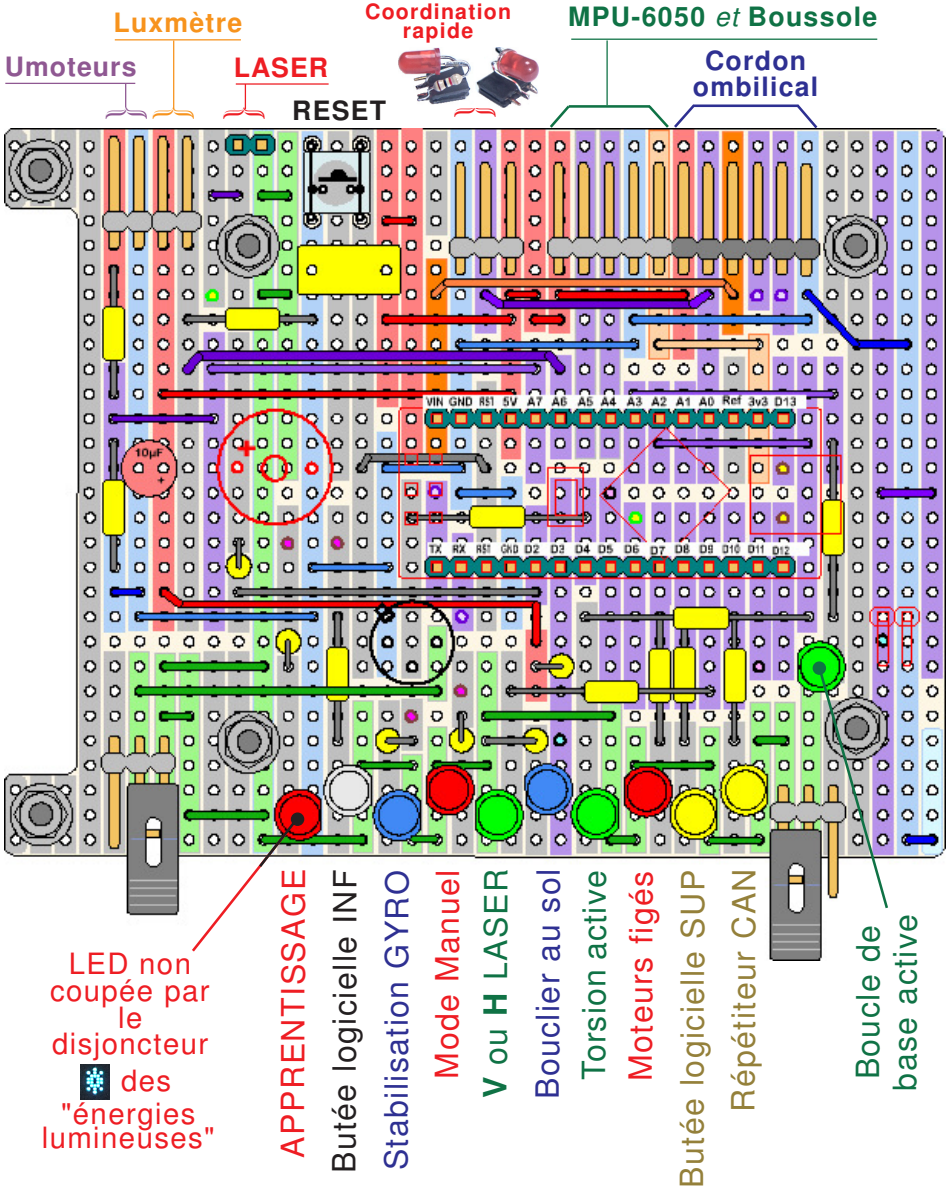
Affectation des LEDs témoin sur la sonde.

Le BUZZER de la sonde n'est plus utilisé par P50. Les alertes sonores se font par P40 directement sur le pupitre. Il peut éventuellement être déclenché lors de la mise au point de P50 pour servir de "sentinelle" pour vérifier l'appel à certaines instructions.

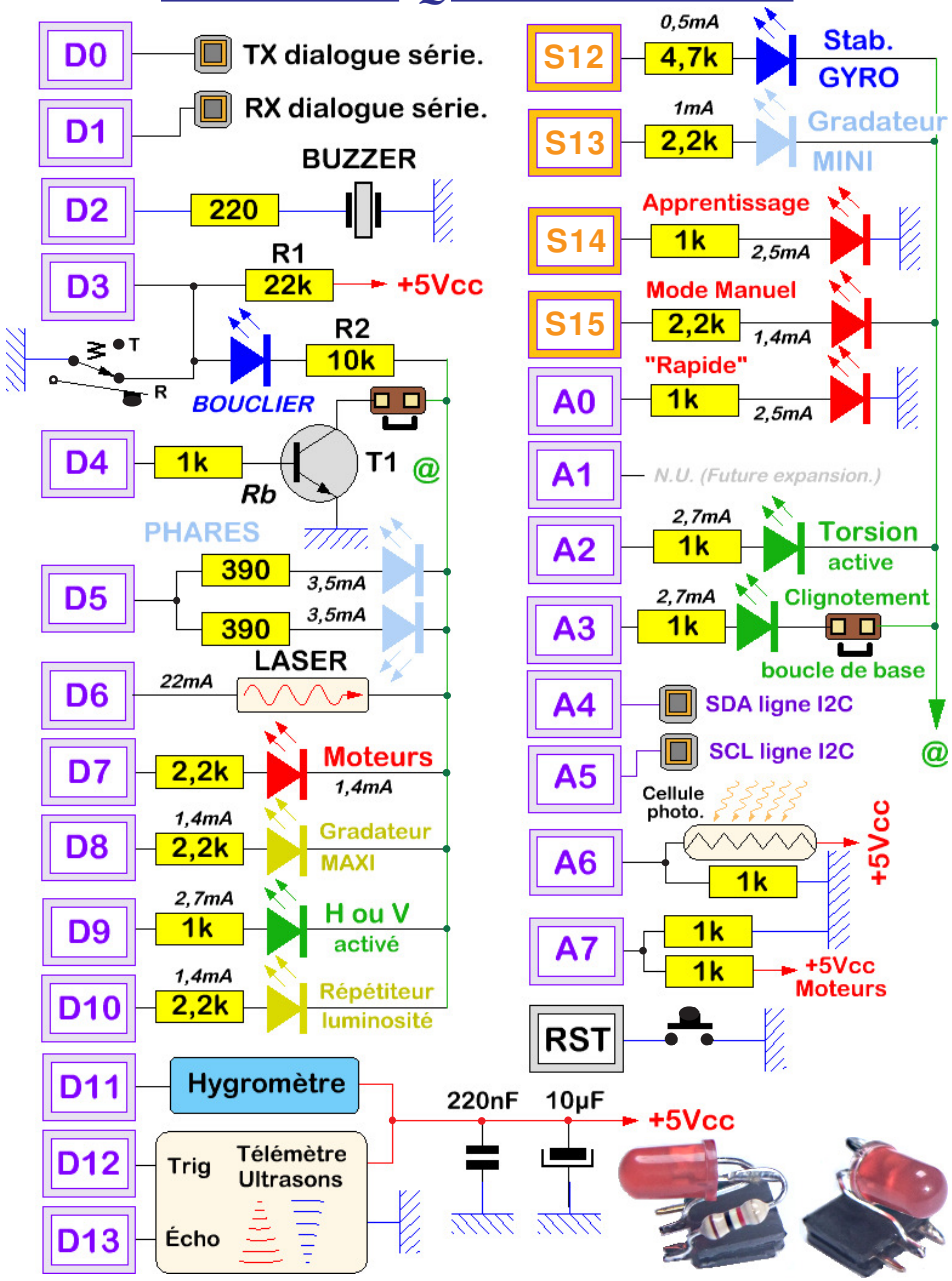


DOSSIER TECHNIQUE

ÉLECTRONIQUE de la SONDE. P02
Affectation des Entrées / Sorties. P03
ÉLECTRONIQUE du pupitre. P04
Affectation des Entrées / Sorties. P05
Circuit imprimé du clavier coté pistes cuivrées. P06
Circuit imprimé du clavier coté composants. P07
Affectations des touches du clavier. P08
Plaque cuivrée du circuit de recharge. P09
Menu EXPLOITER. P10
Boucle de base du pupitre. P11
Traitement des touches du clavier. P12
Touches d'ouverture des MENUS. P13
Informations diverses sur le mode apprentissage. P14
Traitement des consignes en mode apprentissage. P15
Enchaînement de plusieurs programmes. P16
Protocole pour recharger les accumulateurs. P18
Alimentation en puissance 6Vcc autonome. P19
Plaque cuivrée du circuit de recharge. P20
Gestion du pointage LASER. P21
Les POSTURES DE BASE en EEPROM. P22
Consignes pour les postures P23
Valeurs des consignes pour Stable Transversal. P24
Compensation de la rotation terrestre. P25
Enregistrer la dérive gyroscopique. P26/P27
Implantation EEPROM du Pupitre. P28/P29
Implantation EEPROM du programme SONDE. P30/P31
Dépose des accumulateurs de puissance. P32
Remontage des accumulateurs de puissance. P33
Branchements du circuit de puissance. P34
Dépose de la carte Arduino NANO. P35
Remontage de la plaque support intermédiaire. P35
Affectation des LEDs témoin sur la sonde. P36

Couleurs utilisées : Utilisation / Maintenance / Matériel / Logiciel

ÉLECTRONIQUE de la SONDE.



La LED de coordination "Rapide" branchée sur **A0** est placée sur le petit dispositif situé à tribord de la sonde et motré ci-dessus.

Dépose de la carte Arduino NANO.

1) Déposer les accumulateurs de puissance. *(Voir Page 32.)*
2) Débrancher tous les connecteurs HE14 :
   - Clavier *(6 broches.)* et LEDs clavier. *(8 broches.)*
   - Afficheur OLED *(4 broches.)*
   - TX/RX vers face avant. *(2 broches.)*
   - TX/RX/+/- vers INV face AR. *(4 broches.)*
   - Connecteur vers le codeur rotatif. *(5 broches.)*
3) Dévisser les quatre écrous et déposer la carte Arduino NANO. *(Pour dévisser l'écrou coté USB il faut passer la clef plate par l'ouverture rectangulaire. Attention au HE14 de RESET.)*

On peut à ce stade retourner le circuit pour vérifications.

4) Si on désire entièrement déposer la carte NANO il faut débrancher le circuit des LEDs de la face avant ce qui oblige à déposer la plaque support intermédiaire :
   - Enlever les quatre écrous et rondelles qui l'immobilisent,
   - Extraire la plaque le juste ce qu'il faut pour libérer le HE14 du circuit des LEDs de la face avant. *(La plaque peut rester à l'intérieur, on décale juste pour libérer les deux HE14.)*

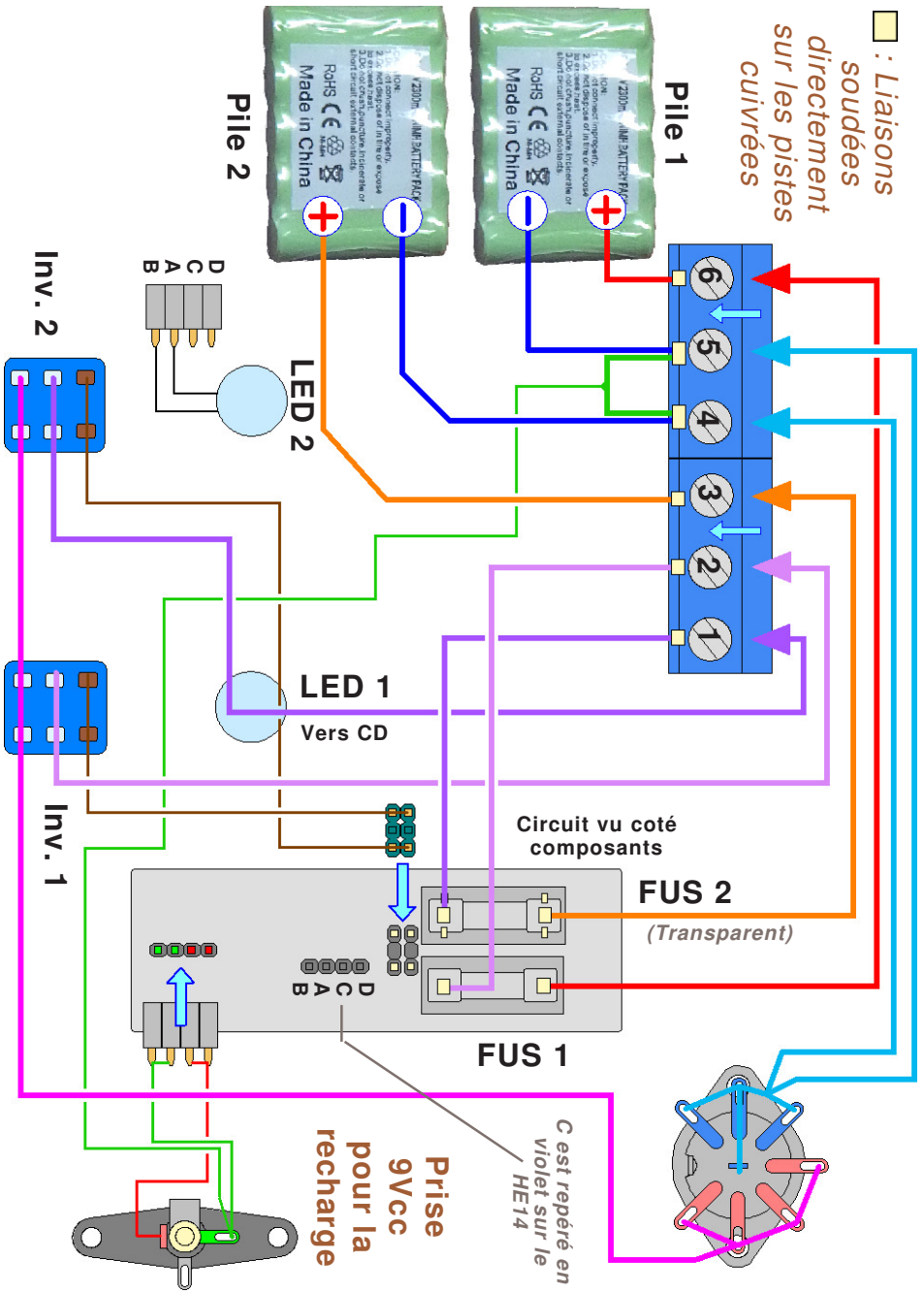
Remontage de la plaque support intermédiaire.

