

CARTE 2021 M5STICKC CHENILLE

1. Cahier des charges de l'objet technique.....	2
2. Étude SYSML.....	2
3. Documents de fabrication.....	5
4. Etude de mise en conformité.....	8
5. Détail du coût.....	13
6. Partie software (IR).....	14

1. . Cahier des charges de l'objet technique

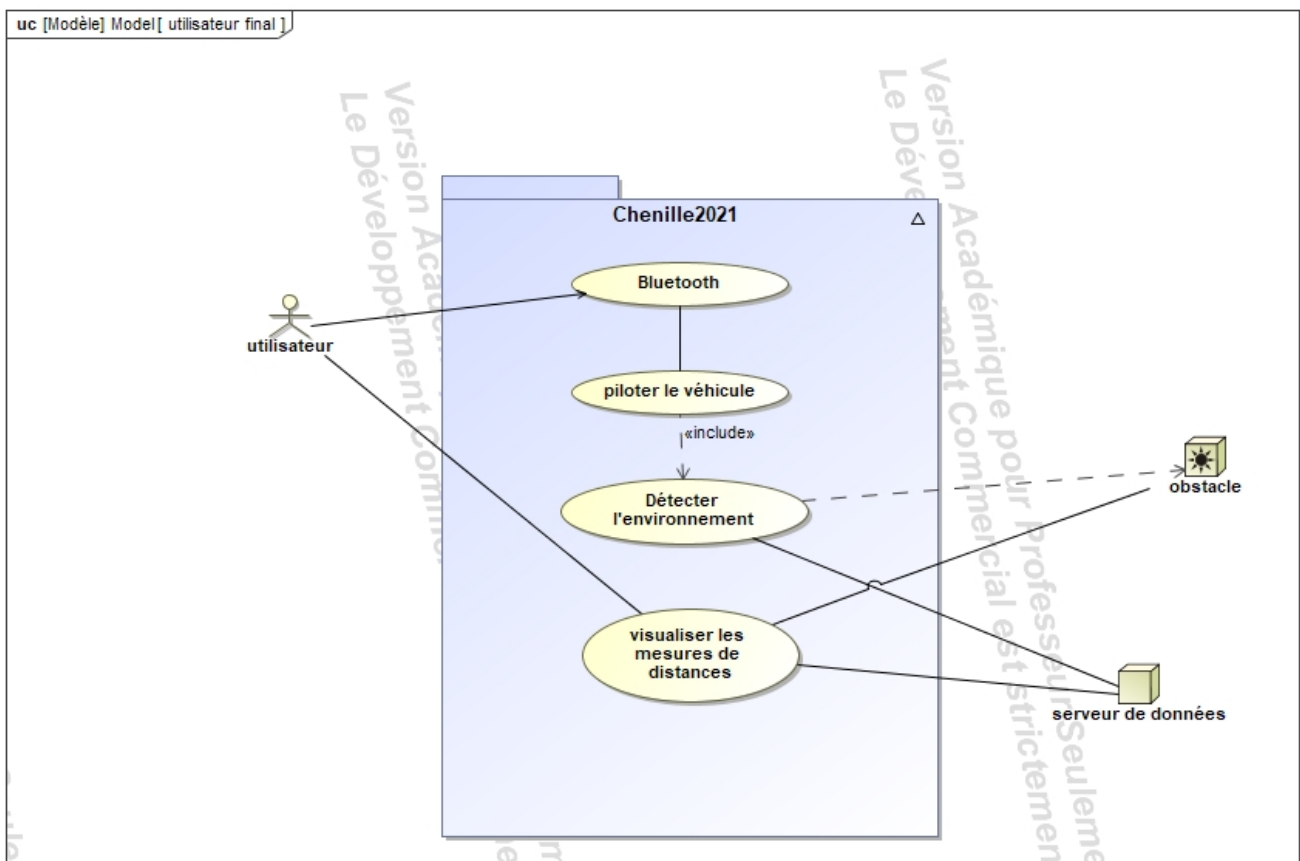
Contrôler la chenille à l'aide d'un module M5StickC (ou C+).

Possibilité de détecter son environnement à l'aide d'un LIDAR (UART). Contrôle des 2 chenilles par PWM vers moteur à courant continu (MCC) à travers un pont en H, positionnement du LIDAR par servomoteur.

