

## Rapport séance 7 février 2022

### Travail personnel

Afin de terminer la construction du robot, je me suis occupé de terminer la partie programmation. Le problème soulevé à la séance précédente était que les servomoteurs ne prenaient pas en compte l'état du bouton à chaque instant. Afin de le résoudre, j'ai modifié le programme du robot en obligeant d'une part à ce que le téléphone envoie des caractères spécifiques pour l'état de chacun des boutons, et d'autre part, j'ai imposé au programme de récupérer les caractères envoyés par la télécommande en permanence (et pas seulement lorsque une modification d'état est effectuée). Finalement le programme fonctionne. Le robot est donc terminé en construction et en programmation. Il est temps de tester sa capacité à grimper un obstacle.

Malheureusement, le robot en son état actuel ne semble pas être capable de franchir un obstacle. En effet on observe lors de la phase de test que lorsque l'on relève l'une des articulations avant ou arrière, le robot s'écrase sur lui-même. Il s'agit du fait que les servomoteurs augmentent énormément le poids central de l'appareil. Et que ces servomoteurs ne sont pas assez puissants pour pouvoir se maintenir dans leur position d'arrêt. De plus il est très probable que les axes en sortie des servomoteurs se plient et se tordent avec la pression qu'ils subissent ce qui amplifie ce phénomène d'écrasement. Le robot, au lieu de maintenir sa position en soulevant une articulation, bascule vers l'avant. Lorsque l'articulation quitte enfin le sol, la capacité de montée est largement diminuée et il n'est plus en position horizontale mais affiche un fort angle d'inclinaison vers le bas.

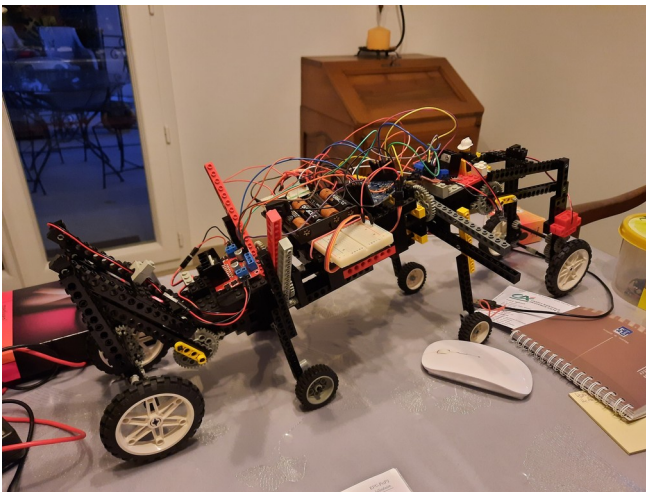


Image de la version complexe  
du robot

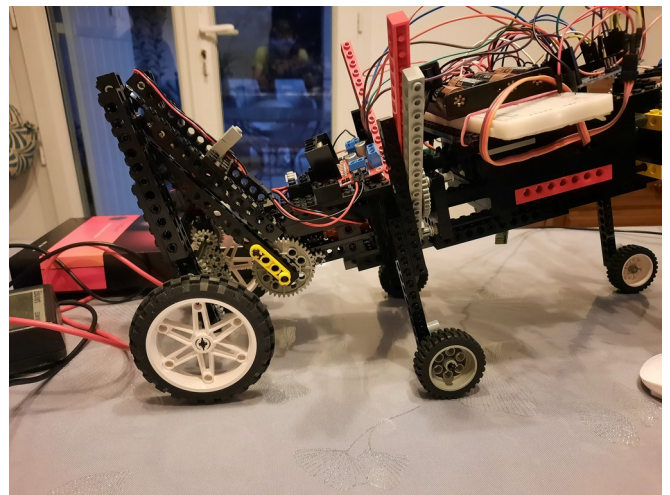


Image traduisant l'écrasement du robot  
lorsque le train avant tente de se soulever

Afin de pouvoir bloquer les servomoteurs dans leurs positions, il serait indispensable d'utiliser des vis sans fin. Hors elles ne s'adaptent pas au montage. Il ne sera pas possible de recourir à cette solution. Il va donc falloir restructurer le robot.