Le petit circuit imprimé des deux LEDs blanches témoins de rechargement est en place sur la plaque support des accumulateurs

1) Approcher Arduino, brancher le HE14 à 8 broches sur LEDs face AV.
2) Brancher le codeur rotatif. *(HE14 : 5 broches les 3 vertes à droite.)*
3) Dégager les torons et les immobiliser sur les cotés avec de petits fils électriques. Bien dégager les trous pour les deux LEDs blanches.
4) Placer la plaque sur ses quatre boulons support. Elle doit s'y insérer strictement sans forcer. *(ATTENTION aux LEDs blanches.)*
5) Mettre en place la carte Arduino NANO. Serrer les écrous **A** et **B** en premier avec la clef à tube. Puis **C** avec la clef plate par l'ouverture USB. Enfin **D** par l'intérieur avec la clef plate. ***Faire bien attention au fil X repéré en rouge*** sur la Fig.1 ci-dessous.
6) Brancher dans l'ordre le HE14 deux broches, puis le 4 broches de TX/RX/+/- suivi du 4 broches de l'afficheur OLED. *(Sa ligne passe sous la carte NANO.)*
   Brancher enfin le clavier et faire un test en alimentant avec la mini prise USB.

Fig.1

Branchements du circuit de puissance.



Affectation des Entrées / Sorties.

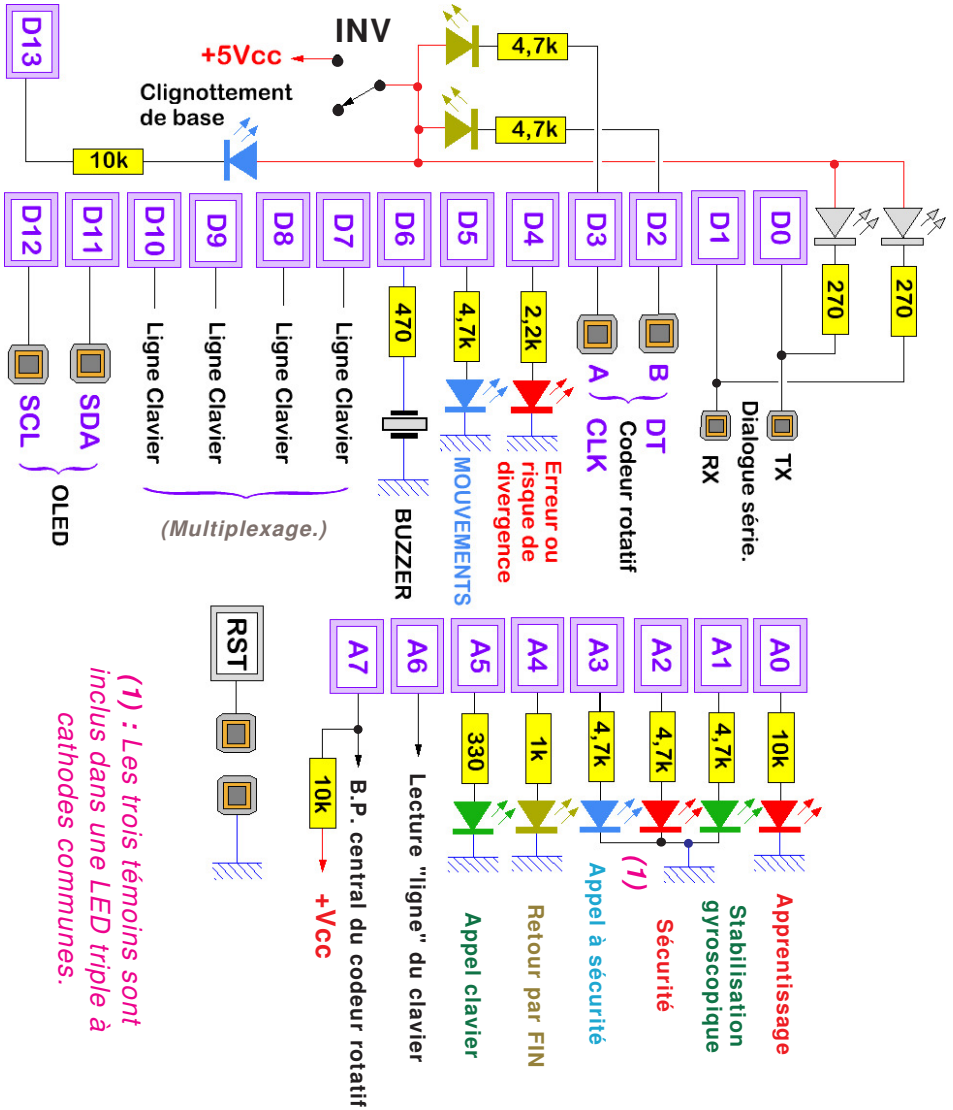
D0 : TX. *(Dialogue série entre deux Arduino.)*
D1 : RX. *(Dialogue série entre deux Arduino.)*
D2 : Pilotage du BUZZER.
≈ D3 : Détection du contact du bouclier avec le sol.
D4 : Coupure de masse de toutes les LED sauf celle sur S14 APPRENTISSAGE. *(Pilotage de T1.)*
≈ D5 : Éclairage analogique *(PWM)* des PHARES.
≈ D6 : Pilotage du LASER.
D7 : Pilotage LED témoin Moteurs figés.
D8 : Débordement positif du codeur incrémental.
≈ D9 : Pilotage LED témoin mode H ou V.
≈ D10 : Pilote la LED répétiteur du CAN "niveau énergie".
≈ D11 : Capteur Humidistance et de température.
D12 : Transmetteur ultrasons.
D13 : Récepteur ultrasons.

A0 : Sortie de pilotage de la LED "BUZZER".
A1 : Libre pour F.E
A2 : Pilotage LED de torsion active.
A3 : Clignotement de la LED de "boucle de PGM active".
A4 : SDA pour la ligne I2C.
A5 : SCL pour la ligne I2C.
A6 : Mesure de la lumière pour le luxmètre.
A7 : Mesure de la tension d'alimentation des moteurs.

Sortie S12 Multiplexeur : LED qui signale une stabilisation Gyroscopique est en cours. *(LED bleue.)*
Sortie S13 Multiplexeur : Butée logicielle négative. *(Niveau éclairages , pointage LASER ...)*
Sortie S14 Multiplexeur : LED du mode APPRENTISSAGE. *(Cette LED n'est pas coupée par le disjoncteur.)*
Sortie S15 Multiplexeur : LED rouge qui signale que le mode de pilotage manuel des moteurs est en cours

ÉLECTRONIQUE du pupitre.

Placé en face avant **INV** permet de couper les "témoins système" lumineux quand ils sont trop présents. Ces indicateurs sont techniques et ne présentent de la pertinence que durant certaines manipulations relativement épisodiques.

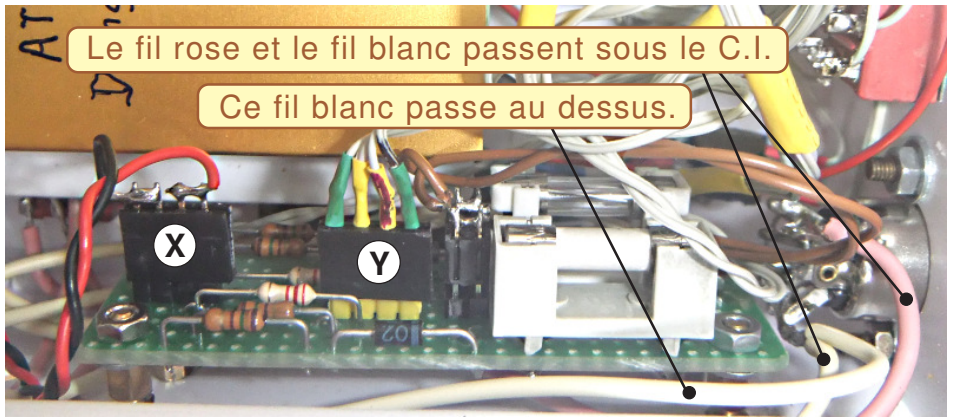


Le schémas du circuit de rechargement des accumulateurs de la motorisation est donné en page Page 20 et explicité en Page 19.

Remontage des accumulateurs de puissance.

1) La plaque support est en place ainsi que la carte Arduino. *(La procédure est donnée en Page 35.)*
2) Débrancher les connecteurs HE14 du clavier et dégager les torons vers l'arrière. Les y mobiliser par des petits fils rigides. *(Brider aussi la ligne +5Vcc débranchée qui va à l'accumulateur de la semelle.)*
3) vérifier les trois inverseurs coupés et les fusibles déposés.
4) Insérer les deux accumulateurs sans mettre la bride orange.
5) Immobiliser la prise de recharge 9Vcc sur la face avant.
6) Immobiliser le bornier de puissance sur la face avant.
7) Brider les accumulateurs avec la plaque orange.
8) Insérer le HE14 double sur le MSF. *(Fils marrons.)*
9) Immobiliser le MSF sur le coté gauche du coffret. *(Faire coïncider les trous taraudés des entretoises en utilisant provisoirement deux vis ϕ M3 longueur 40mm insérées en diagonale.)*

ATTENTION à ne pas pincer des fils. La mise en position manuelle doit se faire sans avoir à forcer sauf pour la raideur des gros fils.



10) Mettre en place le HE14 en **X** et le HE14 en **Y**. *(Faire attention au sens de branchement. Sur **Y** le souplisseau violet est à droite.)*
11) Établir les lignes de puissance du bornier en serrant avec le petit tournevis orange court. Replacer les fusibles et tester la sortie +6Vcc sur la prise DIN pour chaque inverseur.
12) Rebrancher le clavier et provisoirement le +5V issu de la semelle du coffret. Faire les essais sans la sonde puis en branchant la ligne de puissance et en testant entièrement la motorisation.

... / ...

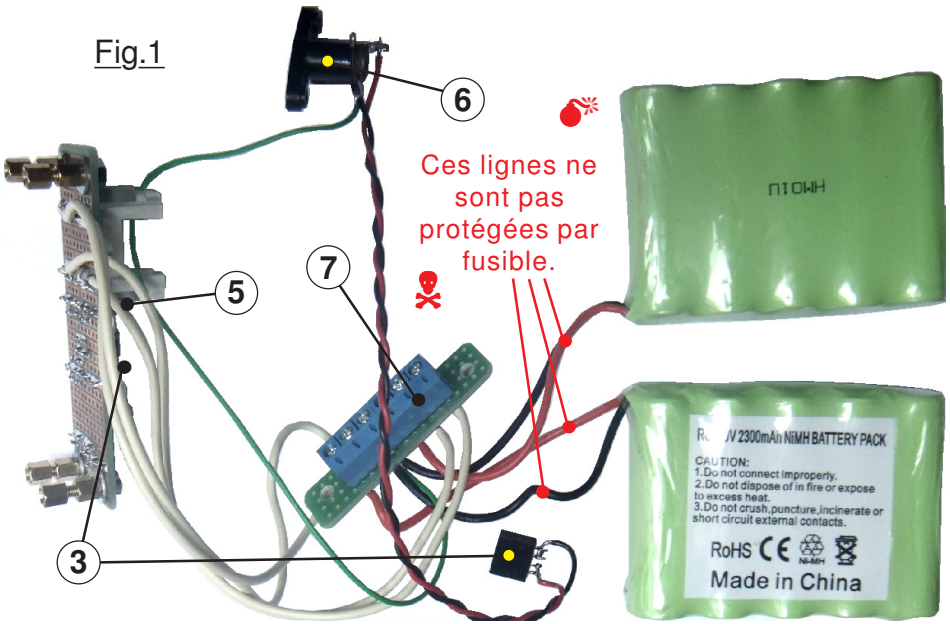
Dépose des accumulateurs de puissance.

