

## Le BMMJ : le Block Manuel Module Junior Décryptage ....

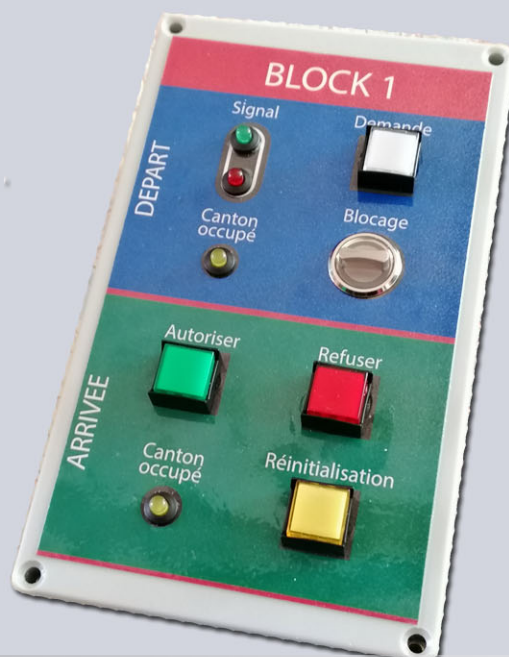
En s'appuyant sur des travaux réalisés pour des écoles de la SNCF, sur le thème du cantonnement des lignes à voie unique (BMVU – Block Manuel de Voie Unique), l'idée de décliner ce principe pour exploiter nos réseaux en Module Junior a fait son chemin. Baptisé BMMJ (Block Manuel Module Junior), ce système est composé notamment d'un bus informatique pour l'échange des données entre les deux boîtiers placés aux extrémités du canton (entre deux gares).

Afin d'équiper les associations d'un BMMJ pour leur réseau Module Junior, une étude technique, utilisant les microcontrôleurs Arduino a été développée sur le Comité Nord pour préparer l'animation d'un stage. Quatre associations (Mouy-Bury, Compiègne, Amiens, et Tergnier) vont réaliser un ensemble de deux boîtiers munis de cartes électroniques pour équiper leurs modules. Le matin a été consacré au câblage des pupitres de chaque boîtier et la réalisation des cartes. L'après-midi a été réservée pour vérifier le fonctionnement de chaque ensemble.

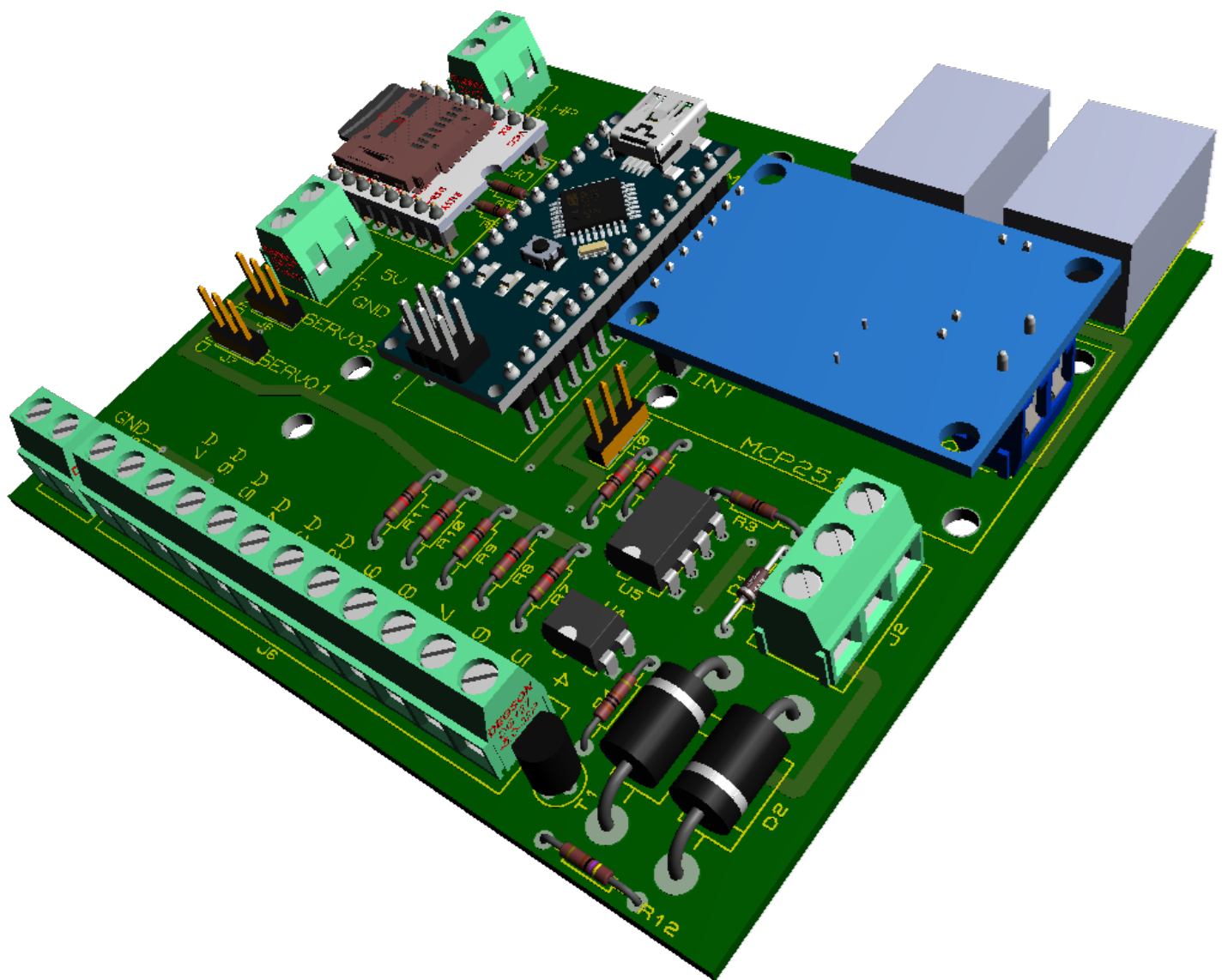
Sur le plan technique, deux sous-ensembles dialoguent grâce à un bus et assure l'alimentation de toutes les cartes avec un câble RJ d'une longueur de 10 mètres environ (ce qui correspond à la longueur maximale du canton). Un

premier sous-ensemble, placé sous chaque module d'entrée du canton et relié aux deux files de rail, permet d'assurer la détection d'un train entrant dans le canton pour provoquer la fermeture du signal lumineux (ou d'un signal mécanique utilisant un servomoteur). Le deuxième sous-ensemble concerne les boîtiers du block (les pupitres pour effectuer les opérations de cantonnement). Sur chaque boîtier, un bouton permet de « bloquer » le canton pour engager une évolution (ou une desserte), c'est-à-dire, d'envoyer un train seulement vers un embranchement particulier situé dans le canton, puis de revenir à sa gare de départ.