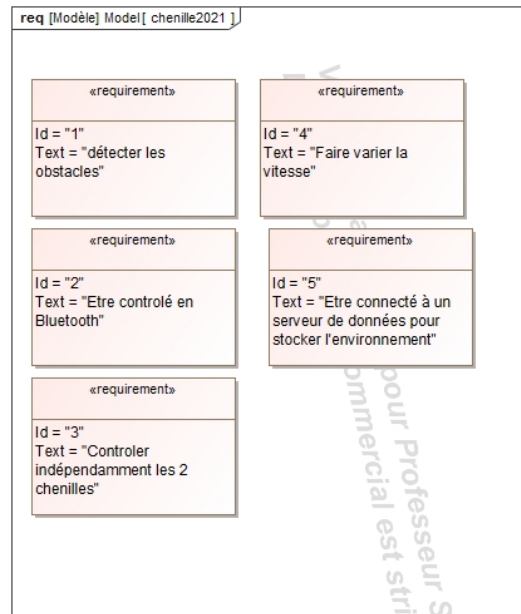
Envoi de mesure vers serveur BDD pour affichage en VR de l'environnement.

2. . Étude SYSML

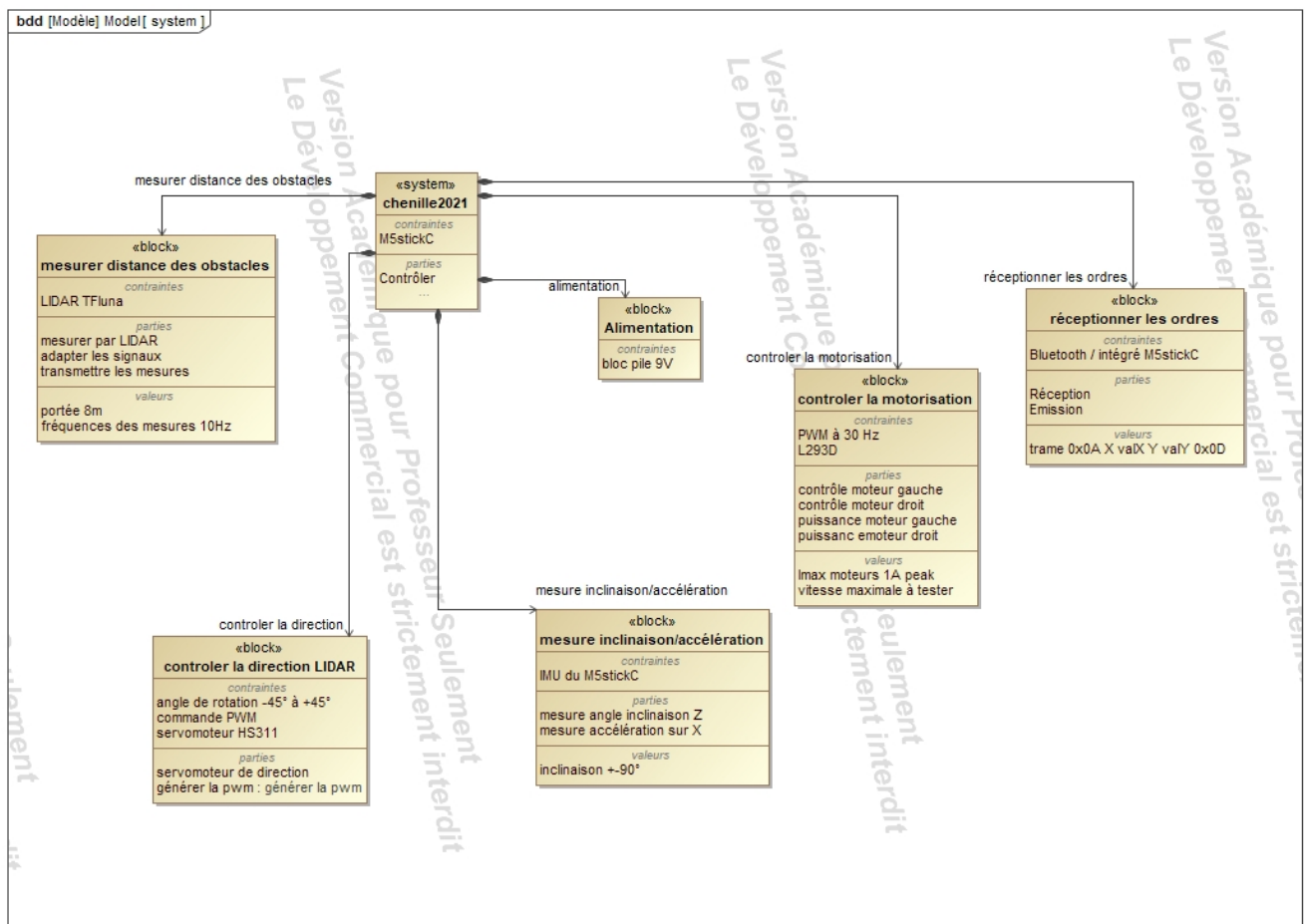
2.1 Diagramme de cas d'utilisation (UC)