1) Couper les trois inverseurs. *(Motorisation et console.)*
2) Enlever les deux fusibles. *(MSF : Module Support des Fusibles.)*
3) Débrancher le HE14 "Plus/moins" à quatre broches et le HE14 des deux LEDs blanches de rechargement du MSF.
4) Libérer les écrous et enlever la bride orange qui immobilise les deux accumulateurs 6V de puissance sur la plaque intermédiaire.
5) Débrancher le HE14 double du MSF. *(Fils marrons.)*
6) Libérer la prise 9Vcc de rechargement de la face avant.
7) Débrancher ①,②, ④ et ⑤ du bornier. *(Voir page 34.)*
(Utiliser le tournevis plat orange court.)
8) Enlever les vis de liaison du MSF et le dégager sur le coté.
9) Libérer le bornier de la face avant verticale en enlevant les boulons ϕ M3 et les entretoises d'écartement.

ATTENTION : Les accumulateurs sont soudés directement sur les pistes cuivrées sans fusible de protection. Éviter tout contact électrique intempestif avec les vis, les entretoises etc.

C'est l'ensemble complet *(Et "vulnérable" aux contacts électriques intempestifs.)* de la Fig.1 qui peut être entièrement déposé.

Fig.1

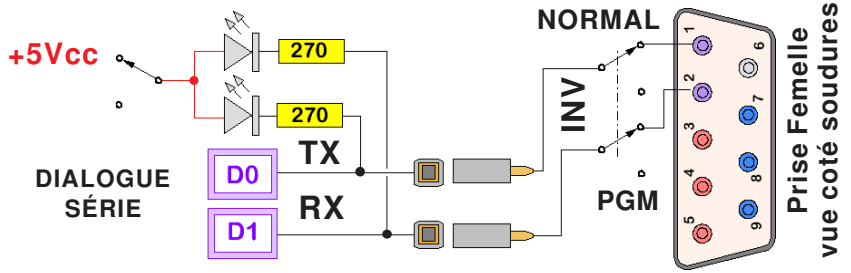


Affectation des Entrées / Sorties.

D0 : TX. *(Dialogue série avec la sonde.)*
D1 : RX. *(Dialogue série avec la sonde.)*
D2 : B *ou Ligne DT* du codeur rotatif.
≈ D3 : A *ou Ligne CLK* du codeur rotatif.
D4 : Pilotage LED "Erreur". Prévient également d'un risque potentiel pour la sonde si la LED clignote.
≈ D5 : LED qui signale le Menu MOUVOIR.
≈ D6 : Pilotage du BUZZER.
D7 : Balayage "ligne" du clavier.
D8 : Balayage "ligne" du clavier.
≈ D9 : Balayage "ligne" du clavier.
≈ D10 : Balayage "ligne" du clavier.
≈ D11 : SDA pour l'afficheur OLED.
D12 : SCL pour l'afficheur OLED.
D13 : LED qui signale l'activité de la boucle de base.

A0 : LED du témoin de MODE APPRENTISSAGE.
A1 : LED du témoin de la stabilisation gyroscopique.
A2 : LED qui signale un état SÉCURITÉ armée.
A3 : LED qui signale qu'il faut armer SÉCURITÉ.
A4 : LED qui précise un RETOUR par FIN exigé.
A5 : LED d'appel à une touche quelconque du clavier.
A6 : Lecture "ligne" du clavier.
A7 : BP central du codeur rotatif.

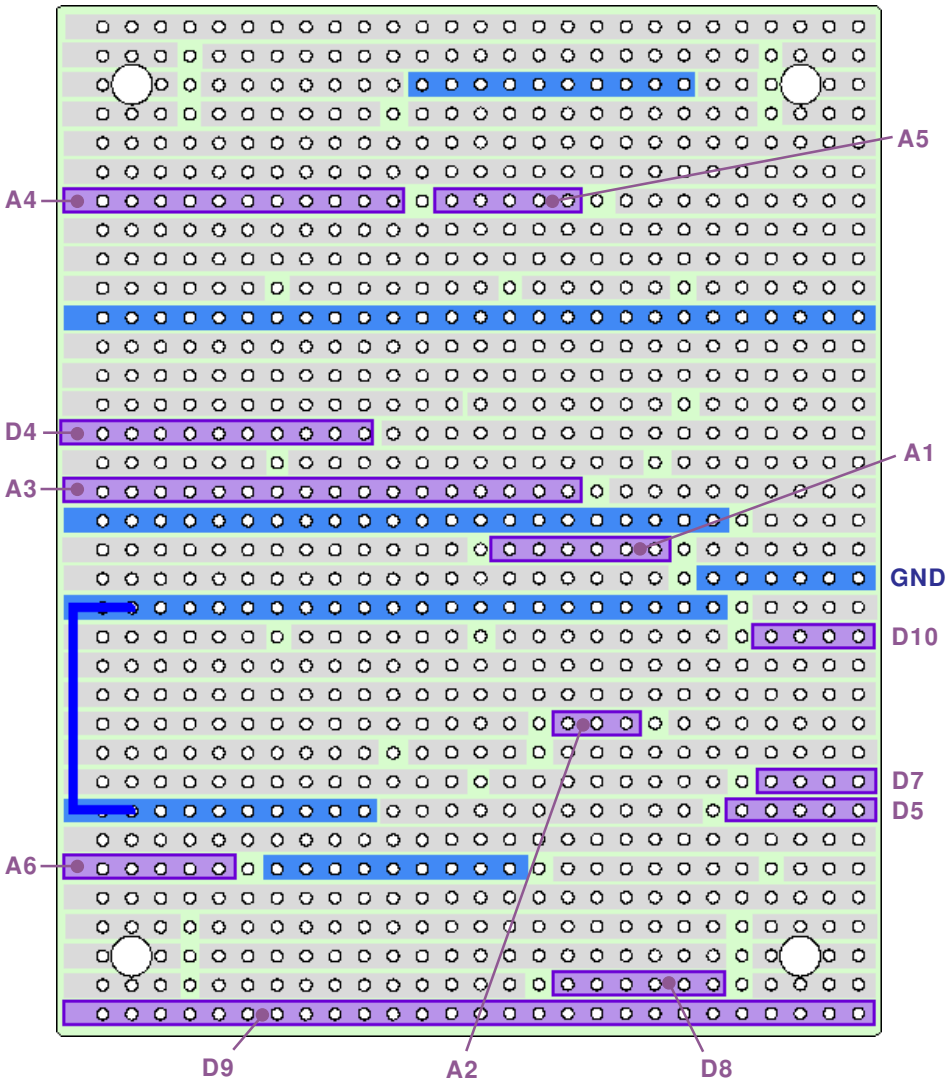
Pour pouvoir programmer librement l'un des deux ATmega328 alors que la sonde est reliée au pupitre, l'inverseur **INV** permet d'isoler la ligne de dialogue sur les deux voies du cordon ombilical.



Circuit imprimé du clavier coté pistes cuivrées.

Représenté vu coté cuivre ci-dessous, seules les pistes reliées par des liaisons filaires souples à l'ATmega328 du pupitre sont mises en évidence par coloriage sur le dessin. Les quatre diodes pour le multiplexage du clavier sur la broche **A6** du microcontrôleur sont des composants de commutation standard au silicium de type 1N4148. La chute de tension en mode direct est de l'ordre de 1V.

(Elles remplacent l'ancienne 1N914 : Courant de fuite 200 fois plus faible.)



Implantation EEPROM du programme SONDE.

	ADRS	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
(PGM n°2)	0512	00	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
	0528	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
(PGM n°3)	0544	00	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
	0560	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
(PGM n°4)	0576	15	37	36	1B	5A	1C	5A	1B	5A	1C	5A	1B	5A	1C	5A	1B
	0592	5A	1C	5A	1B	5A	1C	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
(PGM n°5)	0608	1A	1A	36	1D	5A	1E	5A	1D	5A	1E	5A	1D	5A	1E	5A	1D
	0624	5A	1E	5A	1D	5A	1E	5A	1D	5A	1E	19	FF	FF	FF	FF	FF
(PGM n°6)	0640	15	37	36	1B	5A	1C	5A	1B	5A	1C	5A	1B	5A	1C	5A	1B
	0656	5A	1C	5A	1B	5A	1C	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
(PGM n°7)	0672	1A	1A	36	1D	5A	1E	5A	1D	5A	1E	5A	1D	5A	1E	5A	1D
	0688	5A	1E	5A	1D	5A	1E	5A	1D	5A	1E	19	FF	FF	FF	FF	FF
(PGM n°8)	0704	1E	3D	37	15	0E	03	03	03	03	03	03	03	03	03	04	04
	0720	04	04	04	08	08	08	08	08	07	07	07	07	07	0E	3D	FF
(PGM n°9)	0736	0D	3D	37	16	15	0A	0B	10	11	0F	09	01	0E	3D	FF	FF
	0752	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

POSTURES enregistrées en EEPROM

	ADRS	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
①	0768	40	01	34	01	3C	01	21	01	2F	01	3A	01	06	01	24	01
	0784	36	01	2C	01	1D	01	33	01	FF	FF	FF	FF	FF	FF	FF	FF
②	0800	EF	00	3B	01	AD	01	58	01	28	01	C0	00	DE	00	2A	01
	0816	AA	01	5A	01	23	01	C7	00	FF	FF	FF	FF	FF	FF	FF	FF
③	0832	31	01	3B	01	A5	01	12	01	28	01	C0	00	1F	01	2A	01
	0848	AA	01	1D	01	23	01	C7	00	FF	FF	FF	FF	FF	FF	FF	FF
④	0864	EF	00	3B	01	AD	01	58	01	28	01	C0	00	DE	00	2A	01
	0880	AA	01	5A	01	23	01	C7	00	FF	FF	FF	FF	FF	FF	FF	FF
⑤	0896	E1	00	71	01	8B	00	60	01	F2	00	EC	01	D0	00	7B	01
	0912	9B	00	6A	01	D4	00	EC	01	FF	FF	FF	FF	FF	FF	FF	FF
⑥	0928	29	01	35	01	E5	00	00	01	28	01	86	01	1E	01	27	01
	0944	E5	00	0A	01	1C	01	8E	01	FF	FF	FF	FF	FF	FF	FF	FF
⑦	0960	31	01	35	01	E5	00	C2	00	22	01	86	01	8A	01	27	01
	0976	E5	00	04	01	1C	01	8E	01	FF	FF	FF	FF	FF	FF	FF	FF
⑧	0992	EF	00	3B	01	A5	01	58	01	28	01	C0	00	DE	00	2A	01
	1008	AA	01	5A	01	23	01	C7	00	FF	FF	FF	FF	FF	FF	FF	FF

ZONE	Nature des données.
Verte	Table des postures et des configurations.
Bleue	Enregistrement d'un balayage télémétrique.
Violet	Spectre colorimétrique.

Implantation EEPROM du programme SONDE.

P60_Clone_EEPROM_sonde.ino

ADRS	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
0000	31	01	D5	00	33	01	12	01	96	01	3A	01	1F	01	D7	00
0016	3F	01	1D	01	84	01	3A	01	FF	FF	FF	FF	FF	FF	FF	FF
0032	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0048	40	01	34	01	3C	01	21	01	2F	01	3A	01	06	01	24	01
0064	36	01	2C	01	1D	01	33	01	D3	00	98	00	3B	01	9D	01
0080	E6	01	3C	01	9F	00	8B	00	FF	00	9B	01	CF	01	45	01
0096	2F	01	EA	01	9A	00	2F	01	8D	00	E1	01	2F	01	D8	01
0112	91	00	2F	01	86	00	F5	01	EF	00	D5	00	33	01	58	01
0128	9E	01	3A	01	DE	00	D7	00	3F	01	5A	01	84	01	3A	01
0144	EF	00	98	00	E6	00	58	01	E3	01	87	01	DE	00	87	00
0160	E8	00	5A	01	D3	01	8E	01	12	01	D5	00	33	01	7B	01
0176	9E	01	3A	01	BB	00	D7	00	3F	01	37	01	84	01	3A	01
0192	EF	00	3B	01	A5	01	58	01	28	01	C0	00	DE	00	2A	01
0208	AA	01	5A	01	23	01	C7	00	38	01	3A	01	87	00	58	01
0224	9E	01	3A	01	DE	00	D7	00	3F	01	AB	00	84	01	3A	01
0240	EF	00	D5	00	33	01	58	01	9E	01	3A	01	DE	00	D7	00