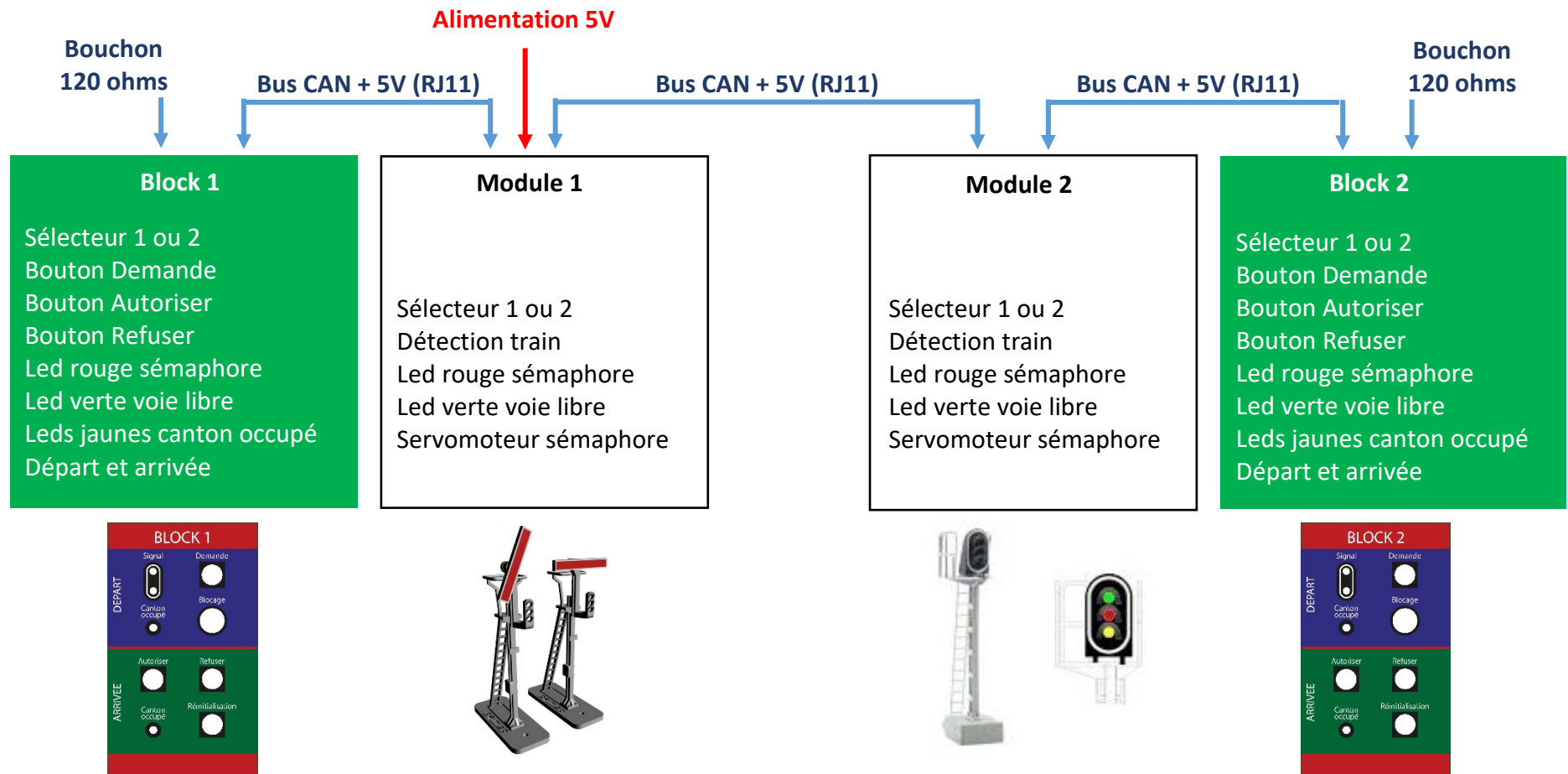
Romain Lecomte et Pierre Lherbon



# Block Manuel Module Junior v1



## Principe



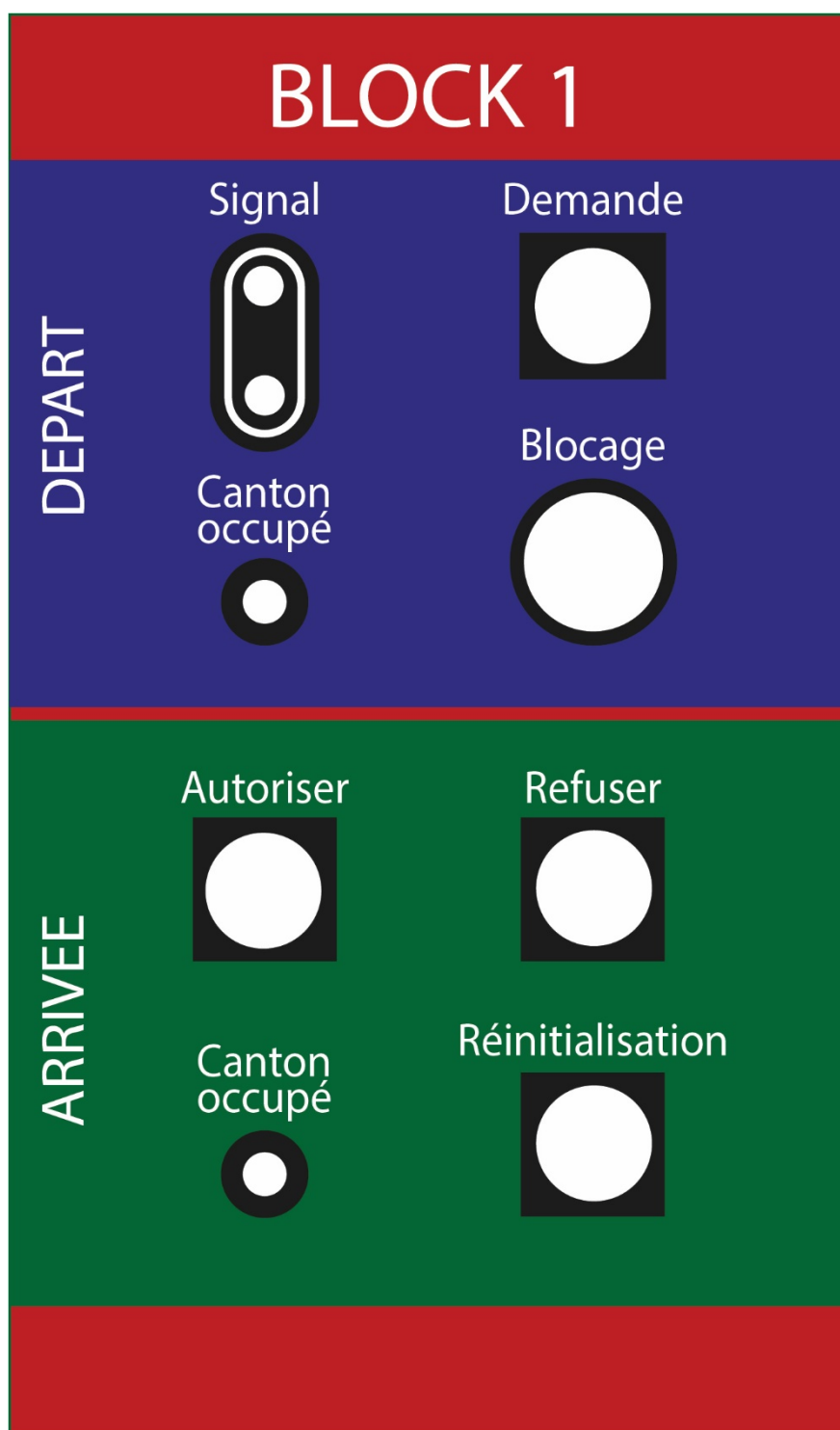
Avec ou sans module les 2 blocks communiquent entre eux via des messages normés sur le bus CAN