2.2 Diagramme d'exigences (req)

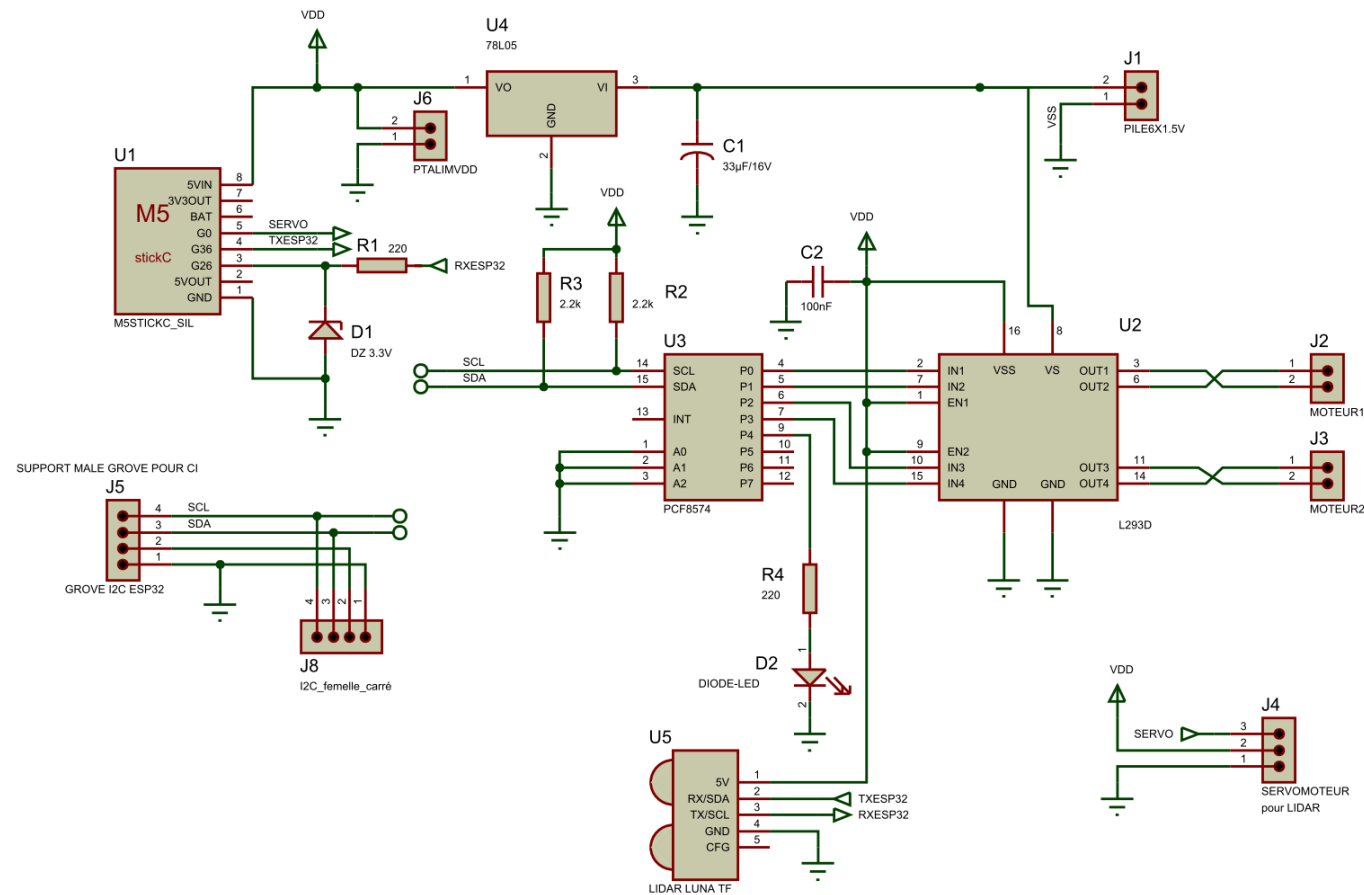


2.3 BDD



2.4 IBD (à faire)

3.1 Schéma structurel.



BD6230F SOP8 fab. rohm
DRV8830 demi pont H - I2C
MAX14870

TITLE: ESP32 chenilles.pdsprj		DATE: 28/09/2021
BY: S.B.	REV: 1.0	PAGE: 1/1

3.4 Nomenclature des composants

Catégorie	Références	Valeur	fabrication
Condensateurs	C1	33µF/16V	POLARISE ELECTROLYTE AXIAL
Condensateurs	C2	100nF	NON POLARISE PLASTIQUE
Résistances	R1	220	TRAVERSANT
Résistances	R2	2.2k	TRAVERSANT
Résistances	R3	2.2k	TRAVERSANT
Résistances	R4	220	CMS
Circuits intégrés	U1	M5STICKC_SIL	SIL MALE CARRE
Circuits intégrés	U2	L293D	SUPPORT DIL
Circuits intégrés	U3	PCF8574	SUPPORT DIL
Circuits intégrés	U4	78L05	TO92 (SUPPORT SIL3)
Circuits intégrés	U5	LIDAR LUNA TF	SIL5 MALE CARRE
Diodes	D1	DZ 3.3V	TRAVERSANT
Diodes	D2	DIODE-LED	5mm
Divers	J1	PILE6X1.5V	BORNIER A VIS
Divers	J2	MOTEUR1	BORNIER A VIS
Divers	J3	MOTEUR2	BORNIER A VIS
Divers	J4	SERVOMOTEUR	SIL3 MALE CARRE
Divers	J5	GROVE I2C ESP32	GROVE MALE

Divers	J6	PTALIMVDD	SIL2 FEMELLE ROND
Divers	J8	I2C_femelle_carré	

4. . Etude de mise en conformité

4.1 Les protocoles de test détaillés

A faire...

4.2 Test du bus I2C vers PCF8574

4.2.1 Organisation

On doit envoyer des données vers le PCF à partir de l'ESP32

Matériel :

ESP32, cable groove, PCF8574, alim pile 9V

mesure : oscilloscope + analyseur logique

Logiciel : zeroplus + arduino IDE (avec librairie PCF8574 library)

Créer un programme simple permettant de mettre toutes les sorties à 0 puis à 1.
Prévoir un délai de 100ms.

4.2.2 Programme :

```
#include <M5StickC.h>
#include "PCF8574.h"

// Set i2c address
PCF8574 pcf8574(0x20);