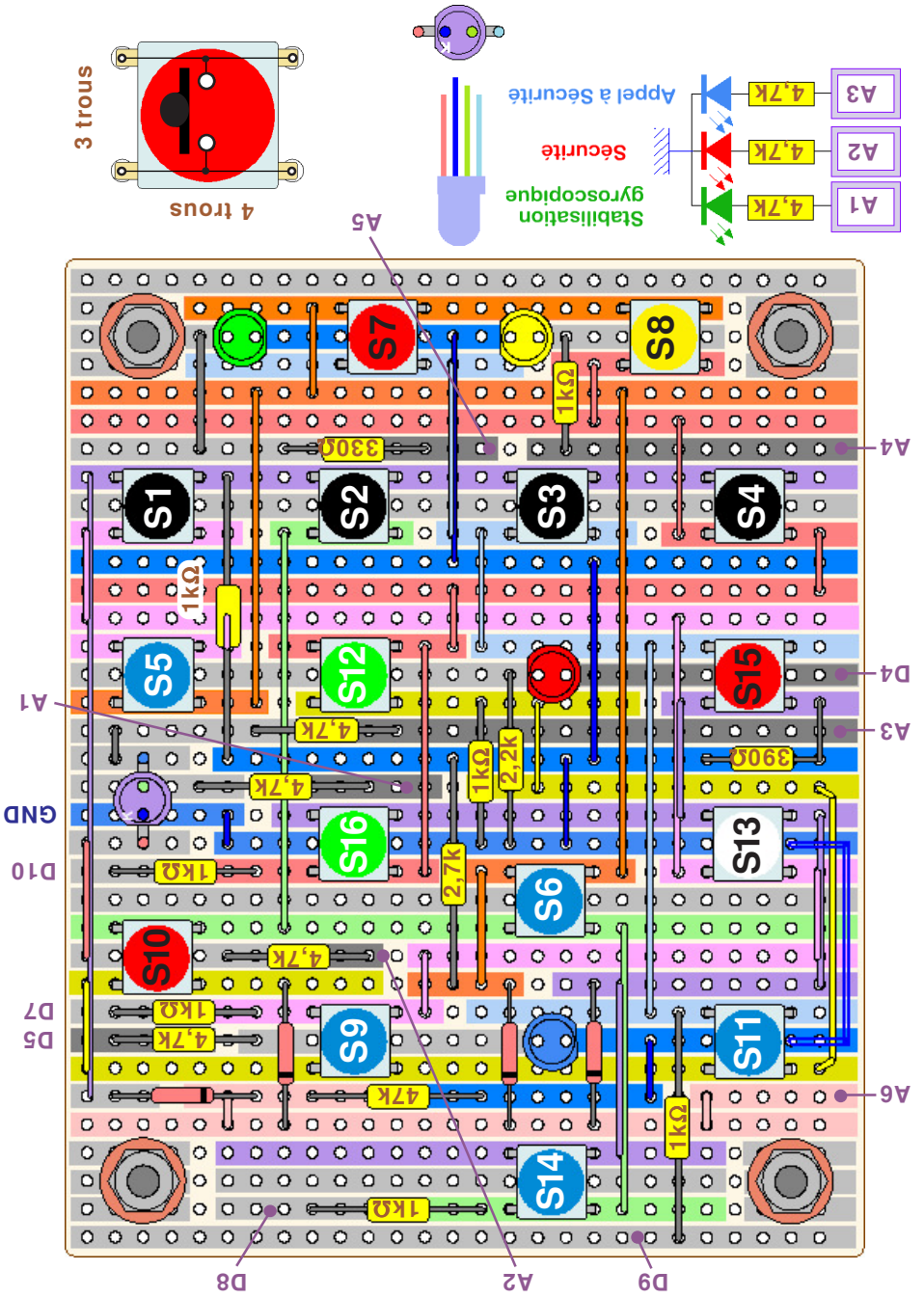
ADRS	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
0256	3F	01	AB	00	84	01	3A	01	CC	00	D5	00	33	01	35	01
0272	9E	01	3A	01	01	01	D7	00	3F	01	7D	01	84	01	3A	01
0288	EF	00	08	01	C1	01	58	01	5B	01	A7	00	DE	00	06	01
0304	D0	01	5A	01	38	01	9E	00	0D	01	D5	00	33	01	76	01
0320	9E	01	3A	01	FC	00	D7	00	3F	01	78	01	84	01	3A	01
0336	D1	00	D5	00	33	01	3A	01	9E	01	3A	01	C0	00	D7	00
0352	3F	01	3C	01	84	01	3A	01	EF	00	0E	01	19	01	58	01
0368	58	01	53	01	DE	00	D7	00	B7	01	5A	01	7D	01	BC	00
0384	EF	00	E6	00	AE	01	58	01	88	01	BE	00	DE	00	02	01
0400	1F	01	5A	01	4E	01	5A	01	EF	00	3B	01	A5	01	58	01
0416	28	01	C0	00	DE	00	2A	01	AA	01	5A	01	23	01	C7	00
0432	1C	1B	1A	19	18	16	14	11	0D	0C	0A	08	08	06	06	05
0448	05	06	06	07	07	08	09	0A	0C	0E	10	12	15	16	17	17
0464	EF	02	52	02	E2	01	13	02	78	02	B4	02	82	02	FF	FF
0480	00	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
0496	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

PTR = 24
PTR = 48
PTR = 72
PTR = 96
PTR = 120
PTR = 144
PTR = 168
PTR = 192
PTR = 216
PTR = 240
PTR = 264
PTR = 288
PTR = 312
PTR = 336
PTR = 360
PTR = 384
PTR = 408
PTR = 432
PTR = 464
PTR = 478

(PGM n°1)

NOTE : Actuellement la posture qui était située entre 24 et 48 n'est plus utilisée. La zone EEPROM est donc disponible en cas de besoin.

Circuit imprimé du clavier coté composants.



Affectations des touches du clavier.

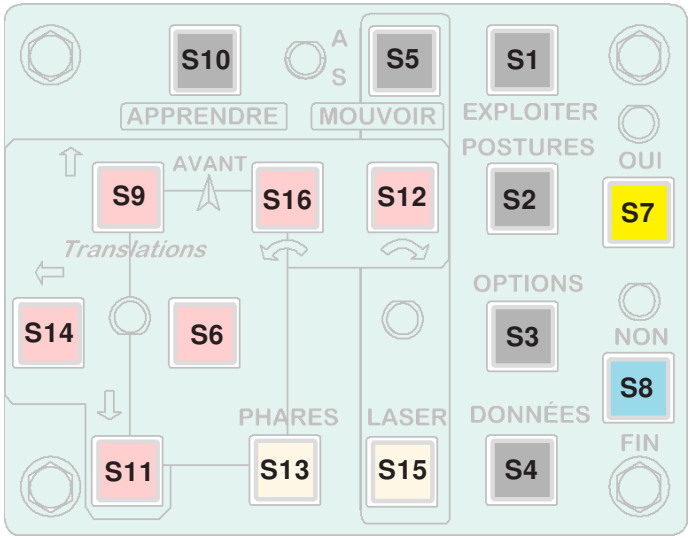


Fig.1

La Fig.1 précise les codes affectés à chaque touche du clavier. La répartition des divers codes résulte de l'utilisation d'un petit clavier expérimental dont la disposition des touches est "matricielle". Lors du développement du projet les fonctions des touches ont été distribuées comme montré sur la Fig.2 pour en faciliter l'usage.

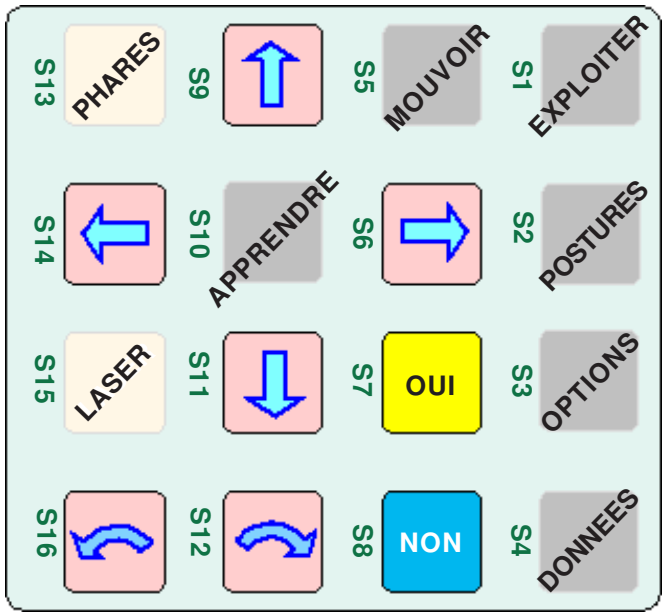


Fig.2

Implantation EEPROM du Pupitre. *(2/2)*

ADRS	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
0512	e		e	n		E	E	P	R	O	M	A	t	t	e	r
0528	r	i	s	s	a	g	e	D	e	c	o	l	l	a	g	e
0544	D	e	p	o	s	e	r		T	r	a	n	s	v	e	r
0560	a	b	l	e	N	T	I	O	N		:		R	e	t	o
0576	A	T	I	E	N	T	I	O	N		(	B	E	R	C	
0592	u	r		p	a	r		F	I	N		f	o	i	s	D
0608	E	A	U		!	)	R	o	t		f	o	r	t	e	u
0624	p	h	a	s	a	g	e		S	U	R	E	S	h	i	b
0640	·	T	E	L	E	M	E	S		u	n		b	a	l	a
0656	r	n	e		:		T	O	R	S	I	O	N		a	c
0672	e		?		?	P	i	l	o	t	a	g	e		:	=
0688	e		?		?	M	o	t	e	u	r		:		G	e
0704	E	L			?		M	O	T	E			H	a	n	c
0720	B	E		:	f	f	e	P	o	s	t	u	r	e		e
0736	G	r	i	f	f	e	P	o	s	t	u	r	e		e	n
0752	t	u	e	l	1	e	?	E	n	m	e	r	g	i		e
0768	1		a	2	5	4	·	A	r	m	e	r	f		a	c
0784		S	E	C	U	R	I	T	E		?	E	f	f	a	c
0800	e	r	r		C	1	e		P	G	M		L	i	s	t
0816	S	u	p	p	r	·		d	e	f	i	n	i	e	r	c
0832	o	d	e	?	R	E	C	H	A	R	G	E	d	i	t	C
0848	e	r	S		O	U	R	C	E	1	:		I	F	R	·
0864	A	P		a	c	t	u	e	l		=	T	a	n	g	2
0880					R	o	u	t	i	s		E			E	1
0896	m	E	2	1	x	x	x	x	x	E	c		a	r	t	e
0912	r	o	u	t	e		x	x	x			N	a	v	·	
0928		c	o	n	t	i	n	u	(	A	F	F	·	f	i	c
0944	r	)	E	1	9	E	2	2	1	m	s	g	e	A	c	t
0960	i	v	e	r	/	D	e	s	a	c	t	i	v	e	r	s
0976	?	N	b		d	e		c	h	a	i	n	a	g	n	e
0992	I		M		=		E		=	m	a	g	n	e	t	
1008	i	q	u	e		U	H	F	?	/	/	/	/	/	O	K

ADRS	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
0000	S	O	N	D	E		d	a	n	s		M	O	U	V	E
0016	M	E	N	T	S		P	O	S	T	U	R	E	S	C	o
0032	n	f	i	g	u	r	a	t	i	o	n	C	h	a	n	g
0048	e	r		P	G	M		E	E	P	R	O	M		?	T
0064	i	r		O	P	T	I	O	N	S		D	O	N	N	E
0080	E	S	E	X	P	L	O	I	T	E	R	e	p	p	e	t
0096	r		-		D	E	P	L	A	C	E	M	E	N	T	
0112	S		-	C	o	u	p	e	r		l	e	s			
0128	?	M	o	t	e	u	r	s		O	N		?		M	o
0144	u	v	e	m	e	n	t	s		R	A	P	I	D	E	S
0160		?	S	t	a	b	·		G	y	r	o	s	c	o	p
0176	e	?	C	L	V	M	o	d	e		A	P	P	R	R	E
0192	N	T	I	S	S	A	G	E	?	V	e	r	s	i	o	
0208	n		s	u	r		a	c	t	i	f	s	:		M	O
0224	U	R	S		a	c	t	i	f	s	:		S	a		M
0240	·	r		R	A	P	P	I	D	E		:	S	a	u	v
0256	r		l	a		P	O	S	T	U	R	E		?		Q
0272	I		T	T	E	R	B	o	u	c	l	i	e	r		a
0288		s	o		l		A	P	P	R	E	N	T	I		S
0304	G	E		:		P	o	i	n	t	e	r		E	1	1
0320	E	1	2	E	1	5	T	B	M	R	O	J	V	S	A	V
0336	s	p	e	c	t	r	e		c	o	u	l	e	u	r	
0352	?		A	F		O	U	I	N	O		N	P	H	A	R
0368		S			(	N	u	m			U	t	i	l	i	s
0384	e	r		l	e	T	e	s	t	R	e	v	e	i	l	l
0400	e	r	e		G	r	a	d	a	t	e	u	r		P	h
0416	r	e	s		·	L	A	S	E	R	·	P		1	2	8
0432		&		L	A	S	E	R		=		1	2	8	V	E
0448	I	L	L	E	S	t	a	b	l	e	R	a	i	s	o	
0464	n	a	b	l	e	H	a	u	t	e	u	r			M	a
0480	x	i	m	a	l	e	M	o	t	e	u	r	s		a	u
0496	N	e	u	t	r	e		O	P	A	p	p	r	i	s	

Implantation EEPROM du Pupitre. (1/2)

Pratiquement l'intégralité des textes qui sont affichés sur l'écran OLED sont logés en EEPROM de l'ATmega328. Comme le pupitre n'enregistre aucune donnée dans sa mémoire non volatile, les 1024 octets disponibles sont réservés pour du texte. C'est le programme P35_Ecrire_les_textes_en_EEPROM.ino qui en version ultime des logiciels sert à inscrire ces textes dans le circuit intégré. Les dialogues TX/RX ayant été modifiés à de nombreuses reprises se trouvent dans un ordre relativement quelconque.

```
SONDE dans MOUVEMENTS POSTURESConfiguration
Changer PGM EEPROM ?Tir
 OPTIONS DONNEESEXPLOITERepeter
  - DEPLACEMENTS -Couper les   ?Moteurs ON ?
Mouvements RAPIDES ?Stab. Gyroscope ?CLV
Mode APPRENTISSAGE ?
Version sur OFF : MOTEURS actifs : Mvt. RAPIDE :
Sauver la POSTURE ?QUITTERBouclier au sol
APPRENTISSAGE : Pointer E11E12E15TBMROJV
SAV spectre couleur ?AFFOUINON
PHARES    (Num
Utiliser leTestReveiller Gradateur Phares.
LASER.Phares & LASER = 128VEILLEStable Raisonnable
Hauteur MaximaleMoteurs au Neutre OPApprise en EEPROM
AtterrissageDecollageDeposer JEKERTStable Transversal
ATTENTION : Retour par FIN (BERCEAU !)Rot fois
Dephasage porteuse.TELEMESUREShiberne : un balayage ?
TORSION active ?Pilotage MANUEL ?Moteur : JAMBE :
HancheGenouGriffePosture actuelle ?Energie 1 a 254.
Armer la SECURITE ?Effacer le PGM Lister
Suppr. dernier code ?RECHARGEditerSOURCE :TFR.
CAP actuel =Tang.     Roulis E162mE21xxxxxx
Ecart route =AFF. Nav. continu(Afficher)E19E221msge
Activer/Desactiver ?Nb de chainagesT =M = E =magnetique
 UHF ?////OK
```


Diverses organisations logicielles.