L'électronique et le câblage est standard : un cavalier permet de sélectionner si c'est le block 1 ou 2 dans le câblage. Sur le bus CAN un block doit avoir l'adresse 1 et l'autre en adresse 2 pour s'envoyer respectivement les messages.

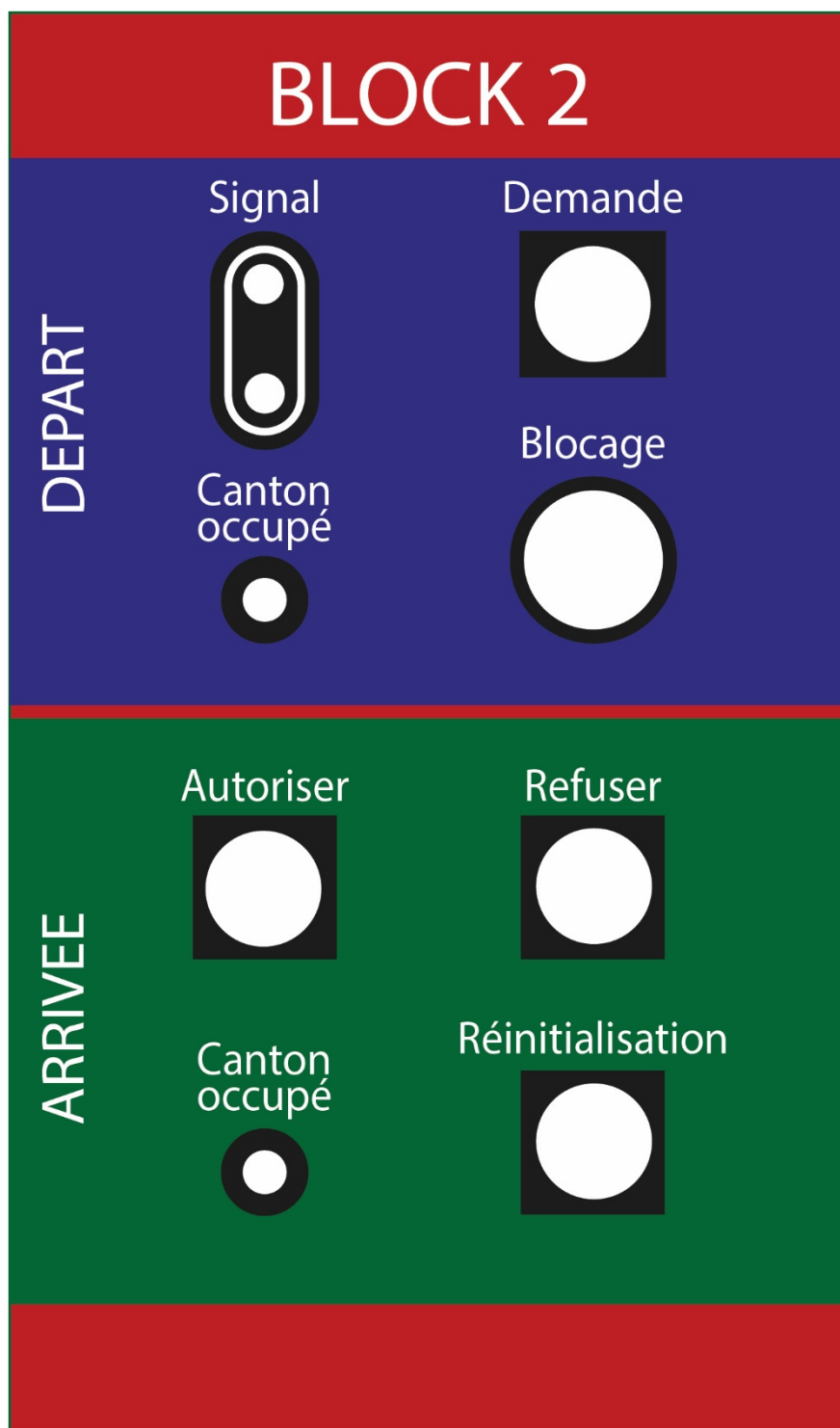
Câblage bus CAN extensible jusqu'à 100 mètres sur 2 fils via câble RJ11. Pour ne pas avoir de perturbations sur la ligne, il est nécessaire en extrémité de shunter les deux fils CAN\_H et CAN\_L par des résistances de 120 ohms (possibilité de les dissimuler sur le bornier des cartes MCP2515 dans les appareils de block).

Une seule alimentation 5V est nécessaire pour l'ensemble, le 5V est ensuite distribué par les câbles RJ11