Après démontage de la partie électronique, j'ai commencé par retirer les deux servomoteurs et les articulations associées. Il s'agit ici de réduire le poids de la structure globale. Il ne reste ainsi que les articulations avant et arrière. Il faut ensuite restructurer celles-ci car avec l'absence du soutien des trains centraux, elles ne peuvent plus quitter le sol. J'opte donc pour modifier ces deux

articulations en retirant la partie verticale auxquelles sont rattachées les roues, pour les transformer en deux articulations équipées d'un seul axe avec deux paires de roues aux extrémités (une paires de roues avec fort diamètre, et une paire de roues plus petites). Lorsque les grandes roues sont au sol, les petites sont en l'air. Lors de la phase de montée, les petites roues, en basculant vers le sol, remplacent les grandes roues qui se soulèvent pour atteindre le haut de l'obstacle. Cela permet alors d'effectuer la phase de montée. Avec un système d'axe entre les deux paires de roues, on peut rendre motrices les 4 roues du système. En détachant le moteur de l'articulation arrière et en le rajoutant à l'avant, on obtient un moteur pour les roues de gauches et un autre pour les roues de droites (toujours sur le train avant). Il sera donc possible par la suite de faire tourner le robot dans différentes directions en faisant varier la vitesse de chaque roues.

A présent il faut rééquiper le robot de la carte et de l'alimentation. Puis réadapter la partie programmation. On note aussi qu'il sera nécessaire de modifier la télécommande du téléphone afin d'être adapté au nouveau robot et ses capacités.

### Objectif de la séance

1. câblage du nouveau robot
2. première programmation du robot
3. recherche de solutions afin de résoudre le problème des servomoteurs

### Partie 1 : câblage du nouveau robot

Pour commencer, il faut raccrocher la partie électronique au robot. J'ai tout d'abord retiré tous les câbles rattachés aux servomoteurs. Ensuite, j'ai branché les 4 moteurs aux deux ponts en H. Deux moteurs servent à la rotation des roues et les deux restants gèrent le mouvement des articulations. Afin de pouvoir modifier la vitesse de rotation des roues, j'ai relié les moteurs associés à des entrées PWM. Je n'ai pas installé l'électronique sur le robot mais je l'ai laissé de côté afin de pouvoir passer à la partie programmation.

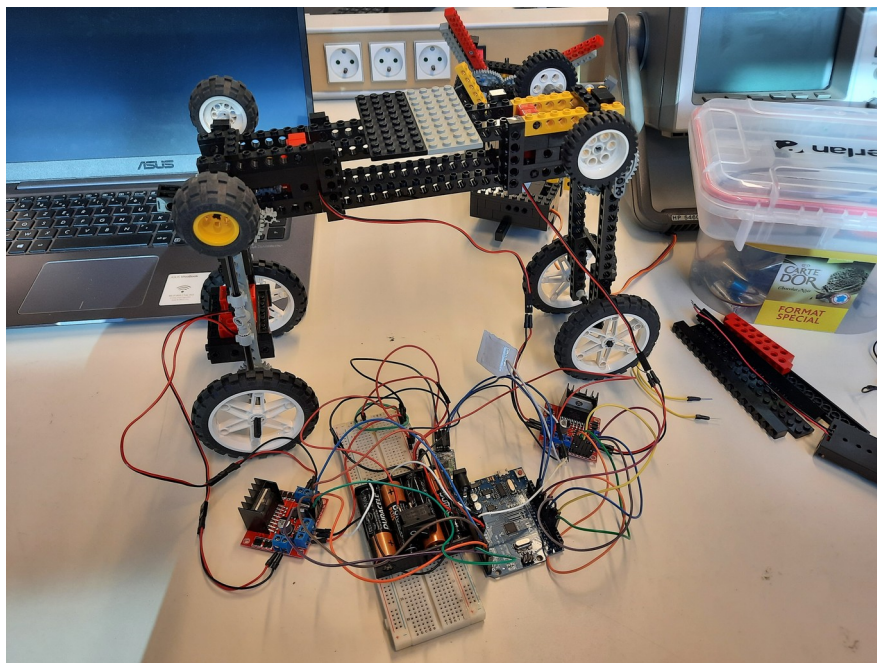


Image de la  
version simplifié  
du robot et de son  
câblage

## Partie 2 : première programmation du robot

Dans un 2<sup>e</sup> temps, je me suis attelé à programmer le robot. Cela a demandé de s'inspirer très fortement du code déjà établi pour la première version du robot. Afin de faciliter la compréhension, certains noms de variables ont été modifiés et la partie du code destinée aux servomoteurs a bien évidemment été retiré. Une nouvelle télécommande a été créée sur l'application avec un bouton en croix pour le contrôle de la marche avant et arrière ainsi que les virages. Puis 4 boutons ont été placés pour diriger le mouvement des articulations. Chaque bouton envoie par le biais du Bluetooth un caractère spécifique qui désigne son état.

