|  |
| --- |
| wETML |
| Pre-TPI Démonstrateur PID |
| Rapport R&D |

|  |
| --- |
| Eden Wegger Funderskov  29/02/2024 |

**Table des matières**

[1 Cahier des charges 3](#_Toc130472158)

[2 Planification 3](#_Toc130472159)

[3 Analyse 3](#_Toc130472160)

[3.1 Analyse de l'existant 3](#_Toc130472161)

[3.2 Prise de décisions 4](#_Toc130472162)

[4 Choix des composants principaux 4](#_Toc130472163)

[5 Mécanique 5](#_Toc130472164)

[6 Schémas 7](#_Toc130472165)

[6.1 Dimensionnement de composants 11](#_Toc130472166)

[7 Programmation 12](#_Toc130472167)

[7.1.1 Paramétrages du μC 12](#_Toc130472168)

[7.1.2 Astuces de codage 12](#_Toc130472169)

[8 Évaluation du projet 13](#_Toc130472170)

[8.1 État d'avancement du projet 13](#_Toc130472171)

[8.2 Travaux restants à effectuer 13](#_Toc130472172)

[9 Conclusion 13](#_Toc130472173)

[Annexe A Planification 14](#_Toc130472174)

[Annexe B Documents de production 14](#_Toc130472175)

[Schémas🡪 14](#_Toc130472176)

[Annexe C Data Sheets 14](#_Toc130472177)

[Data Sheets 14](#_Toc130472178)

**Versions**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Auteur** | **Remarques** |
| 00 | 2024-02-29 | EFV | Version initiale |

# Cahier des charges

Dans le cadre du cours d’électronique analogique, les élèves sont amenés à étudier la régulation en boucle fermée d’un système asservi. Par exemple, une résistance de chauffage maintient une certaine température (consigne) dans une cuve grâce à la mesure de la température (mesure). La dissipation thermique refroidit la cuve (erreur) et le système régule le courant dans la résistance afin de maintenir la température.

Dans le cadre de ce projet (N°1391), l’idée est de maintenir une balle à une hauteur définie par la consigne dans un tube placé verticalement. La position de la balle sera réglée par un clapet permettant d’amener un flux d’air depuis un ventilateur.

Ce rapport contiendra l’ensemble du travail du pre-TPI afin de documenter les recherches et réalisations,

# Planification

|  |  |  |
| --- | --- | --- |
| ***Liste des tâches*** | ***Durée Planifiée*** | ***Durée***  ***Réalisée*** |
| Planifier | 2h |  |
| Analyse/recherche problèmes/solutions | 10h |  |
| Vérification schéma électrique | 10h |  |
| recréation du schéma électrique | 6h |  |
| Documentation | 5h |  |
| **Total** | 32h |  |
| **Erreur Relative** | % | |

|  |  |  |
| --- | --- | --- |
| ***Liste des tâches*** | ***Durée Planifiée*** | ***Durée***  ***Réalisée*** |
| Planifier | 2h | 1h30 |
| Décider | 11h | 13h50 |
| Réalisation 3D | 16h | 5h30 |
| Schéma bloc | 3h | 🡪 |
| Schéma électrique | 5h | 12h50 |
| Total 🡪 | 37h | 34h |
| Erreur Relative🡪 | -8% | |

~~Je n’ai pas fait de planification sous forme de diagramme de Gantt, car je n’y ai pas pensé, et le faire vers la fin du pré-TPI me semblait une perte de temps, connaissant déjà la répartition de mon temps, je pourrais pas estimer/deviner cela~~.

Voir planification

# Analyse

## Analyse de l'existant

Documents :

* Cahier des charges 🡪 TPI 2023 - Blatti - CdC pré-TPI.pdf

Logiciel :