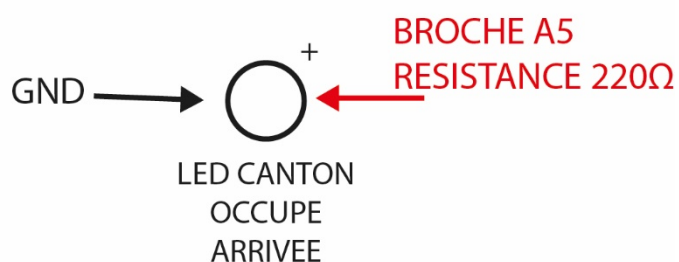
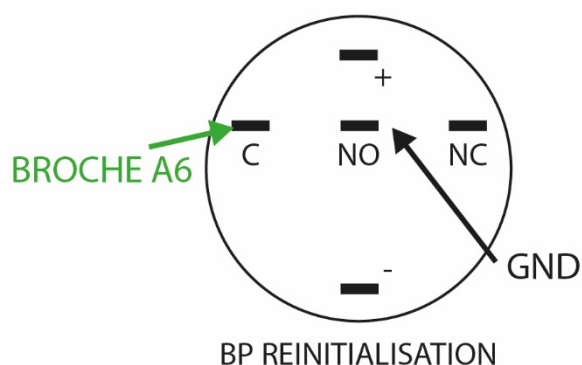
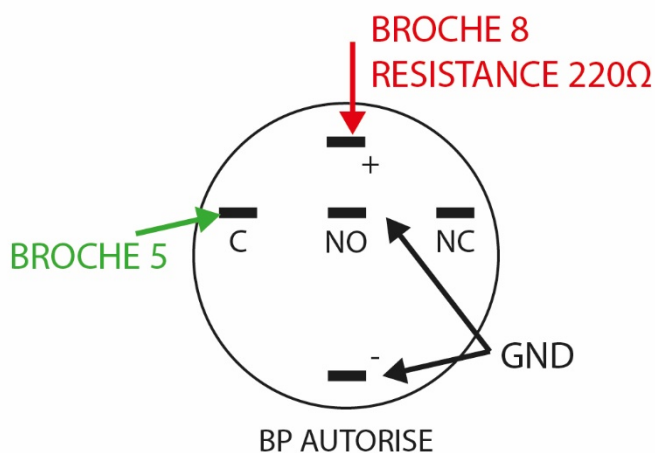
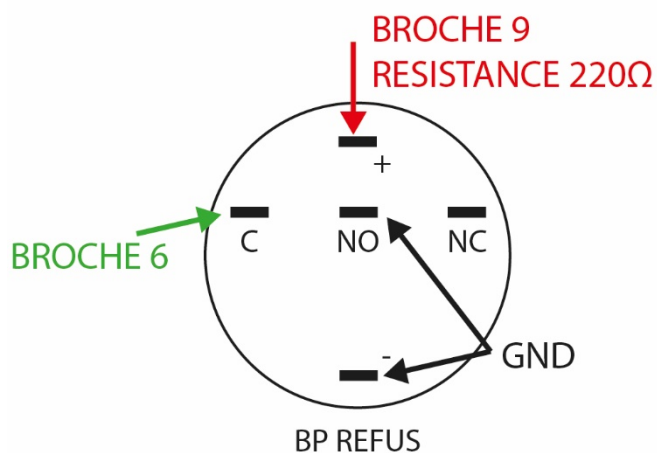
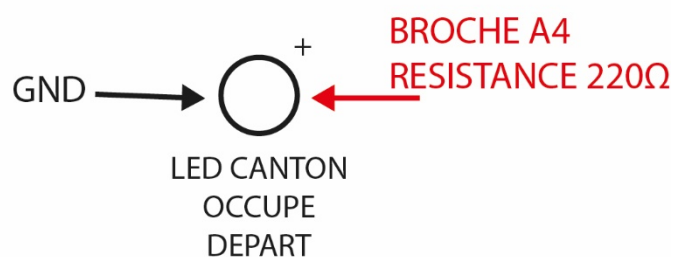
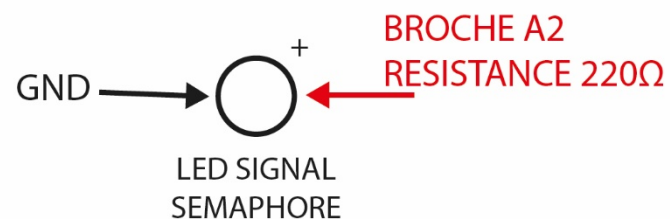
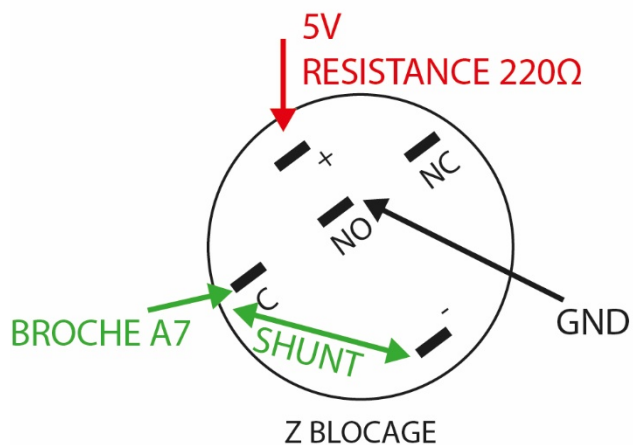
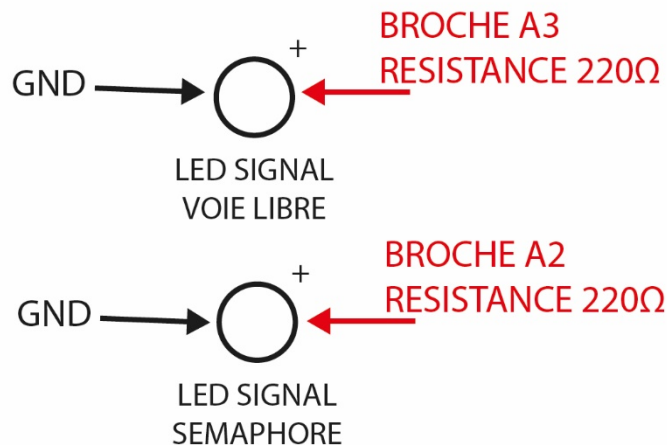
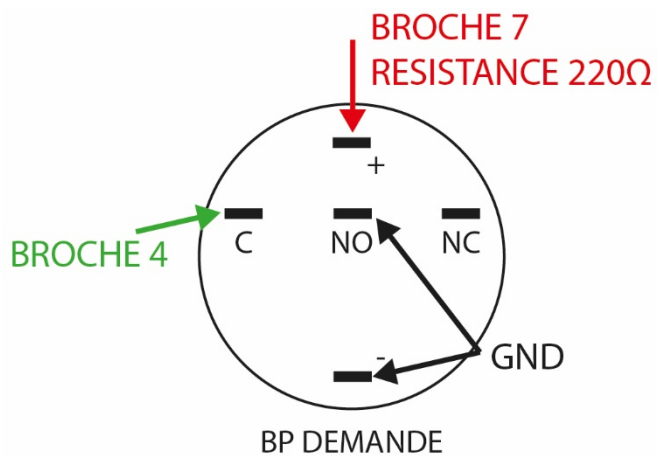
## Infographie du block 1



