9 void HemLCD_Line1_Left(char* Data)

But : Afficher ligne 1 justifiée à gauche. Pour cela il faut utiliser la table de caractères Line1.

Exemple :

```
strcpy(Line1,"Hello");  
HemLCD_Line1_Left(Line1);
```

4.1.10 void HemLCD_Line1_Centered(char* Data)

But : Afficher ligne 1 centrée. Pour cela il faut utiliser la table de caractères Line1.

Exemple :

```
strcpy(Line1,"Hello");  
HemLCD_Line1_Centered(Line1);
```

4.1.11 void HemLCD_Line1_Right(char* Data)

But : Afficher ligne 1 justifiée à droite. Pour cela il faut utiliser la table de caractères Line1.

Exemple :

```
strcpy(Line1,"Hello");  
HemLCD_Line1_Right(Line1);
```

4.1.12 void HemLCD_Line2_Left(char* Data)

But : Afficher ligne 2 justifiée à gauche. Pour cela il faut utiliser la table de caractères Line2.

Exemple :

```
strcpy(Line2,"Hello");  
HemLCD_Line2_Left(Line2);
```

4.1.13 void HemLCD_Line2_Centered(char* Data)

But : Afficher ligne 2 centrée. Pour cela il faut utiliser la table de caractères Line1.

Exemple :

```
strcpy(Line2,"Hello");  
HemLCD_Line2_Centered(Line2);
```

4.1.14 void HemLCD_Line2_Right(char* Data)

But : Afficher ligne 2 justifiée à droite. Pour cela il faut utiliser la table de caractères Line1.

Exemple :

```
strcpy(Line2,"Hello");  
HemLCD_Line2_Right(Line2);
```

Rappel : pour utiliser ces fonctions, vous devez avoir préalablement fait un include de la librairie au début de votre code : #include "HemLCD.h"

4.2 Contrôle par port série

A partir de la version 1.30 du Firmware d'Hemisson disponible sur le site web (<http://www.hemisson.com/French/assistance.html>) vous pouvez accéder au modules I2C par le biais de commande RS232. Pour cela référez-vous à la dernière version du manuel utilisateur Hemisson (! les fonctions décrites ci-dessous ne sont disponibles qu'à partir de l'HemiOs et du firmware V1.41!).

Exemples :

- Y,A0,11,01,41 > : affiche le caractère A justifié à gauche sur la première ligne.
- R,A0,00,01 > : renvoi la version du firmware.

5 EXAMPLE



Voici un exemple de code permettant d'utiliser le module HemLCD :

```
void main(void)
{
    hemisson_init(); // Initialisation d'Hemisson
    HemLCD_Init();   // Initialisation du module HemLCD
    strcpy(Line1,"Hemisson"); // Chargement de la chaîne "Hemisson"
                               // dans la variable Line1

    while(1)
    {
        HemLCD_read_Interruptors(); // Lecture des interrupteurs
        if (SW2 != 0)                // Test si le bouton du milieu est appuyé
        {
            hemisson_led_frontleft(1); // Allumage LED Hemisson
            hemisson_led_frontright(1); // Allumage LED Hemisson
            HemLCD_Line1_Centered(Line1); // Afficher ligne 1 centrée
            HemLCD_Set_Backlight(100); // Backlight à 100%
        }
        else
        {
            HemLCD_Set_Backlight(0); // Backlight éteint
            HemLCD_Clear_Line1(); // Effacer ligne 1
            hemisson_led_frontleft(0); // Eteindre LED Hemisson
            hemisson_led_frontright(0); // Eteindre LED Hemisson
        }
        delay_ms(100);
    }
}
```

A SPÉCIFICATIONS TECHNIQUES



- Dimensions [mm] : NC
- Poids [g] : NC
- Tension [V] : 5
- Courant [mA] : NC
- Freq. Max. I2C [KHz] : 400
- Nombre de caractères x lignes : 12 x 2
- Niveaux de contraste : 8 bits
- Niveaux de luminosité : 8 bits

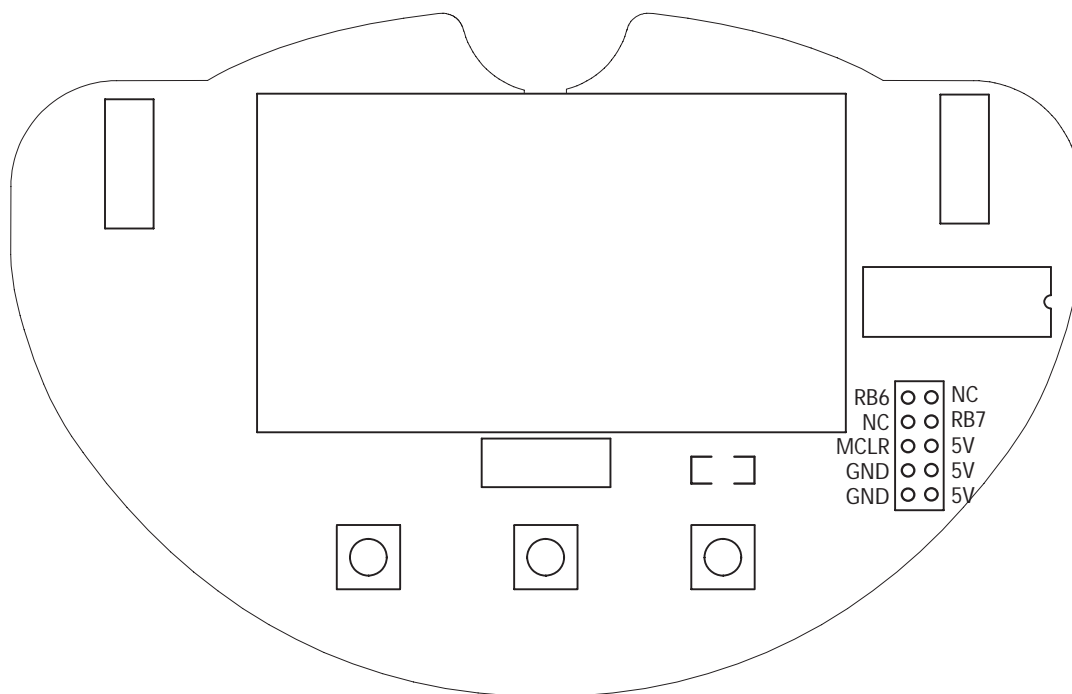


FIG. B.1 – Détails du connecteur de reprogrammation

C TABLE ASCII



Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[[123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

Source: www.asciitable.com

FIG. C.1 – Table ASCII



K-Team SA
1028 Préverenges
CH DE VUASSET, CP 111
SWITZERLAND