// the setup routine runs once when M5StickC starts up
void setup() {

    // initialize the M5StickC object
    M5.begin();
    M5.Lcd.setRotation(3);
    // text print
    M5.Lcd.fillScreen(BLACK);
```



```
M5.Lcd.setCursor(0, 10);
M5.Lcd.setTextColor(WHITE);
M5.Lcd.setTextSize(1);
M5.Lcd.printf("Chenille 2021!");

Serial.begin(115200);
//ESP32 example. You can use overloaded function with no parameters in
startI2C() method.
Serial.print("Init pcf8574...");
if (pcf8574.begin()){
    Serial.println("OK");
}else{
    Serial.println("KO");
}
// Set pinMode to OUTPUT
pcf8574.pinMode(P0, OUTPUT);
pcf8574.pinMode(P1, OUTPUT);
pcf8574.pinMode(P2, OUTPUT);
pcf8574.pinMode(P3, OUTPUT);
pcf8574.pinMode(P4, OUTPUT);
pcf8574.pinMode(P5, OUTPUT);
pcf8574.pinMode(P6, OUTPUT);
pcf8574.pinMode(P7, OUTPUT);
}

// the loop routine runs over and over again forever
void loop(){
    M5.Lcd.printf(".-");

    Serial.println("port H");//portA4 PCF8574 pour la LED
    pcf8574.digitalWrite(P0, HIGH);
    pcf8574.digitalWrite(P1, HIGH);
    pcf8574.digitalWrite(P2, HIGH);
    pcf8574.digitalWrite(P3, HIGH);
    pcf8574.digitalWrite(P4, HIGH);
    pcf8574.digitalWrite(P5, HIGH);
    pcf8574.digitalWrite(P6, HIGH);
```

Carte 2021 M5stickC chenille

```
pcf8574.digitalWrite(P7, HIGH);
```

```
delay(400);
```

```
Serial.println("port L");
```

```
pcf8574.digitalWrite(P0, LOW);
```

```
pcf8574.digitalWrite(P1, LOW);
```

```
pcf8574.digitalWrite(P2, LOW);
```

```
pcf8574.digitalWrite(P3, LOW);
```

```
pcf8574.digitalWrite(P4, LOW);
```

```
pcf8574.digitalWrite(P5, LOW);
```

```
pcf8574.digitalWrite(P6, LOW);
```

```
pcf8574.digitalWrite(P7, LOW);
```