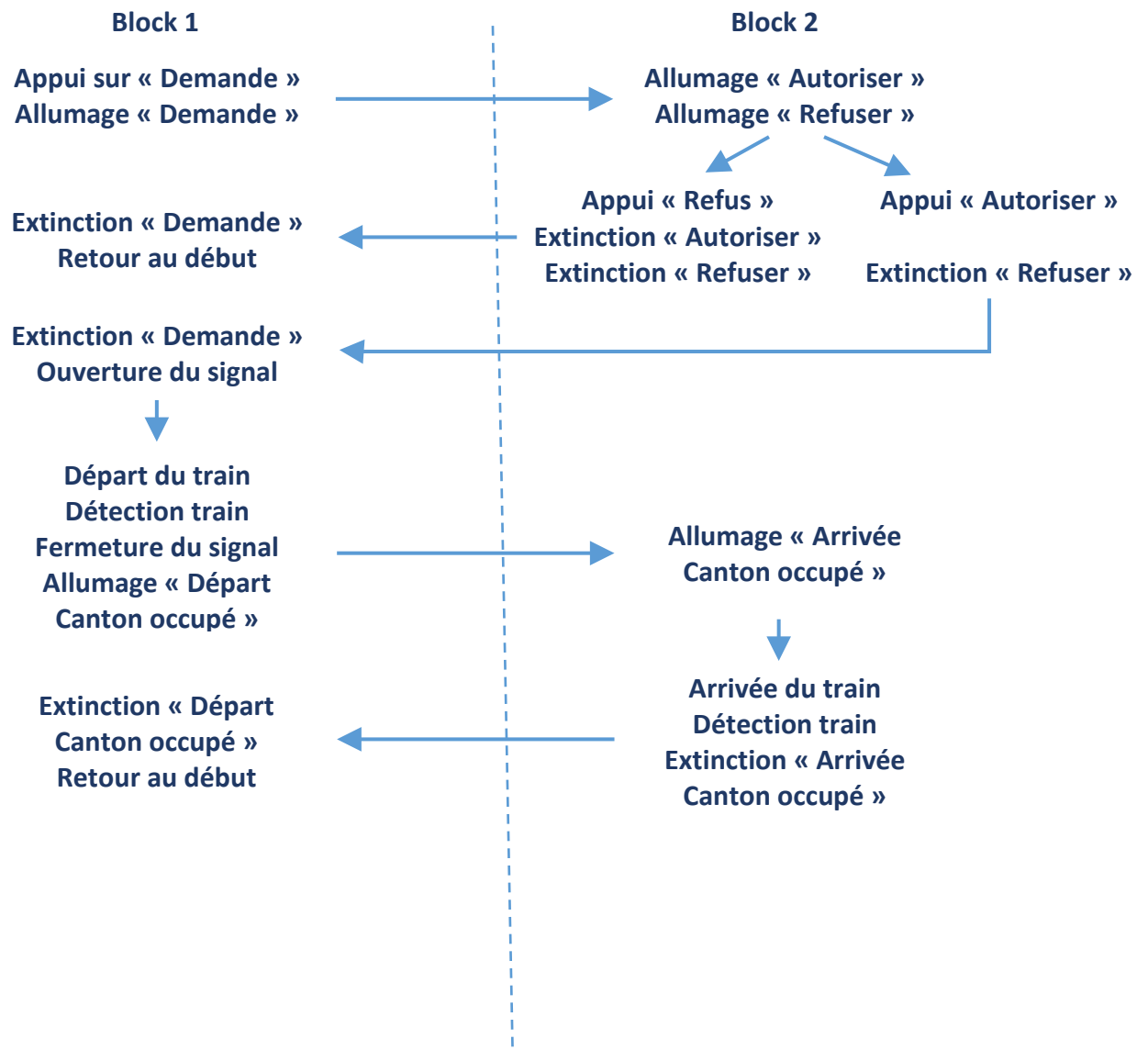
## Infographie du block 2







## Principe de fonctionnement normal pour expédier un train de 1 vers 2



## Lexique des trames CAN

Trame	Grafcet	Variable
3 : 01 01 01	BP(DEMANDE)1 Actionné	bp_demande_1_sw = 1
3 : 01 01 02	BP(DEMANDE)1 Relaché	bp_demande_1_sw = 0
3 : 01 01 03	Allumer BP(DEMANDE)1	bp_demande_1_led = 1
3 : 01 01 04	Eteindre BP(DEMANDE)1	bp_demande_1_led = 0
3 : 01 02 01	BP(ACCORD)1 Actionné	bp_accord_1_sw = 1
3 : 01 02 02	BP(ACCORD)1 Relaché	bp_accord_1_sw = 0
3 : 01 02 03	Allumer BP(ACCORD)1	bp_accord_1_led = 1
3 : 01 02 04	Eteindre BP(ACCORD)1	bp_accord_1_led = 0
3 : 01 03 01	BP(REFUS)1 Actionné	bp_refus_1_sw = 1
3 : 01 03 02	BP(REFUS)1 Relaché	bp_refus_1_sw = 0
3 : 01 03 03	Allumer BP(REFUS)1	bp_refus_1_led = 1
3 : 01 03 04	Eteindre BP(REFUS)1	bp_refus_1_led = 0
3 : 01 04 03	Allumer LS(ENVOI)1	ls_envoi_1_led = 1
3 : 01 04 04	Eteindre LS(ENVOI)1	ls_envoi_1_led = 0
3 : 01 05 03	Allumer LS(RECEPTION)1	ls_reception_1_led = 1
3 : 01 05 04	Eteindre LS(RECEPTION)1	ls_reception_1_led = 0
1 : 01 06 03	Allumer SEMAPHORE(S)1	semaphore_s_1_led = 1
1 : 01 06 04	Eteindre SEMAPHORE(S)1	semaphore_s_1_led = 0
1 : 01 07 03	Allumer SEMAPHORE(VL)1	semaphore_vl_1_led = 1
1 : 01 07 04	Eteindre SEMAPHORE(VL)1	semaphore_vl_1_led = 0
2 : 01 08 01	PEDALE(1) Actionné	pedale_1_sw = 1
2 : 01 08 02	PEDALE(1) Relaché	pedale_1_sw = 0