➢ ***Répartition des booléens sur l'ACR du code 44.***

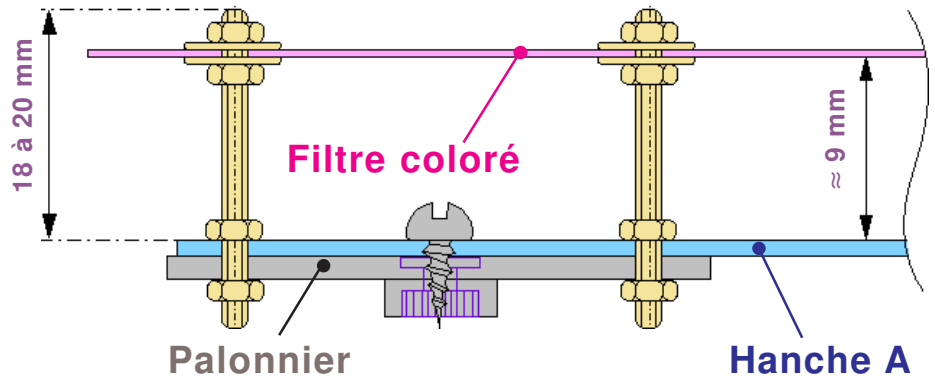
Ligne disjonctée	Moteurs ON	Coordination Rapide	Stab. Gyroscopique	Bouclier au sol	PHARES actifs	LASER actif	Apprentissage ouvert	Sentinelle : Fin d'ACR
0	1	0	1	0	0	1	1	*

➢ ***Mode_en_cours.***

Cette variable de type byte est utilisée dans tout le logiciel du pupitre pour que les séquences puissent savoir quel est le menu actuellement ouvert. La valeur de Mode_en_cours en fonction des menus vaut :

1 : EXPLOITER,
2 : POSTURES,
3 : OPTIONS,
4 : DONNEES,
5 : MOUVOIR,
6 : APPRENTISSAGE.

Montage mécanique du filtre colorimétrique.



Menu EXPLOITER.

L'ordre de traitement des items résulte de diverses évolutions du développement des programmes. Sur un RESET on pointe l'option n°4. *(Directive #define Num_initial_Exploitation 4.)* La procédure void **setup**() simule une rotation positive du codeur incrémental pour afficher le menu EXPLOITER. Pointer l'option n°4 permet donc d'ouvrir la page sur la fonction Reveiller JEKERT.

Le tableau donné ci-dessous résume l'ordre des fonctions dans :

```
if (Mode_en_cours == 1) { // Menu EXPLOITER.
    switch (Num_Exploitation) {
```

Num	Code	Fonction réalisée.
1	69	SAV un balayage ?
2	70	AFF un balayage ?
3	18	Sauver la POSTURE ?
4	61	QUITTER ?
5	55	Réveiller JEKERT.
6	59 / 57	Test Gyroscope.
7	54	Phares & LASER = 128.
8	43	Gradateur Phares.
9	42	Gradateur LASER.
10	62	Utiliser le LASER ?
11	67	SAV spectre couleur ?
12	---	AFF spectre couleur ?
13	38 / 39	TORSION active ?
14	73 / 74	Pilotage MANUEL ?
15	82	(Afficher) Posture actuelle ?
16	83	(Afficher) Posture en EEPROM ?
17	9	Tir LASER ?
18	---	CAP magnétique ?

Num dans ce tableau représente la variable Num_Exploitation. Les codes consigne oranges correspondent à ceux qui sont envoyés si on clique sur **FIN** pour terminer proprement la fonction.

➢ *Résultats et conclusions.*

Réalisé "sans finesse", P70_Derive_Gyroscopique.ino se contente d'enregistrer les valeurs de dérive retournées par la sonde sans se préoccuper du fait que cette dernière est assortie d'un signe et que son amplitude est limitée à 180°. Il aurait été relativement aisé de corriger les informations issues de la sonde pour afficher une valeur de dérive toujours positive est variant de zéro à +360°, cependant consommer du temps à ce perfectionnement a été estimé "non rentable".

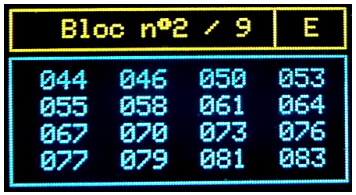


Fig.2

Interpréter les valeurs issues de l'IMU-6050 n'est pas immédiat car plusieurs phénomènes interviennent, et pour bien les cerner il serait indispensable de se plonger dans la théorie de fonctionnement de ce capteur complexe. La Fig.2 donne l'une des pages de données enregistrées. Entre chaque valeur il s'est écoulé une période de dix minutes. En théorie la variation devrait être constante. Comme les décimales ne sont pas affichées, on s'attendrait à une alternance ± 1° puisque tout affichage numérique est donné à ± une unité. Pour éluder ce problème d'affichage, on utilise le bloc de données n°5 montré sur la Fig.3 dont les valeurs négatives sont barrées en rouge. Avant ce bloc il y a 4 x 16 = 64 échantillons. L'échantillon encadré en rose est donc le n°68. Il s'est écoulé 68 fois dix minutes. Donc pour 170° de rotation enregistrés, une période de 680 minutes s'est écoulée. La vitesse de rotation enregistrée est donc de 170° / 680 = 0,25°/min soit exactement 15°/H. *(15°/H est aussi égal à 360°/Jour.)*

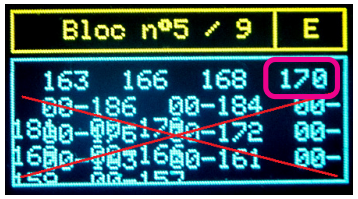


Fig.3

CONCLUSION : L'indication de l'Ecart de route calculé avec la référence interne Gyroscopique correspond à la rotation en LACET affectée de la rotation terrestre de 15°/H. Comme cette dérive est constante, il est possible de la compenser par logiciel en ajoutant 1° toutes les 4 minutes, option incluse à la version ultime du programme. Sur le plan de la navigation c'est idéal. Reste que pour l'expérimentation ludique, conserver l'influence de ce phénomène est formateur, raison pour laquelle le mode G est conservé.

Enregistrer la dérive gyroscopique.

Purement informative cette expérience impose de téléverser provisoirement P70_Derive_Gyroscopique.ino sur Arduino du pupitre de commande. Ce démonstrateur s'utilise avec sur la sonde le programme normal P50_PGM_ESCLAVE_SONDE.ino et ne modifie strictement rien dans l'EEPROM du microcontrôleur maître. Dans un premier temps on téléverse le démonstrateur P70 puis on établit la liaison **TX** / **RX** entre les deux cartes Arduino.

➢ Protocole de mesure de la dérive gyroscopique.

>>>>>> Pas d'alimentation de puissance pour la motorisation.

01) Établir la liaison de dialogue entre les deux ATmega328 >
02) RESET sur les deux cartes Arduino >
03) Cliquer sur un BP quelconque pour passer la page d'accueil >
04) **OUI** pour vérifier que la sonde est active >
05) OPTIONS > **BP***ccr* pour allumer la LED SÉCURITÉ >
06) Un pas ↺ pour avoir l'item AFF. Nav. continu ? >
07) **OUI** pour valider le mode >
08) Positionner la sonde bien à l'horizontale : R = 0° et T = 0° >
09) Orienter la sonde plein Nord >
10) **FIN** pour sortir proprement de l'affichage >
11) Menu MOUVOIR : **Le processus démarre** >
12) Activer une touche du clavier pour éteindre l'écran OLED >

À tout moment on peut allumer ou éteindre l'écran par une touche quelconque. Quand les 24H sont écoulées, la LED bleue s'allume, l'écran se "réveille" et affiche le premier bloc de données en mode "Ecart de route". Chaque touche autre que **OUI** change de page écran et affiche seize échantillons de plus. Quand les neuf pages sont affichées il y a recyclage à la page n°1. La touche **OUI** fait alterner entre le mode "Ecart de route" symbolisé par E ou la dérive mesurée en dix minutes symbolisé par la lettre D'. *(D pour différence.)* Sur la Fig.1 dix neuf échantillons ont été enregistrés comme indiqué en **2**. La dérive actuelle précisée en **1** est de 46°. Un échantillon est enregistré toutes les dix minutes. Le chronomètre en **3** décompte à partir de 600 toutes les secondes.

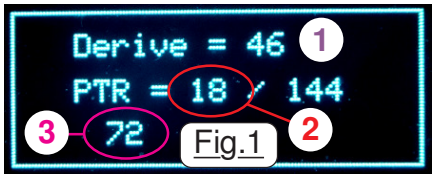


Fig.1

13) Téléverser 50_PGM_ESCLAVE_SONDE.ino pour terminer.

Boucle de base du pupitre. *(P40)*

Mise à part la séquence (1) qui du reste ne comporte que quelques instructions, void **loop**() ne fait appel qu'à quatre procédures. Ainsi dépouillée, les structures du programme de base et de la tâche de fond sont très faciles à comprendre. Représentée sous la forme d'un organigramme sur la Fig.1 on observe que cette séquence (1) chronomètre pour déclencher à chaque dixième de seconde les éventuels clignotements de trois diodes électroluminescentes. Le chronométrage est effectué en permanence même si le processeur est occupé dans d'autres procédures, car il fait appel aux ressources internes de l'ATmega328 par le biais de l'instruction millis() du langage C d'Arduino.

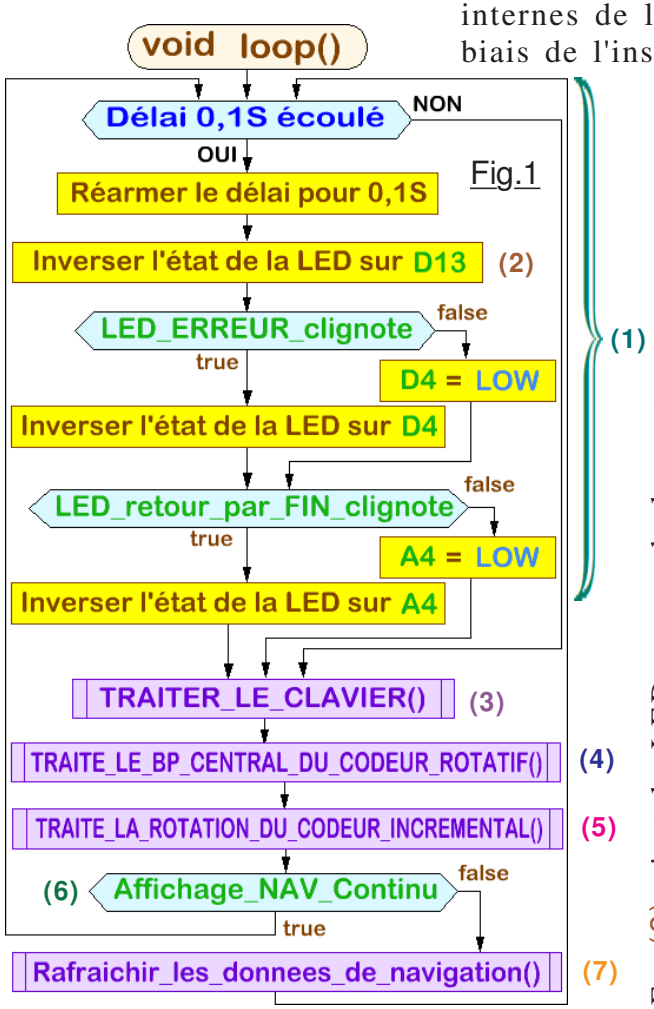


Fig.1

En (2) c'est la LED rouge de la carte NANO Arduino qui en clignotant visualise le fonctionnement de void **loop**(). En (3), si une touche est cliquée sur le clavier elle sera traitée. (4), pour son compte s'occupe de savoir si le **BP***ccr* a été activé, et effectue le traitement qui le concerne. Si le capteur incrémental a tourné, il est pris en compte en (5). Enfin si le test (6) est positif, en (7) le programme saisit les données de navigation et les affiche sur l'écran OLED.