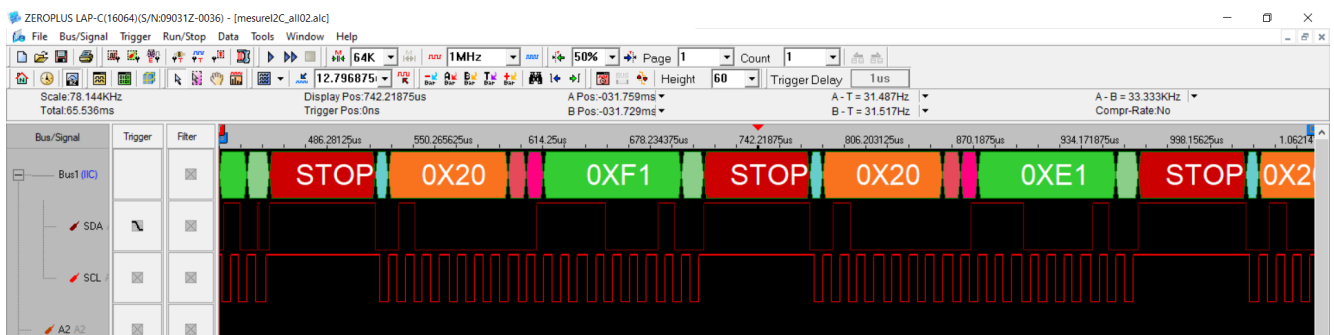
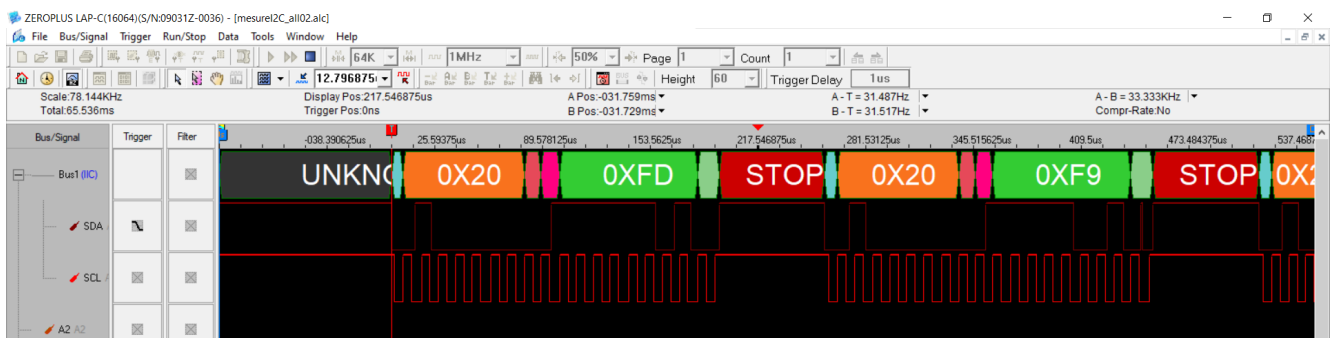
```
delay(400);
```

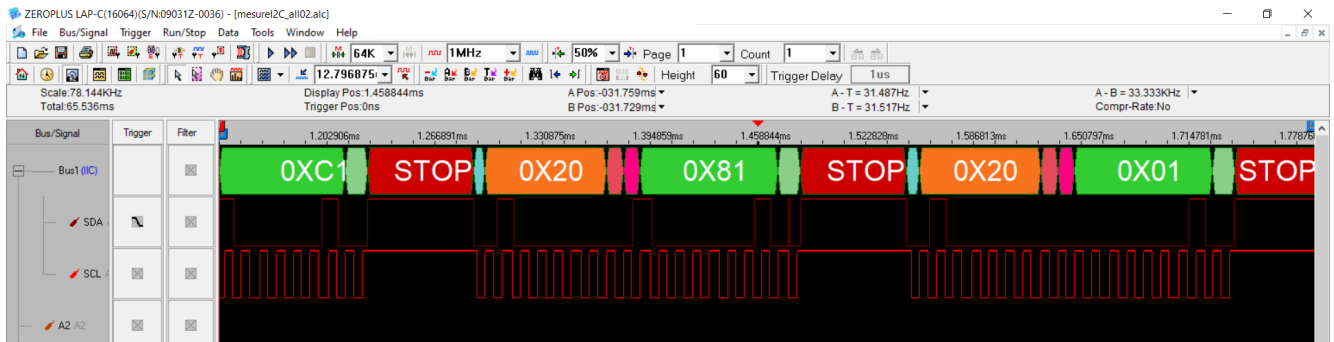
```
//M5.update();
```

```
}
```

4.2.3 Mesures

A l'aide de l'oscilloscope ET de l'analyseur logiques, visualiser le bus I2C.





Normalement la LED branchée sur P4 (broche 9) doit clignoter. (Attention une erreur c'est glissée dans le schéma : à partir de la doc du PCF expliquer pourquoi la LED ne fonctionne pas. (IoL trop faible il faut changer le montage autour de la LED afin d'utiliser I²C ! : inverser la LED couper le lien avec le GND créer un lien avec le 5V.

4.3 Test de la liaison UART

Envoyer un texte vers la liaison série du ESP32 (broche TX (G36) RX (G26)).

A l'aide des appareils de mesure (oscillo et analyseur logique valider le bon fonctionnement de la liaison.

Attention : il faut créer une liaison

....