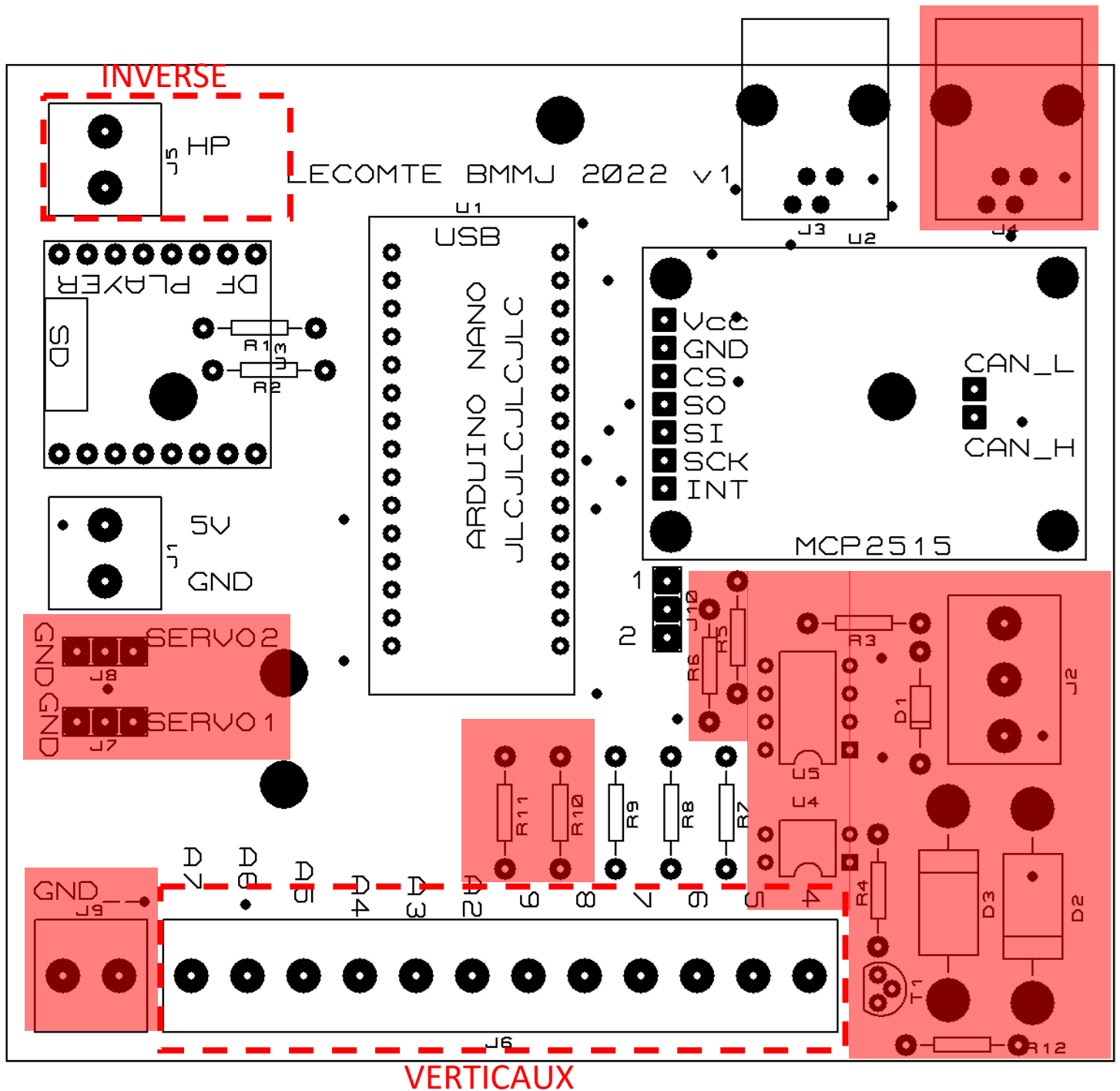
Trame	Grafcet	Variable
3 : 02 01 01	BP(DEMANDE)2 Actionné	bp_demande_2_sw = 1
3 : 02 01 02	BP(DEMANDE)2 Relaché	bp_demande_2_sw = 0
3 : 02 01 03	Allumer BP(DEMANDE)2	bp_demande_2_led = 1
3 : 02 01 04	Eteindre BP(DEMANDE)2	bp_demande_2_led = 0
3 : 02 02 01	BP(ACCORD)2 Actionné	bp_accord_2_sw = 1
3 : 02 02 02	BP(ACCORD)2 Relaché	bp_accord_2_sw = 0
3 : 02 02 03	Allumer BP(ACCORD)2	bp_accord_2_led = 1
3 : 02 02 04	Eteindre BP(ACCORD)2	bp_accord_2_led = 0
3 : 02 03 01	BP(REFUS)2 Actionné	bp_refus_2_sw = 1
3 : 02 03 02	BP(REFUS)2 Relaché	bp_refus_2_sw = 0
3 : 02 03 03	Allumer BP(REFUS)2	bp_refus_2_led = 1
3 : 02 03 04	Eteindre BP(REFUS)2	bp_refus_2_led = 0
3 : 02 04 03	Allumer LS(ENVOI)2	ls_envoi_2_led = 1
3 : 02 04 04	Eteindre LS(ENVOI)2	ls_envoi_2_led = 0
3 : 02 05 03	Allumer LS(RECEPTION)2	ls_reception_2_led = 1
3 : 02 05 04	Eteindre LS(RECEPTION)2	ls_reception_2_led = 0
1 : 02 06 03	Allumer SEMAPHORE(S)2	semaphore_s_2_led = 1
1 : 02 06 04	Eteindre SEMAPHORE(S)2	semaphore_s_2_led = 0
1 : 02 07 03	Allumer SEMAPHORE(VL)2	semaphore_vl_2_led = 1
1 : 02 07 04	Eteindre SEMAPHORE(VL)2	semaphore_vl_2_led = 0
2 : 02 08 01	PEDALE(2) Actionné	pedale_2_sw = 1
2 : 02 08 02	PEDALE(2) Relaché	pedale_2_sw = 0