... / ...

Traitement des touches du clavier.

L'organigramme de la Fig.1 résume le déroulement du programme quand on clique sur l'une des touches du clavier. La procédure (1) retourne la variable BP égale à zéro si aucune touche n'est activée lors du test, provoquant la sortie immédiate de la subroutine. Si une touche est détectée, la première action en (2) consiste à empiler son code dans la table de l'historique. Si dans le menu EXPLOITER on clique sur la touche ⇧, en (4) on affiche l'historique des touches sur l'écran OLED. Mais dans le menu EXPLOITER on peut aussi cliquer sur ⇧ pour diriger le LASER ou mouvoir un moteur en mode manuel. Le test (3) élimine ces deux cas.

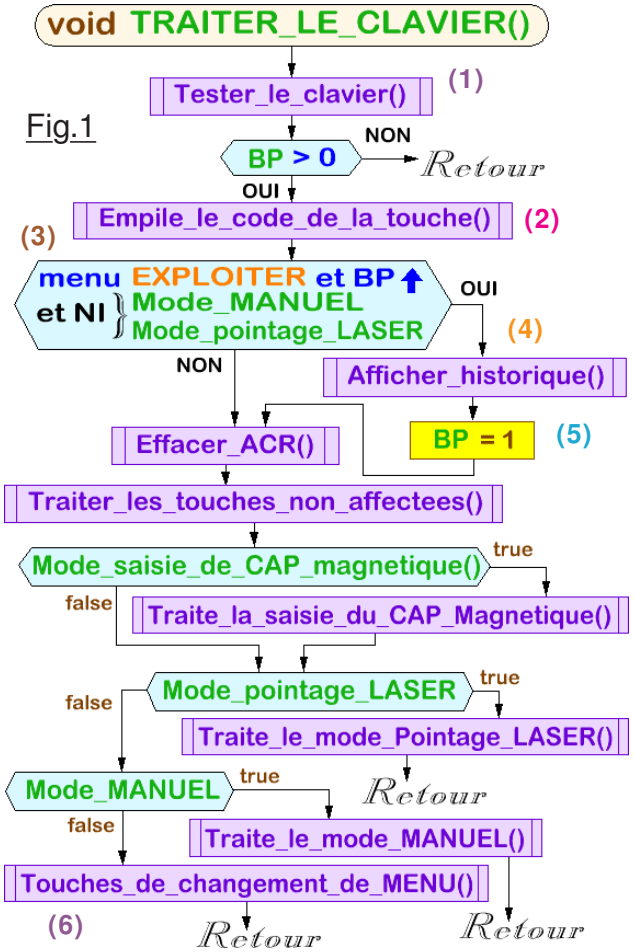


Fig.1

Un problème se pose quand l'affichage de l'historique est invoqué : Pour laisser à l'opérateur le temps d'observer les données, il y a attente de la touche FIN. La procédure ATTENDRE_un_BP() laissera le code touche 8. Hors en (6) la procédure déduirait qu'il faut traiter cette touche FIN. En faisant croire en (5) que S1 a été activée, la subroutine ne fera que confirmer le menu EXPLOITER. NOTE : Quand on clique sur le BP *ccr* un code touche spécifique 99 est généré. Différent de zéro, il est donc empilé dans l'historique.

... / ...

Compensation de la rotation terrestre.

Lorsque l'on utilise la référence interne gyroscopique en LACET de l'IMU-6050 pour déterminer les changements d'orientation en Cap de la sonde, le dispositif est influencé par la rotation terrestre. Pour neutraliser ce phénomène il suffit de compenser régulièrement la valeur calculée de l'Ecart de route que retourne la sonde au pupitre quand elle a reçu la consigne 91 pour afficher en continu les données de navigation.

Dérive	-0,25°/min	-2,5°/10 min	-15°/H	-360°/Jour

➢ Gestion de la correction sur la sonde.

Quand dans le menu OPTIONS on valide l'option C = ////, JEKERT a reçu chronologiquement les consignes :

- Code 97 qui impose la référence Magnétique, *(Ou également le mode Directionnel D)* cette consigne positionne à false les variables Reference_gyroscopique et Corriger_gyroscope.
- Code 98 qui impose la référence Gyroscopique, qui ne fait que positionner à true la variable Reference_gyroscopique. Le booléen Corriger_gyroscope reste donc à false en mode G.
- Code 95 qui valide C = //// : Recale le gyroscope sur la valeur actuelle du LACET, *(Pour avoir Ecart = 0 au début des affichages en continu de la navigation)* force à true Corriger_gyroscope et force à zéro la variable Correction_derive. Le chronomètre pour 4 minutes dans la boucle de base est déclenché.

Ecart = Lacet - Calage_gyro_initial + Correction_derive.

➢ Ajustement de la correction de dérive.

Pour corriger l'Ecart de route qui est de 0,25°/min il suffit d'ajouter 1° à la variable Correction_derive toutes les quatre minutes. Ainsi on manipule des entiers ce qui minimise le code. Pour cette durée théorique le chronomètre devrait être chargé à 60 x 4 = 240 secondes, soit 240 000 mS. Hors dans le programme source on observe une instruction chronometre_a_4_minutes = millis() + 230 000;
Cette valeur plus faible de 4,5% compense des petits retards qui s'accumulent au cours du temps. *(Dialogues série ...)* Avec cette correction on aboutit à +3° en 10H 30min soit une imprécision de l'ordre de +0,3° par heure ce qui pour cette application est dérisoire.

Valeurs des consignes pour Stable Transversal.

Posture "stratégiquement" la plus utilisée, il importe d'en optimiser les consignes, c'est à dire imposer aux divers servomoteurs des orientations qui conduisent à une répartition équivalente du poids de la petite machine sur les quatre Griffes. Le pilotage manuel n'autorise qu'une approximation. On configure visuellement les quatre Jambes. Mais rien ne prouve que chaque membre subisse une charge équilibrée. De plus, avec les nombreux incidents survenus durant la mise au point des programmes on doit s'attendre à de légères déformations des Fémur voir des Tibias.
La posture étant déjà disponible sous forme approximative :

➢ Protocole pour optimiser Stable Transversal

01) POSTURES > ↺ pour Stable Transversal. > **OUI** >
02) **BP***ccr* pour libérer les efforts >
03) ***Vérifier que les griffes sont bien en contact avec le sol*** >
04) OPTIONS > Deux ↺ pour Stab. Gyroscope ? >
05) **BP***ccr* pour armer la SÉCURITÉ >
06) **OUI** pour activer le mode > Surveiller visuellement, et ***si la motorisation diverge couper IMMÉDIATEMENT l'énergie de puissance***. La LED rouge s'éteint, la LED verte s'allume >
07) POSTURES > **BP***ccr* pour libérer les efforts >
08) Couper l'alimentation en puissance des moteurs >
09) OPTIONS > Deux ↺ > **NON** pour couper la stabilisation >
10) Un ↻ pour avoir AFF. NAV. continu ? >
11) **BP***ccr* pour armer la SÉCURITÉ > **OUI** >
12) Vérifier que Tangage et Roulis = 0°.
13) **FIN** > EXPLOITER >
14) Huit ↺ > **OUI** pour faire afficher la table des consignes >
15) Noter les douze valeurs de consignes (@) >
16) Chargez le source P60_Clone_EEPROM_sonde.ino dans l'éditeur de l'**IDE** et modifier les valeurs du bloc // 06 ---- >
17) Téléverser P60_Clone_EEPROM_sonde.ino dans JEKERT >
18) Téléverser P50_PGM_ESCLAVE_SONDE.ino dans JEKERT >
19) Imposer Stable Transversal, vérifier la portée des Griffes.

(@) : Il est fortement recommandé d'effectuer entièrement et deux fois les manipulations et de faire la moyenne des valeurs mesurées.

Touches d'ouverture des MENUS.

Montré sur la Fig.2 les touches dédiées à l'activation des divers menus ont une action permanente et ne sont pas fonction du menu ouvert. En (1) on teste toutes les touches de MENU sauf S10. L'action en (2) a pour effet d'ouvrir la page désirée dans la liste des

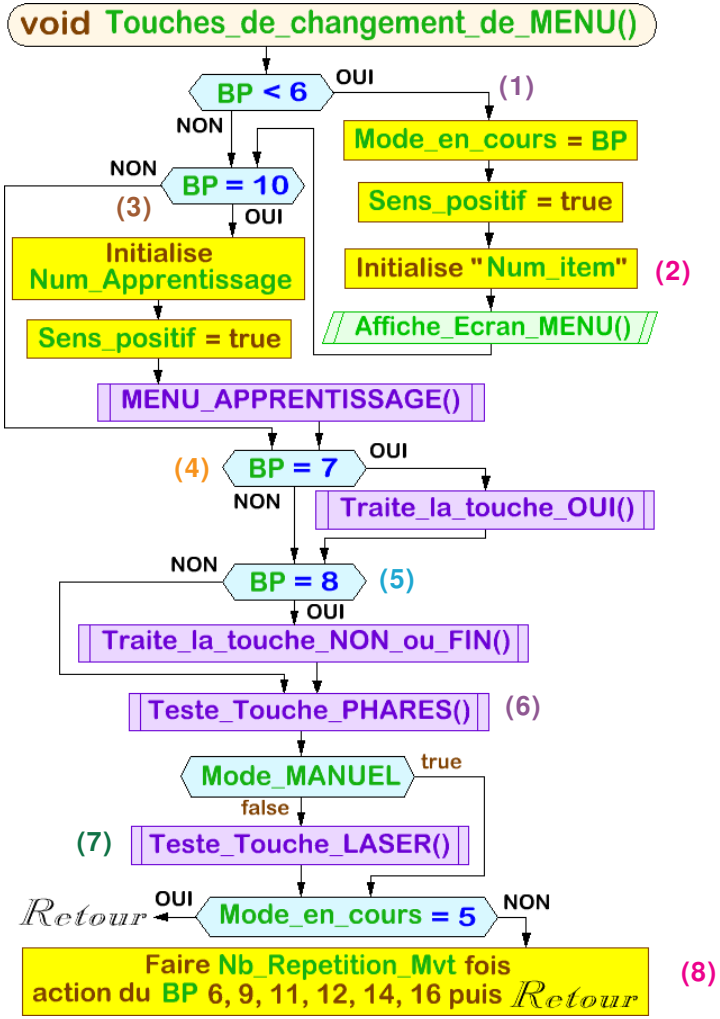


Fig.2

options possibles. En (3) on gère la touche S10. Puis en (4) et en en (5) que le programme prend en compte les touches **OUI** et **NON**. La procédure (6) traite la touche S13 alors que la subroutine (7) se charge de prendre en compte la touche S15. Enfin en (8) le menu MOUVOIR est traité à son tour si sa touche dédiée est cliquée.

Informations diverses sur le mode apprentissage.

Editer un programme consiste à avoir le mode APPRENTsge ouvert et à envoyer des instructions à la sonde. Les codes valides sont alors ajoutés à la suite des instructions actuelles dans le bloc indexé en EEPROM et ce jusqu'à la saturation à 30.

Format d'un bloc programme : *30 codes au maximum*

Chaque bloc programme mobilise en EEPROM 32 octet, le dernier n'étant pas utilisé. Le premier octet contient en permanence le nombre d'instruction actuel dans le programme. Si le programme est effacé, ce nombre est forcé à zéro. Il aurait été possible de sauvegarder 31 instructions par bloc, mais cette valeur n'est pas "esthétique" et surtout la page d'affichage ne comportait que trente cases pour indiquer les codes du programme lors des listages. La Fig.1 présente l'organigramme de traitement de l'ouverture du mode apprentissage quand la sonde reçoit sur RX le code 71.

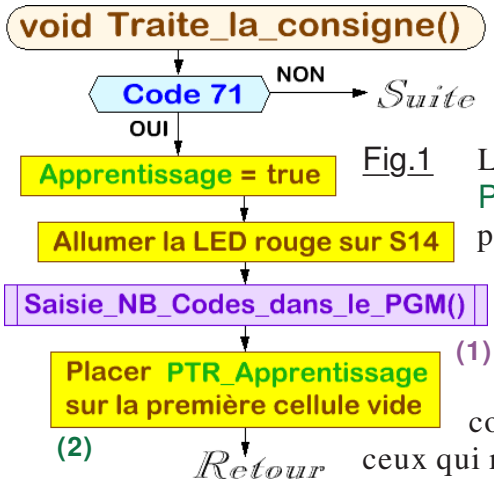


Fig.1

La procédure (1) laisse l'index PTR_Apprentissage sur la première cellule qui contient Nb. Le traitement (2) se contente d'incrémenter le pointeur et d'y ajouter la valeur contenue dans la cellule Nb. Sont filtrés les codes invalides, c'est à dire tous ceux qui retournent autre chose que "OK" ainsi que des instructions qui ne correspondent pas à des actions de la sonde sur le terrain. *(Exemple : Imposer un bloc comme source.)*

Le tableau donné ci-dessous liste les codes NON valides :

12	13	18	20	23	24	37	44	45	46	47
48	49	50	51	52	56	57	58	59	60	68
70	71	72	82	83	84	85	86	87	88	89
91	92	93	94	95	96	97	98			

Ils sont repérés par '*/' dans le manuel d'utilisation. En toute logique ils ne doivent jamais apparaitre dans les listages de blocs enregistrés.