J'ai d'abord programmé le mouvement des roues : avancer, s'arrêter et reculer. Puis je me suis occupé du mouvement des articulations : monter et descendre.

```
Robot2_Programme
1 // Programme robot n°2
2 //-----
3
4 #include<SoftwareSerial.h>
5 #define RX 13
6 #define TX 12
7
8 // Moteurs
9
10 // roues
11
12
13
14 int RDENA = 9 ;
15 int RDIN1 = 4 ;
16 int RDIN2 = 5 ;
17
18
19 int RGENB = 10 ;
20 int RGIN3 = 6 ;
21 int RGIN4 = 7 ;
22
23 // articulations
24
25
26 int ENA = 19 ;
27 int IN1 = 18 ;
28 int IN2 = 17 ;
29
30
31 int ENB = 14 ;
32 int IN3 = 16 ;
33 int IN4 = 15 ;
34
35 // Bluetooth
36
37 SoftwareSerial BlueT (RX, TX);
38
39
40 char Data;
41
42
43
44
45
46
47 //-----SETUP-----//
48 void setup() {
49 // put your setup code here, to run once:
50
51 Serial.begin(9600);
52 Serial.println("BEGIN");
53
54 // bluetooth
55 BlueT.begin(9600);
56
57 // Moteurs
58
59 /* RAPPEL
60 * ENA: puissance
61 * INx INy: sens
62 */
63
64 pinMode(RDENA, OUTPUT);
65 pinMode(RDIN1, OUTPUT);
66 pinMode(RDIN2, OUTPUT);
67
68 pinMode(RGENB, OUTPUT);
69 pinMode(RGIN3, OUTPUT);
70 pinMode(RGIN4, OUTPUT);
71
72 pinMode(ENA, OUTPUT);
73 pinMode(IN1, OUTPUT);
74 pinMode(IN2, OUTPUT);
75
76 pinMode(ENB, OUTPUT);
77 pinMode(IN3, OUTPUT);
78 pinMode(IN4, OUTPUT);
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

### Programme de contrôle du robot simplifié avec Arduino

#### Bilan test :

Après avoir téléversé le programme, le robot avance et recule normalement. Les articulations fonctionnent et le robot est capable de grimper sur le rebord de la table. Il est fonctionnel. Il suffira de charger dessus le matériel électronique ainsi que de programmer rapidement la rotation du robot pour effectuer des virages et il sera terminé .



Image du robot  
simplifié en étape de  
montée (avant  
câblage)

### Partie 3 : recherche de solutions afin de résoudre le problème des servomoteurs

Pour finir, j'ai recherché des solutions techniques afin d'essayer de rendre fonctionnel la première version du robot. Par suggestion du professeur, il sera possible d'en créer un deuxième à l'image de la première version. Pour cela il m'a fourni un double du matériel d'électronique : une carte, deux ponts en H, une platine d'essai, une alimentation et des câbles. L'idée ici est d'essayer de résoudre les problèmes rencontrés auparavant afin d'obtenir une version plus complexe et intéressante (version initiale). Les principaux problèmes viennent des servomoteurs qui ne se maintiennent pas et sont trop lourds. Il sera possible de bloquer l'articulation d'un servomoteur avec une vis sans fin. Pour cela il faut remplacer les servomoteurs par des moteurs classiques. De plus ce type de moteurs étant moins lourd, cela pourra résoudre le problème de poids. On note que remplacer les servomoteurs par des moteurs classiques imposent l'utilisation d'un plus grand nombre de ports numériques. Il faudra peut être devoir utiliser une carte plus grande. De plus une jonction modélisée en 3D entre le moteur jaune et un axe Lego pourra être nécessaire.

### Conclusion

Suite à cette séance, la deuxième version simplifiée du robot est fonctionnelle. Cela garantit le bon aboutissement du projet. De plus certaines options se présentent afin de résoudre les problèmes de la première version plus grosse et délicate.

Pour suivre il faudra tout d'abord compléter et finaliser la version simple en équipant le robot du bloc électronique et en modifiant légèrement son programme afin de lui octroyer certaines capacités. Ensuite il faudra consacrer du temps afin de résoudre les problèmes de la version complexe du robot, quitte à faire fusionner les deux versions afin de rendre le tout fonctionnel.

### Pour la prochaine séance :

- compléter définitivement la version simple du robot
- restructurer les blocs servomoteurs pour essayer d'intégrer des vis sans fin
- retravailler la version complexe du robot (mécanique et électronique)