## Implantation et câblage des composants

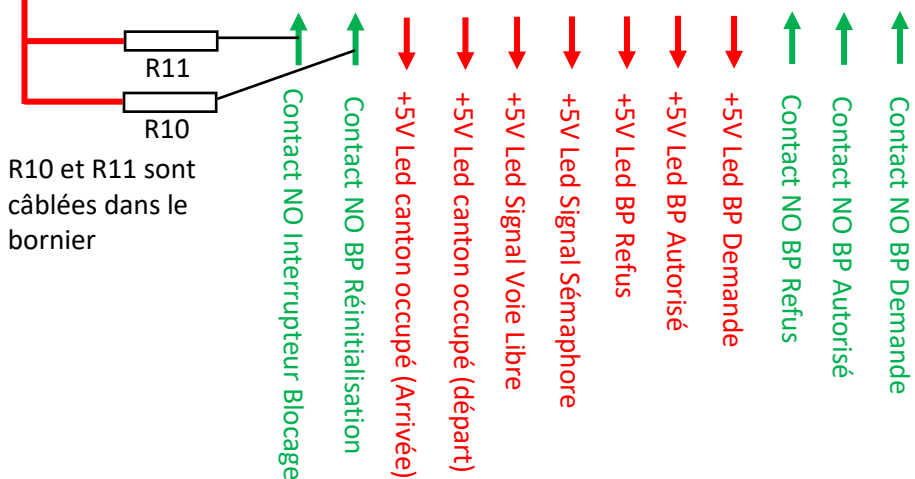
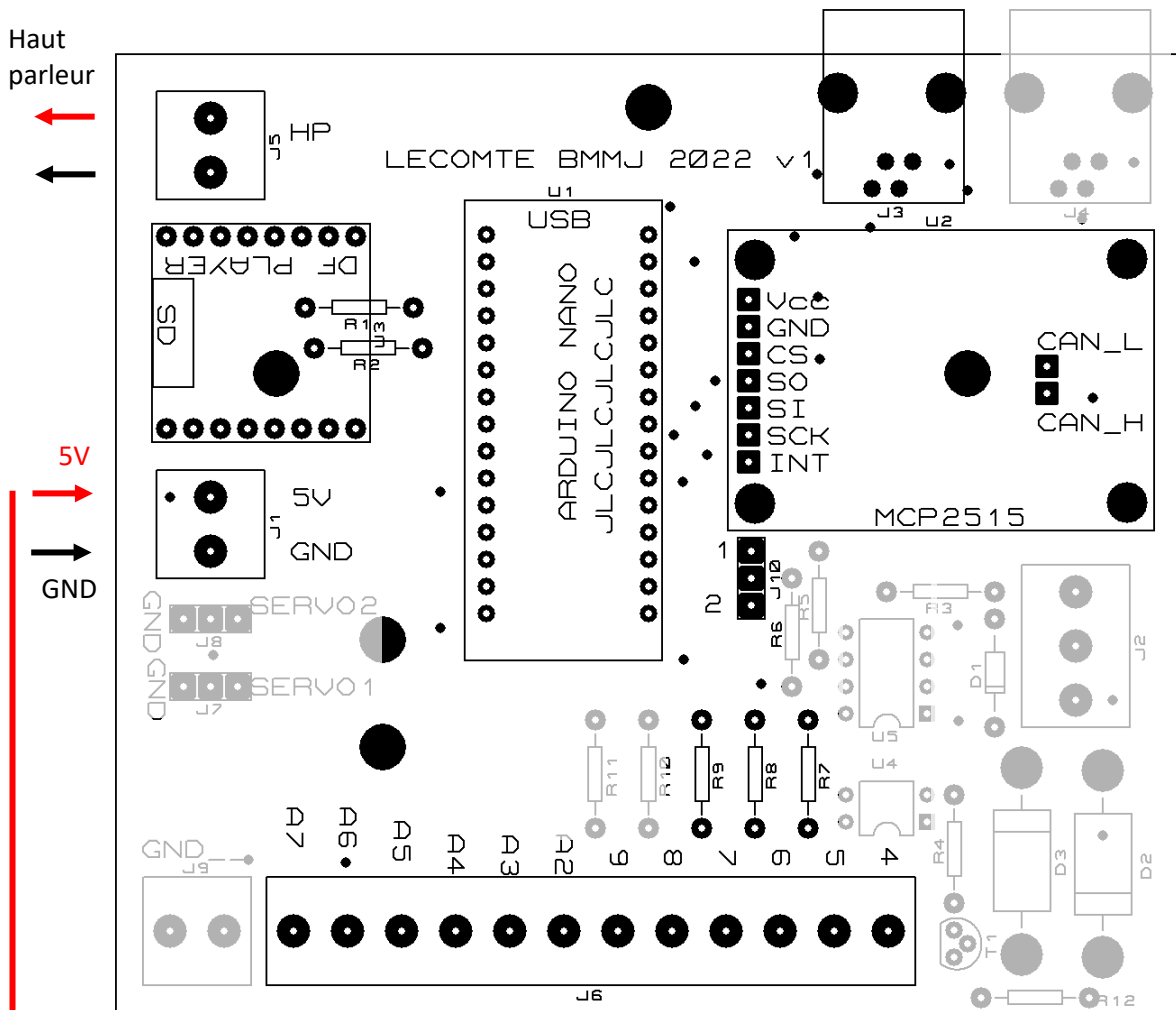
Les circuits imprimés étant standard, certains composants sont spécifiques au carte « Block » et d'autres pour les cartes « Modules »

### Cartes électroniques type « Block »

Les composants surlignés en rouge ne sont pas utilisés



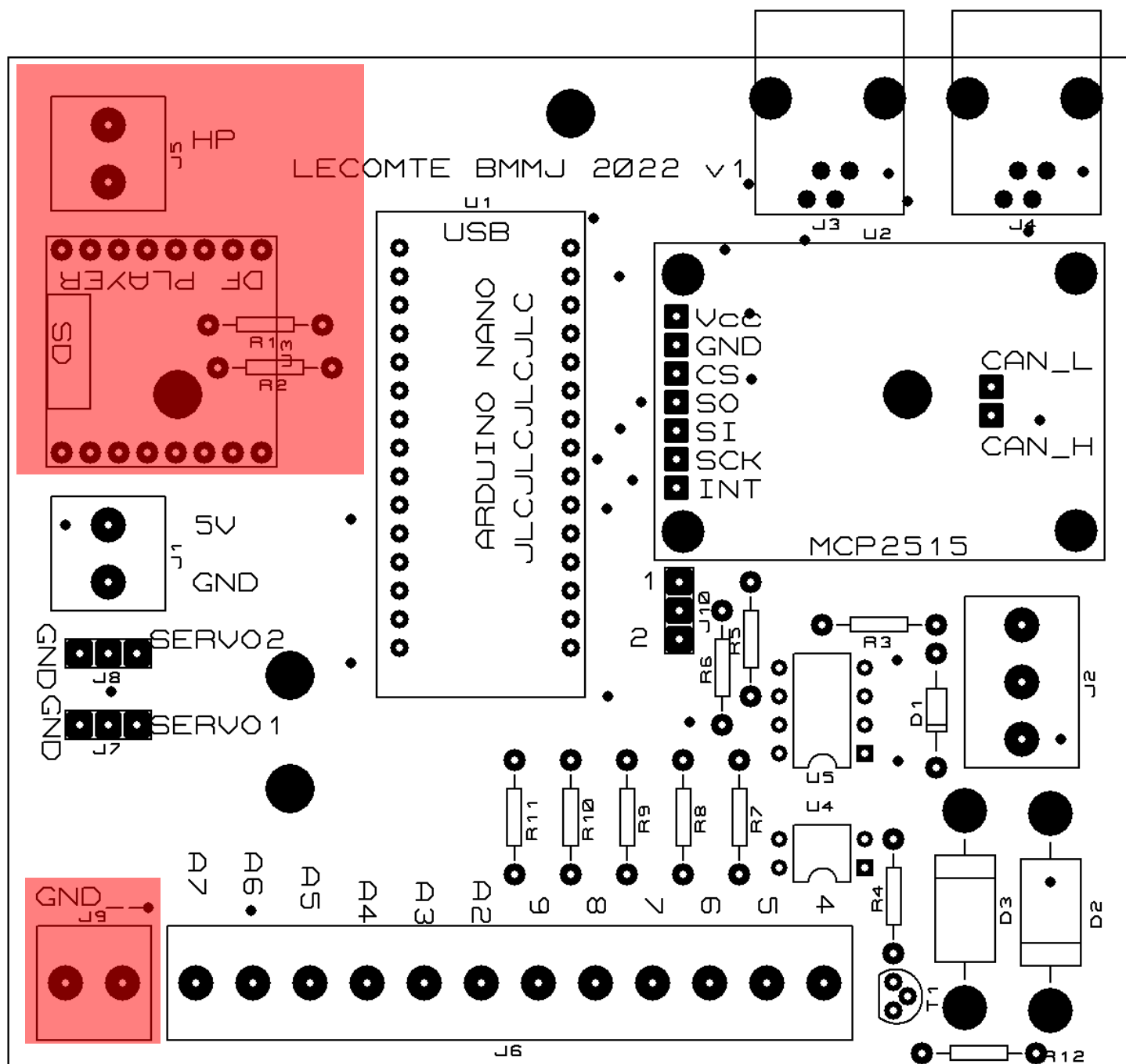
References	Sous fonction	Value
R1	SONNERIE	Résistance 1k
R2	SONNERIE	Résistance 1k
R7		Résistance 10k
R8		Résistance 10k
R9		Résistance 10k
R10		Résistance 10k
R11		Résistance 10k
U1		ARDUINO NANO
U2		MCP2515
U3	SONNERIE	DFPLAYER MINI
J1		Bornier 2 pôles
J5	SONNERIE	Bornier 2 pôles
J9		Bornier 2 pôles
J3		RJ11 6P4C
J4		RJ11 6P4C
J6		Bornier 2x 6 pôles
J10		Broches male 3 pôles