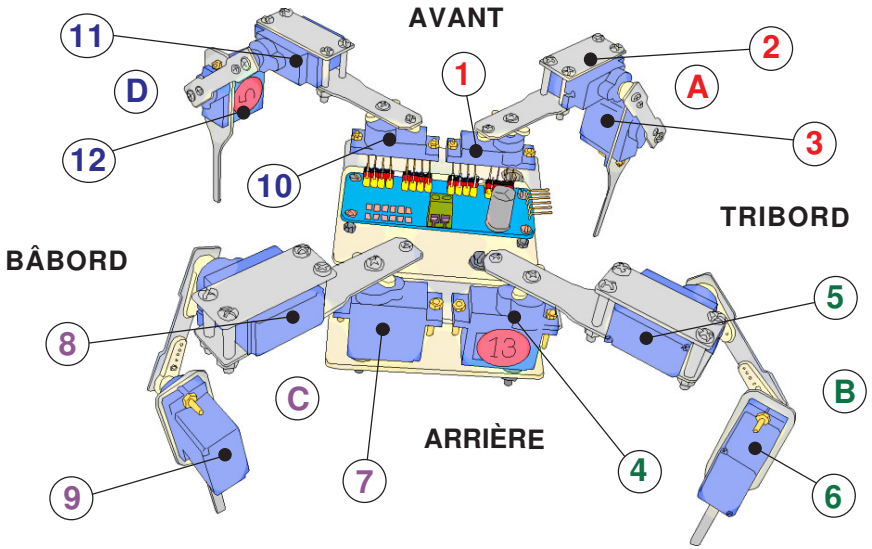
Consignes pour les postures

Jambes	A			B			C			D		
Configuration	1	2	3	4	5	6	7	8	9	10	11	12
Stable Rsble.	305	213	307	274	406	314	287	215	319	285	388	314
Zone inutilisée.												
Neutres opér.	320	308	316	289	303	314	262	292	310	300	285	307
Décollage.	211	152	315	400	418	232	159	139	255	411	463	325
Atterrissage.	303	490	154	303	141	481	303	472	145	303	134	501
Stable Transversal.	239	213	307	344	414	314	222	215	319	346	388	314
Hauteur Maxi.	239	152	230	344	483	391	222	135	232	346	467	398
Prépare Reculer.	274	213	307	379	414	314	187	215	319	311	388	314
JEKER posée au sol.	239	315	421	344	296	192	222	298	426	346	391	199
Pointage LASER.	312	314	135	344	414	314	222	215	319	171	388	314
Retour conf. LASER.	239	213	307	344	414	314	222	215	319	171	388	314
Prépare Avancer.	204	213	307	309	414	314	257	215	319	381	388	314
Avant Hauteur Maxi.	239	264	449	344	347	167	222	262	464	346	312	158
Pour Tourner à Droite.	269	213	307	374	414	314	252	215	319	376	388	314
" " Tourner à Gauche.	209	213	307	314	414	314	192	215	319	316	388	314
Décaler à Droite.	239	270	281	344	344	339	222	215	439	346	381	188
Décaler à Gauche.	239	230	430	344	392	190	222	258	287	346	334	346

Les POSTURES DE BASE en EEPROM.

P60_Clone_EEPROM_sonde.ino

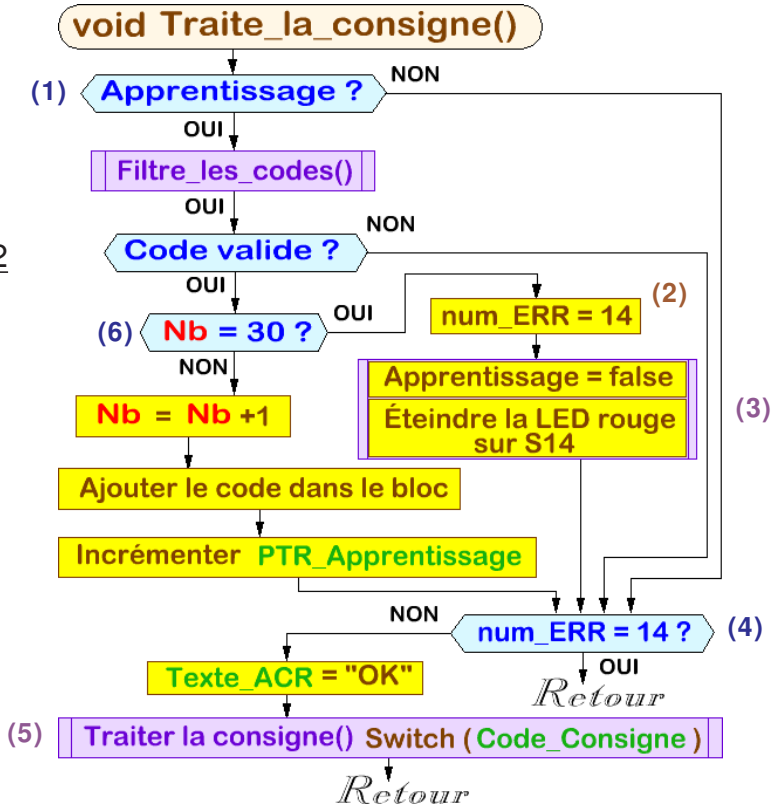
Adresse	Bloc	POSTURE
000	01	Stable Raisonnable.
024	02	Zone inutilisée et disponible.
048	03	Moteurs aux neutres opérationnels.
072	04	Configuration décollage.
096	05	Configuration Atterrissage.
120	06	Stable Transversal.
144	07	Hauteur maximale.
168	08	Configuration Avancée pour Reculer.
192	09	VEILLE. *(Sonde posée sur le sol.)*
216	10	Pointage LASER.
240	11	Retour de Pointage LASER.
264	12	Configuration Reculée pour Avancer.
288	13	Configuration pour avant hauteur maximale.
312	14	Tourner à Droite.
336	15	Tourner à Gauche.
360	16	Décaler à Droite.
384	17	Décaler à Gauche.
408	---	début des postures personnalisées.



Traitement des consignes en mode apprentissage.

Quand le mode est actif et que les deux LEDs rouges sont allumées, chaque changement d'item quand on tourne le codeur rotatif fait réafficher le titre APPRENTsge dans le rectangle jaune, il ne faut pas s'en étonner. La Fig.2 présente l'organigramme de la procédure qui traite les codes de consigne arrivant sur RX.

Fig.2



Dans le cas où le mode apprentissage n'est pas actif, le test (1) branche en (4). À la réception du message arrivant sur RX num_ERR a été forcé à zéro. Le programme poursuit en (5) et ne fait rien d'autre que de traiter la consigne. Si l'apprentissage est en cours, le code est filtré. S'il n'est pas valide le processeur est branché en (4) qui à son tour déroute en (5) pour traiter la commande.
Si le mode apprentissage est actif ET que le code est valide pour être inscrit en EEPROM le logiciel commence par vérifier en (6) que le bloc indexé n'est pas saturé à trente instructions.

... / ...

Prepier cas : Le bloc pointé en EEPROM n'est pas saturé : Dans ce cas la cellule qui indique le nombre d'instructions est incrémentée. Puis le Code_Consigne est inscrit en EEPROM à l'adresse contenue dans la variable PTR_Apprentissage. Cet identificateur pointe en permanence la prochaine cellule disponible pour le programme dans le bloc indexé. Pour qu'il en soit ainsi, PTR_Apprentissage est alors incrémenté d'une unité puisque les codes sont de format byte. L'instruction étant sauvegardée en EEPROM, l'ATmega328 est alors branché en (4) qui, comme pour les divers cas déjà analysés déroute en (5) pour traiter la commande.

(Pour les deux cas voir la Fig.2 donnée en page précédente.)

Deuxième cas : Le bloc pointé en EEPROM est saturé : La variable num_ERR est forcée à E14 valeur significative de l'impossibilité d'ajouter des instructions dans le bloc indexé. Le mode apprentissage est alors interrompu en (3) par la procédure Fin_Apprentissage(). Pour que l'opérateur en soit averti les deux LEDs rouges sont éteintes, et sur le pupitre E14 est complété par un BIP sonore. Le test (4) devient alors positif, entrainant la sortie de la procédure void Traite_la_consigne(). L'instruction arrivée sur RX est ignorée. Ainsi, l'opérateur peut à sa guise désigner un autre bloc programme en EEPROM et reprendre l'apprentissage.

Enchaînement de plusieurs programmes.

Présenté sous forme d'un organigramme en Fig.3 la séquence sur le pupitre qui permet d'exécuter en cascade plusieurs blocs enregistrés en EEPROM est déroulée si le test en (1) est positif. Si le menu APPRENTsge est en cours, en (2) on teste si l'option "déclencher un programme enregistré" vient d'être validée. Si c'est le cas, en (3) on teste SÉCURITÉ. Activer un programme est illégal en mode apprentissage, le test (4) effectue la vérification. Toutes les conditions étant réunies, il y a appel à void Enchainer_les_PGMs(). Dès qu'un programme vide sera détecté en (5) on quittera la procédure en forçant en (6) à 1 le nombre de programmes à chainer. À chaque changement de bloc on affiche le listage des instructions à l'écran en (7). Puis en (8) les instructions du bloc sont réalisées les unes après les autres en cochant les cellules dans la grille de listage. C'est la procédure Realiser_un_PGM() qui stoppera immédiatement le mode exécution si un incident se produit durant la tentative de réaliser le code

... / ...

Gestion du pointage LASER.

Procédant par des calculs de valeurs de consignes envoyées aux servomoteurs, pour que la machine réagisse à pratiquement tous les pas positifs ou négatifs du codeur incrémental, il faut des variations de consigne d'au moins deux unités. C'est le pas adopté lorsque les mouvements "fins" sont désirés.

En option "déplacements rapides", on adopte des variations six fois plus importantes. Les incréments / décréments de consigne sont donc de 12 unités. Des butées de mouvement logicielles génèrent les erreurs E9 et E10 avec allumage des LEDs dédiées sur la SONDE.

(Si la ligne n'est pas disjonctée et que le "strap" des témoins lumineux est en place sur le connecteur HE14.)

Les deux dessins indiquent en violet les valeurs des consignes à envoyer aux servomoteurs de la Jambe **A** pour obtenir les postures de départ ainsi que celles qui déclenchent les alertes E9 et E910.

La Fig.1 présente le membre en vue de profil. Le Tibia pour sa position neutre de départ doit recevoir la consigne 312. *(Orientation horizontale en situation initiale.)*

La Fig.2 présente la Jambe **A** en vue de dessus. La Hanche pour sa position neutre de départ à 45° doit recevoir la valeur de consigne 305.

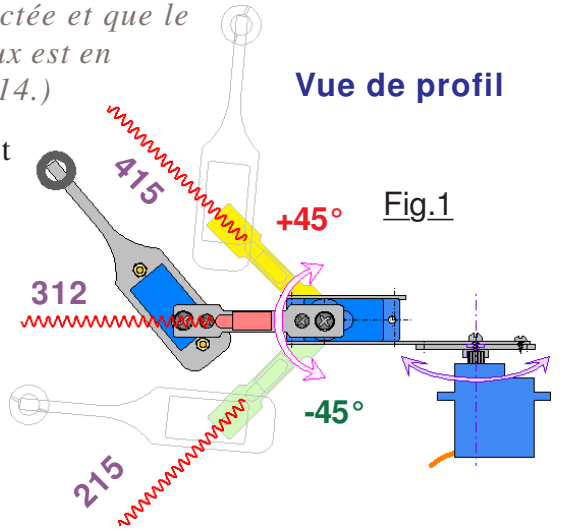


Fig.1

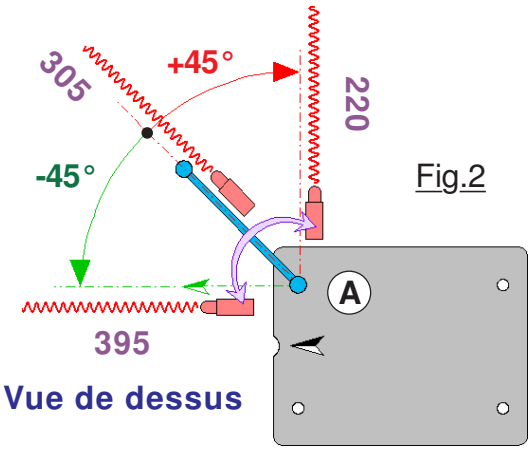


Fig.2

Plaque cuivrée du circuit de recharge.

Prise 6Vcc vue coté soudures.

(Explications en page 19.)

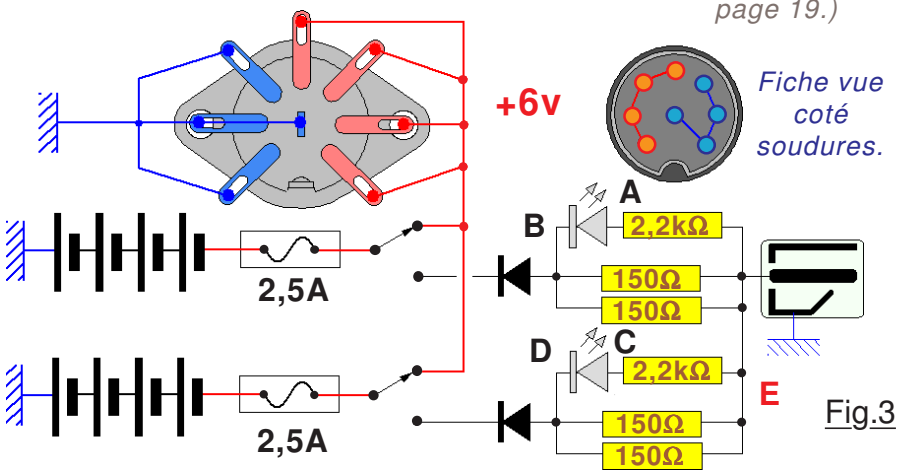


Fig.3

ATTENTION : Surtout ne pas utiliser des fusibles rapides.