4.4 Test adaptateur de tension autour de R1,D1

Placer un GBF (réglages : signal carrée 0 +5 V à fréquence 100kHz...) sur U5B3 (entrée de l'adaptateur) et visualiser à l'oscilloscope U1B3 (sortie de l'adaptateur)

4.5 Test LED D2

Sans U3(PCF8574) :

	Valeur attendue	Valeur mesurée
Fil entre J6B2 (5V) et U3B9	LED allumée	
Fil entre J6B1 (0V) et U3B9	LED éteinte	

Conclusion :

La LED fonctionne ou ne fonctionne pas

4.6 Test du contrôle moteur MCC

4.6.1 autour du L293

Sans programme, tester la table de vérité du L293

Relever le fonctionnement des MCC1 et MCC2.

Mise en œuvre :

A vérifier :

Nom broche	Valeur attendue	Valeur mesurée
U2B16	5V (U4 est en place)	
U2B8	9V (PILE)	
U2B4B5	0V	
U2B1	5V	
U2B9	5V	
U2B13B14	0V	

Test du moteur 1 : brancher le moteur sur J2

1A = U2B2	2A = U2B7	Attendue	Mesurée
0(fil sur J4B1)	0(fil sur J4B1)	arrêt	
0(fil sur J4B1)	5V (fil en J6B2)	Tourne droit	
5V(fil en J6B2)	0(fil sur J4B1)	Tourne gauche	
5V(fil en J6B2)	5V(fil sur J4B2)	arrêt	

Test du moteur 2 : brancher le moteur sur J3

1B = U2B10	2B = U2B15	Attendue	Mesurée
0 (fil J6B1)	0 (fil sur J4B1)	arrêt	
0(fil sur J4B1)	5V (fil en J6B2)	Tourne droit	

5V(fil en J6B2)	0(fil sur J4B1)	Tourne gauche	
5V(fil en J6B2)	5V(fil sur J4B2)	arrêt	

Conclusion du test :

Les deux moteurs tournent bien dans les 2 sens

4.6.2 Avec programmation

Proposer un programme simple de commande des moteurs dans les deux sens.

Essayer le programme

Valider le fonctionnement du contrôle moteur

Attention : le moteur est contrôlé par l'ESP32 à travers le PCF8574...

4.7 Test du LIDAR

ajouter la bibliothèque du lidar TFLuna et tester son bon fonctionnement.

4.8 Les chronogrammes et valeurs de mesure obtenus

A faire...

4.9 Les programmes de test

A faire...

5. . Détail du coût

Reprendre le tableau de la nomenclature, ajouter 2 colonnes et mettre les prix chez farnell ET radiospare.

6. . Partie software (IR)

6.1 Cahier des charges

On demande de réaliser les programmes permettant de contrôler la chenille par un M5StickC.

On s'aidera des codes fournis plus haut ainsi que des documents techniques (schéma structurel, d'implantation, doc. Technique des composants)

Objectif final :

Commander la chenille à l'aide d'un téléphone en bluetooth avec envoi de l'inclinaison et l'accélération du véhicule vers une base de données, détection d'obstacle par LIDAR(positionné par servomoteur : optionnel).

Affichage sur une page web des données avec ihm en VR (A-frame) (sens de marche, inclinaison, sens de rotation, position des obstacles (vision 90° à l'avant)

6.2 Organisation :

3 groupes de 4 étudiants : challenge du plus rapide entre les 3 groupes.

Tâches individuelles : 1 étudiant responsable d'1 tâche (travail par groupe autorisé mais l'étudiant est responsable de sa tâche.)

1. contrôle moteur (pont en H)
2. transmission BTH (appinventor)
3. mesure et envoi vers BDD (mysql+php)
4. mesure LIDAR et contrôle de la position LIDAR.
5. Affichage sur page web en VR (A-frame) (tous les étudiants du groupe)

Aides :

Pont en H ([site stssnsb.free.fr](http://site.stssnsb.free.fr))

M5stickC ([site stssnsb.free.fr](http://site.stssnsb.free.fr))

LIDAR ([site stssnsb.free.fr](http://site.stssnsb.free.fr))

A-frame ([site stssnsb.free.fr](http://site.stssnsb.free.fr))

6.3 Planning prévisionnel (20h : 10 séances)

Prise en main du système (1h) : lecture dossier

Prise en main du M5stickC (2h) : matériel + logiciel (PlatformIO + Vscodé OU IDE arduino si problème) + test blink ou autre

Commande de moteur à courant continu par pont en H (1h) : cours

Mise en oeuvre d'un expandeur de port (1h) : étude de la doc + driver Arduino + programme à faire

Communication BTH avec smartphone (apk fournit – introduction appinventor)(2h)

Mise en œuvre d'un LIDAR : étude de doc, programme test mesure distance(2h)

Commande PWM servomoteur LIDAR : étude servo + test (2h)

Mesure accélération et inclinaison, envoi vers base (2h)

Création base de donnée adaptée – code php associé (2h)

Prise en main A-frame et VR : (2h)

Adaptation des codes à notre projet : (3h)

A vous de jouer....