## Cartes électroniques type « Module »

Les composants surlignés en rouge ne sont pas utilisés

Les composants surlignés en jaune sont optionnels (pas de détection du train par consommation de courant et utilisation d'une cellule photoélectrique ou infrarouge)

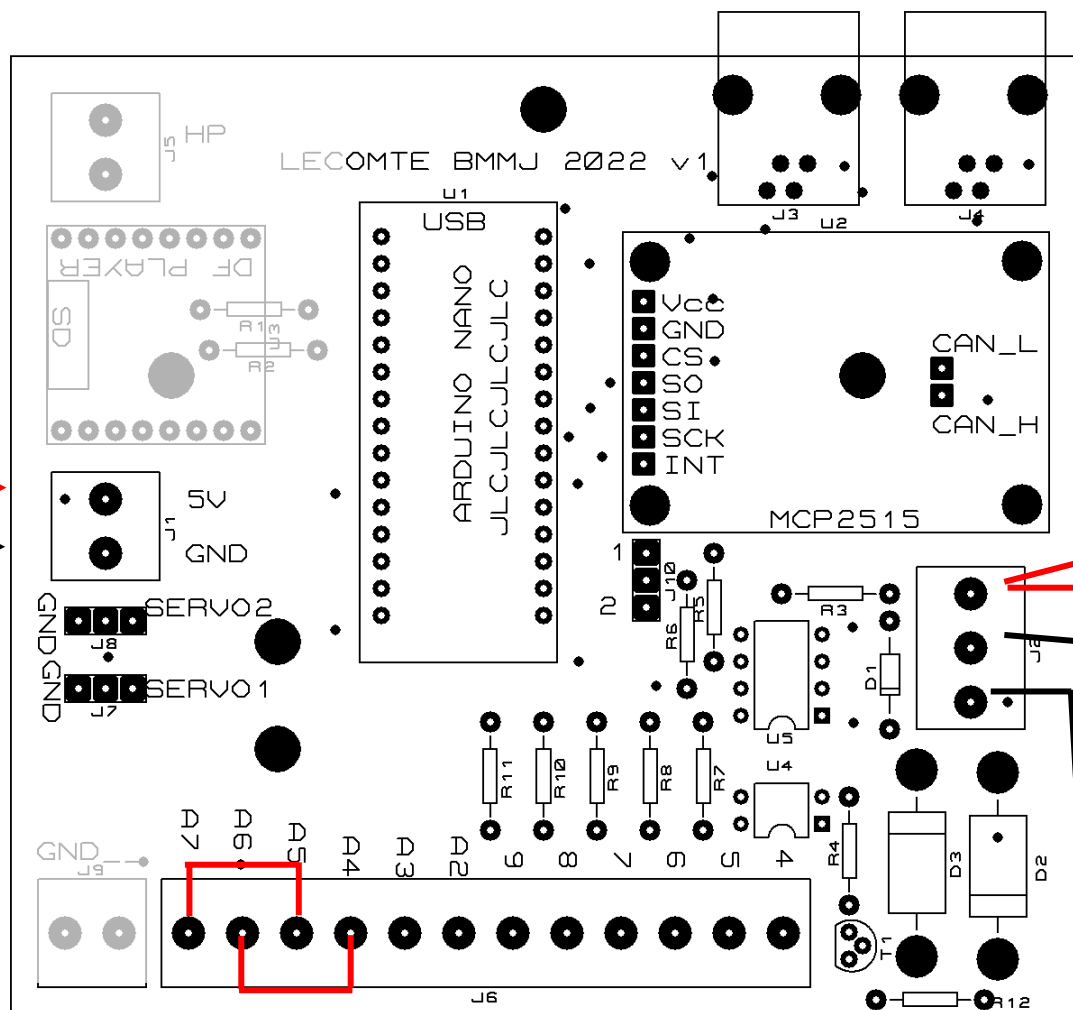


References	Sous fonction	Value
<b>R3</b>	DCC DECODEUR	Résistance 1k
<b>R4</b>	DCC DETECTION	Résistance 1k
<b>R5</b>	DCC DECODEUR	Résistance 10k
<b>R6</b>	DCC DECODEUR	Résistance 10k
<b>R7</b>		Résistance 10k
<b>R8</b>		Résistance 10k
<b>R9</b>		Résistance 10k
<b>R10</b>		Résistance 10k
<b>R11</b>		Résistance 10k
<b>R12</b>	DCC DETECTION	Résistance 4k7
<b>U1</b>		ARDUINO NANO
<b>U2</b>		MCP2515
<b>U4</b>	DCC DETECTION	PC817
<b>U5</b>	DCC DECODEUR	6N137
<b>D1</b>	DCC DECODEUR	1N4148
<b>D2</b>	DCC DETECTION	1N5400
<b>D3</b>	DCC DETECTION	1N5400
<b>J1</b>		Bornier 2 pôles
<b>J9</b>		Bornier 2 pôles
<b>J2</b>	DCC	Bornier 3 pôles
<b>J3</b>		RJ11 6P4C
<b>J4</b>		RJ11 6P4C
<b>J6</b>		Bornier 2x 6 pôles
<b>J7</b>	SERVO	Broches male 3 pôles
<b>J8</b>	SERVO	Broches male 3 pôles
<b>J10</b>		Broches male 3 pôles
<b>T1</b>	DCC DETECTION	BC547

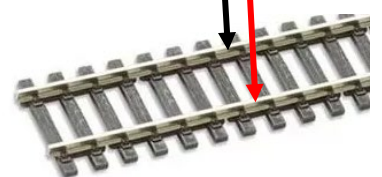


Jack alimentation uniquement module 1

5V  
GND



+5V Led Signal Sémaphore  
+5V Led Signal Voie libre  
Servo 1 (shunt pour prise)  
Servo 2 (shunt pour prise)



Zone détection  
Consommation courant

Prises bananas feeder