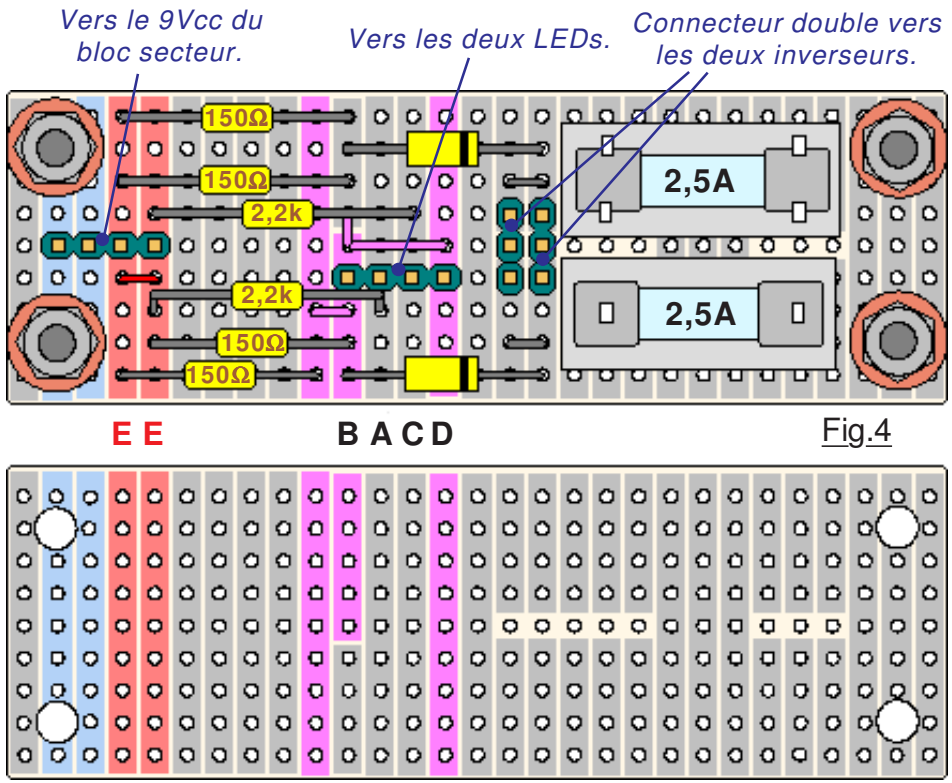


Fig.4

... / ...

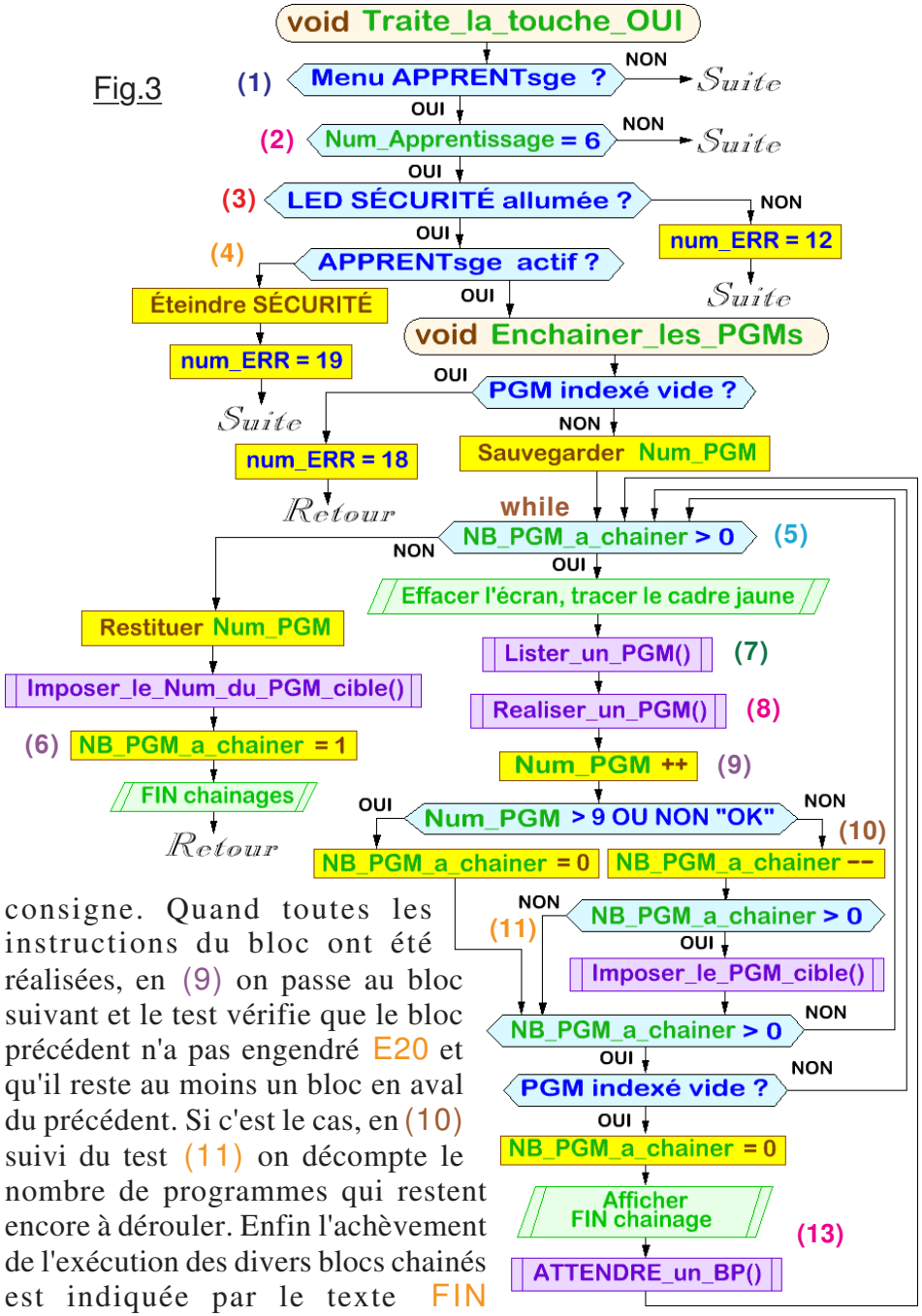


Fig.3

consigne. Quand toutes les instructions du bloc ont été réalisées, en (9) on passe au bloc suivant et le test vérifie que le bloc précédent n'a pas engendré E20 et qu'il reste au moins un bloc en aval du précédent. Si c'est le cas, en (10) suivi du test (11) on décompte le nombre de programmes qui restent encore à dérouler. Enfin l'achèvement de l'exécution des divers blocs chainés est indiquée par le texte FIN chainage dans le cadre jaune, suivi de l'attente d'une touche au clavier.

Protocole pour recharger les accumulateurs.

Important : **La sonde doit être entièrement débranchée**. C'est à dire que la ligne d'alimentation en puissance et le cordon ombilical de dialogue sont déconnectés. Le pupitre fonctionne en chronomètre pour déterminer quand l'accumulateur est entièrement rechargé. Compte tenu de la pile utilisée et du courant de recharge injecté, il faut en théorie 40H pour une réalimentation complète des éléments. Le chronomètre est prévu pour décompter durant 40H en **A**. En **B** un chronomètre partiel recycle toutes les minutes et indique quand **A** va être décrémenté.

01) Vérifier que ***l'interrupteur qui alimente le pupitre est coupé*** >
02) Relier la prise USB de NANO Arduino à un bloc secteur USB >
03) Brancher le module USB sur le 220V~ qui alimente le pupitre >
04) Activer les ***deux inverseurs de l'alimentation de puissance***.
05) Brancher le module transformateur basse tension 9V de rechargement sur la sonde puis sur le secteur 220V~ >
06) Couper les ***deux inverseurs de l'alimentation de puissance***. Les deux LED blanches de rechargement doivent alors s'illuminer.
07) Cliquer sur le BP *"**Avancer**"* pour activer le chronomètre >
08) ***À tout moment cliquer sur un BP quelconque éteindra l'écran ou réaffichera les valeurs des chronomètres de décomptage***.

Durant la charge la LED rouge "pulse" toutes les secondes. Quand le chronomètre arrive à zéro, la LED rouge est alors éteinte et la LED bleue s'allume signalant une durée de recharge entièrement écoulée.

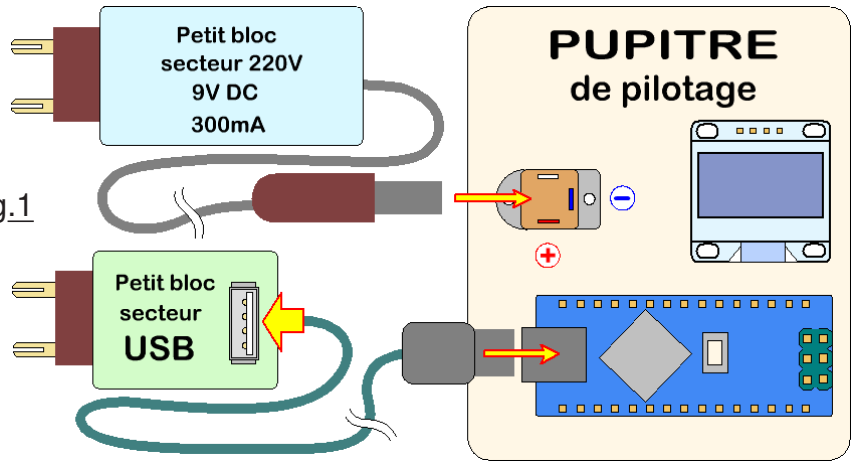


Fig.1

09) Débrancher du secteur les deux petits modules 9V DC et USB puis les deux fiches sur le pupitre de pilotage.

Alimentation en puissance 6Vcc autonome.

Utilisant deux blocs de batteries rechargeables de 6V 2300mAh le schéma électrique donné sur la Fig.2 explique la particularité du protocole de rechargement proposé en page précédente. En effet, couper les deux inverseurs **X** et **Y** pour activer la recharge n'est pas très ordinaire. Bien que réputé 9Vcc nominal, à vide le bloc secteur délivre une tension d'environ 14Vcc. La tension chute à 12V quand l'un des accumulateurs est en charge. Le courant en ligne est alors de 52mA. Lorsque les deux modules sont en charge, la tension chute à environ 11Vcc et surtout le courant de charge se sature à environ 75mA. On peut estimer alors que chaque bloc de piles rechargeables reçoit approximativement 40mA.
Constituée de deux résistances de 150Ω mises en parallèles, la résistance de charge réelle fait 75Ω. *(C'est la disponibilité de ces composants qui a justifié leur emploi.)* La chute de tension aux bornes de cette résistance de charge de 75Ω est ≈ 4,1V soit un courant de 54mA, plus le faible débit traversant la LED blanche.
La mise en parallèle des deux blocs d'accumulateurs lors de l'activation des deux inverseurs ne pose aucun problème, le courant circulant de l'un vers l'autre restant dérisoire. Toutefois, lorsque la machine est remisée durant de longues périodes, ce faible courant finirait par décharger lentement les éléments. C'est la raison pour laquelle les deux diodes **D** interdisent une liaison permanente via les résistances de charge.

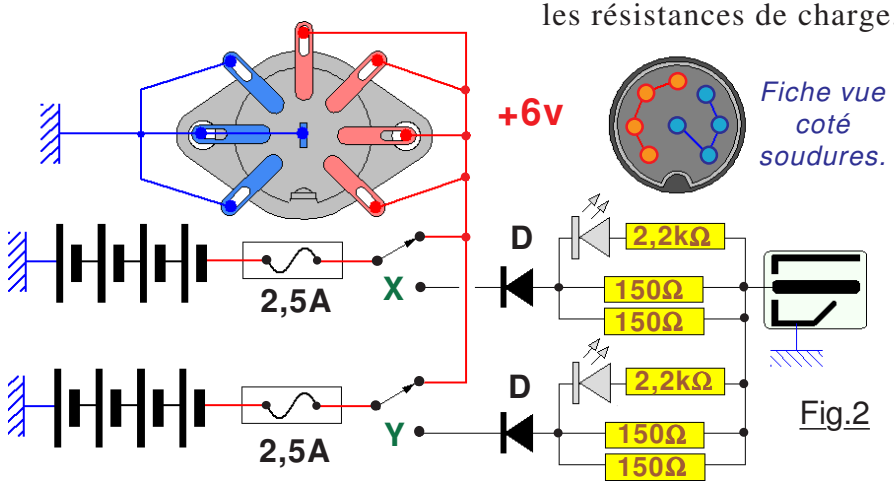


Fig.2

... / ...