6.3.1 *Prise en main*

Lire le présent dossier et le comprendre.

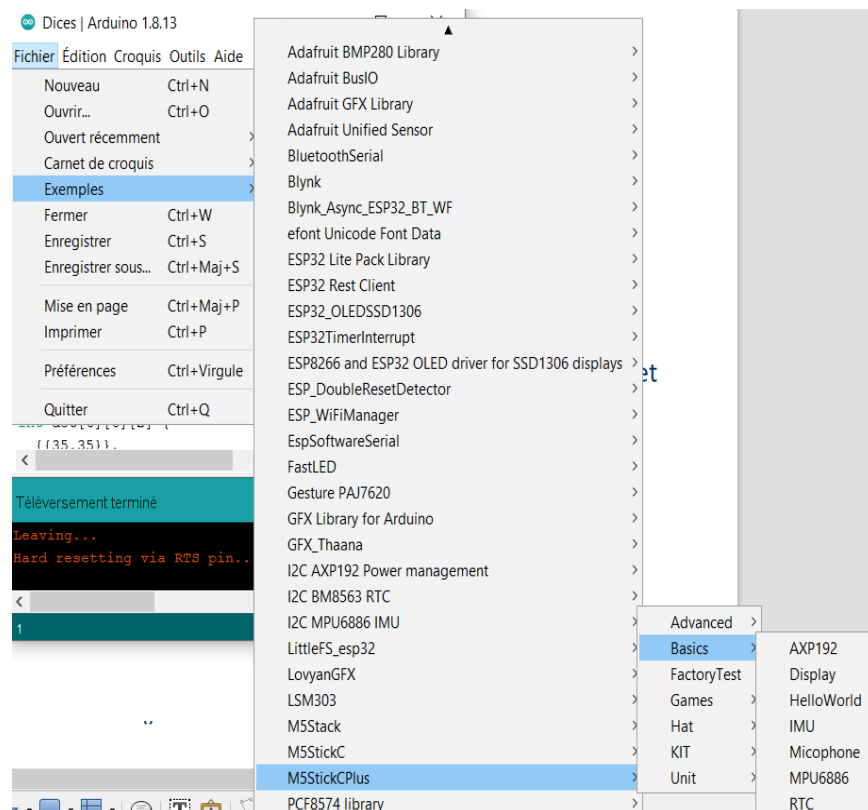
6.3.2 Prise en main M5stickC+

Lire le guide de démarrage (stssnsb.free.fr – iot : M5stickC mise en œuvre bySB.pdf ou https://docs.m5stack.com/en/quick_start/m5stickc_plus/arduino) et le suivre pour mettre en œuvre le 'M5stickC+' avec 'IDE Arduino'

(Rappel : installer en premier : le board M5Stack, puis la librairie 'M5StickCplus')

Remarque : le BluetoothSerial fait une erreur quand on prend le board M5StickCplus mais pas quand on prend un M5StickC en provenance du board 'ESP32 Arduino'!!!!

Brancher le M5StickC+ sur le port USB



Tester les programmes (Exemples + M5StcikCplus + Basic) :

hello world, Display, Dices...

Comprendre les différents exemples afin de connecter les fonctions principales.

Ouvrir avec notepad++ les fichiers M5StickCplus.h et M5StickCplus.cpp, les comprendre. ATTENTION ne pas les modifier.

Amusez vous avec d'autres exemples pour vous familiariser avec ce board.

6.3.3 Commande des moteurs à courant continu par pont en H

1. Lire le cours sur le pont en H : stssnsb.free.fr - cours
2. Faire l'étude du document technique L293 : stssnsb.free.fr – etude de doc. Technique
3. Identifier sur le schéma de la chenille L293 et son cablage avec le M5StickC+.
4. Justifier le rôle du PCF8574 : aidez vous du schéma interactif 'etude de la carte chenille 2021'

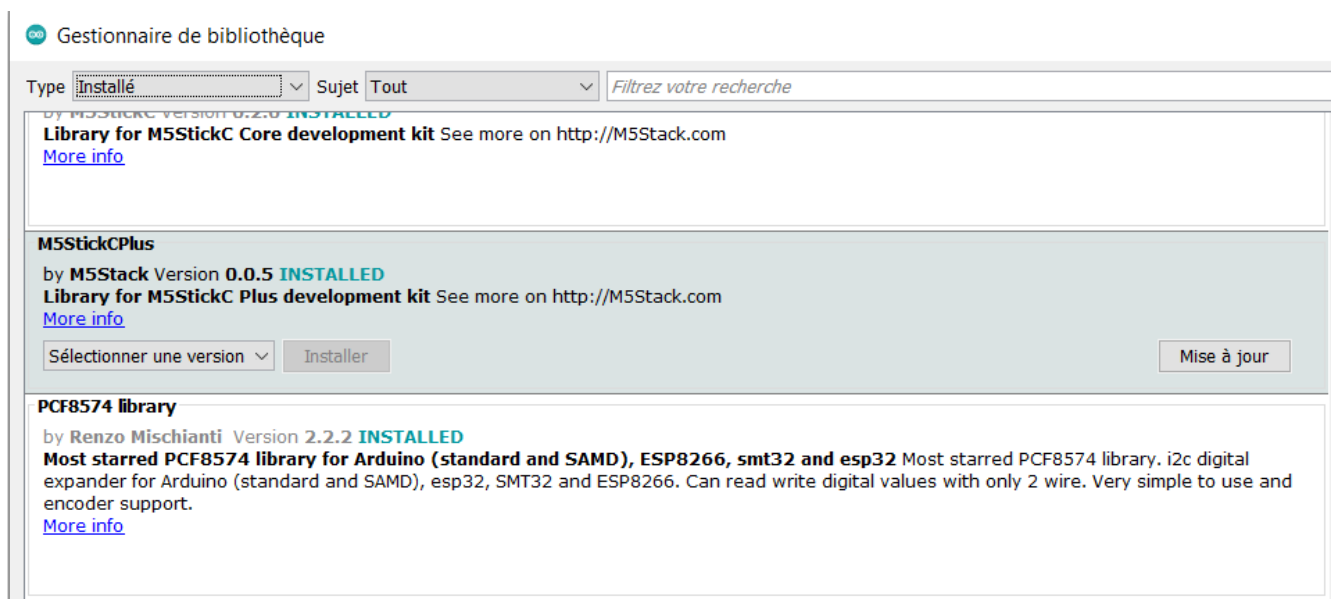
6.3.4 Mise en œuvre d'un expandeur de port PCF8574

a)Etudier la doc du composant.

Comprendre son rôle.

Répondre à l'étude de doc technique du PCF8574

b)Installer le driver :



c)Programmation

Proposer un programme permettant de contrôler les moteurs de la chenille dans les 2 sens. Sans variation de vitesse pour le moment.

6.3.5 Communication BTH avec smartphone

Afin d'envoyer de commander la chenille nous utiliserons une application android déjà développée avec APPINVENTOR.

1. Télécharger le code de l'appli., ouvrez-le sur APPINVENTOR (compte google nécessaire)
2. Etudier le...Quelle est la trame envoyé par l'application ?
3. Proposer un protocole de test de cette application.
4. Installer l'apk sur un smartphone Android.
5. Tester le bon fonctionnement de l'appli avec votre protocole.
6. Installer la librairie : BluetoothSerial
7. Réaliser le code récepteur du M5StickCplus en vous aidant des exemples fournis par l'IDE-Arduino.

6.3.6 Mise en œuvre d'un LIDAR

1. Télécharger la doc du LIDAR
2. Comprendre son rôle et son fonctionnement.
3. Quelle type de communication est utilisée dans notre chenille ?
4. Installer le driver IDE-Arduino du LIDAR TF-Luna.
5. ATTENTION : le signal du LIDAR est 0-5V l'ESP32 ne supporte pas 5V !!! il supporte 3V3 max. NE PAS BRANCHER le LIDAR sur l'ESP32 sans réfléchir !
6. Ouvrir l'exemple de code fournis : stssnsb – iot – LIDAR
7. Etudier le et adaptez le à votre besoin.
8. Créer un code qui fait une mesure et l'affiche sur l'écran du M5StickC. Tester le.
9. Créer un code qui fait une mesure et l'envoi en POST vers un serveur.

6.3.7 Commande du servomoteur

Aide : <https://docs.m5stack.com/en/api/stickc/pwm>

1. Etudier le fonctionnement d'un servomoteur : cours stssnsb.
2. Etudier la documentation du servomoteur fournis.
3. Proposer un programme permettant de faire un déplacement de -45° à +45°.

(Utiliser une librairie ou pas : à votre choix)

6.3.8 *Mesure inclinaison, envoi vers serveur*

Proposer un programme mesurant l'inclinaison en Z de la chenille et l'envoyant en wifi vers le serveur php (option : une base de donnée que vous aurez créée pour mémoriser plusieurs valeurs – sinon utiliser une variable php pour affichage)

Faire un premier programme en utilisant l'exemple 'WifiClient'.