* Altium designer
* Gitkraken

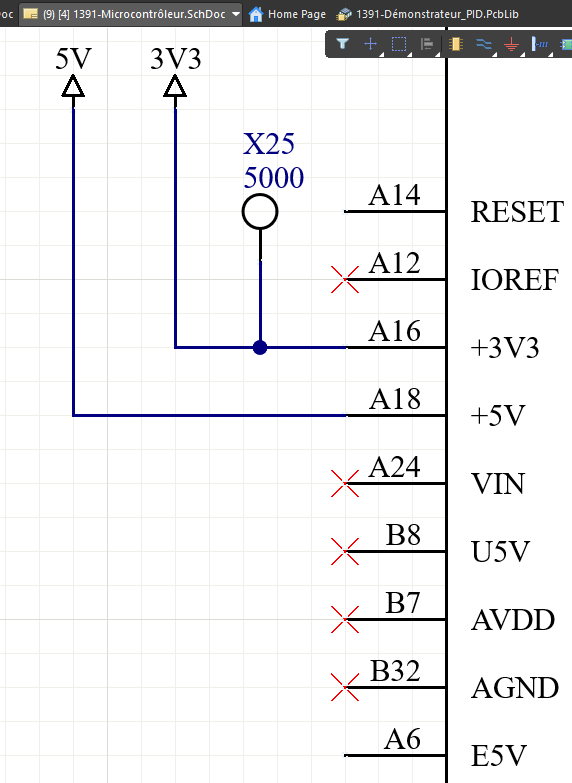
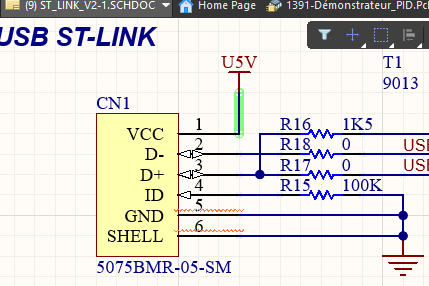
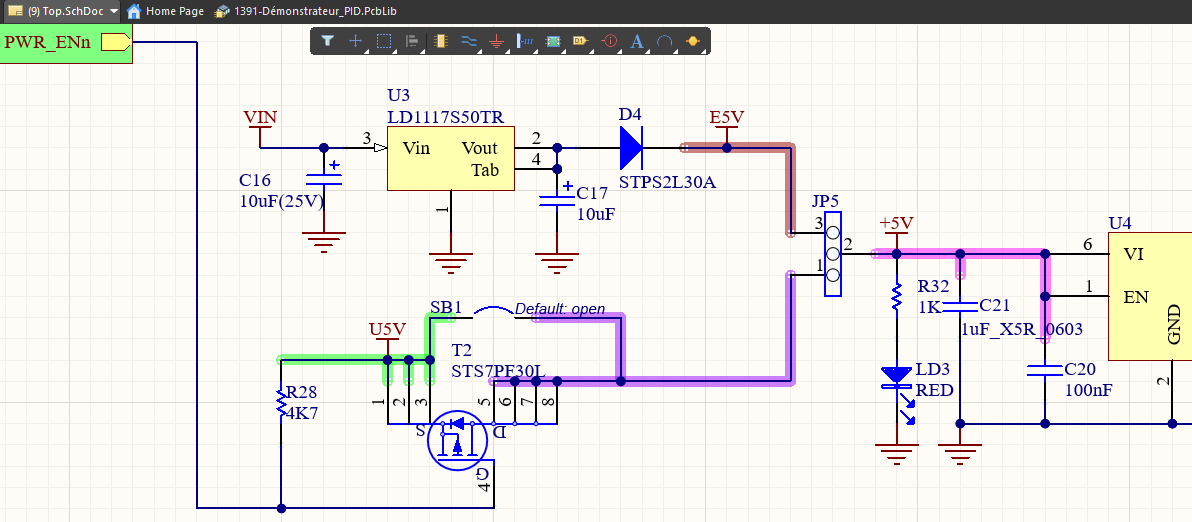
## Prise de décisions

Concernant ce TPI j’ai abordé la chose de la manière suivante :

J’ai décidé d’investiguer la source de ce qui a causé une panne de la carte mère de mon ordinateur portable.

J’ai réussi à trouver la source du problème.

Le routage de l’alimentation entre le nucléo et le reste du PCB était incorrecte. Le 5V du régulateur de tension du PCB était connecté à la sortie +5V du Nucléo.

Pour résoudre cela

De ce que j’ai observé dans les datasheets, il n’existe pas une solution qui empêchera un risque à l’ordinateur connecté au Nucléo. Autre que des précautions d’utilisation, notamment d’alimenter le PCBA en premier puis connecter l’ordinateur au Nucléo. Cependant cela nécessitera de déconnecter l’ordi en cas de coupure de l’alimentation du PCBA.

Pour les modifications du schéma,

**Alimentation +3V3 depuis le nucléo**

J’ai choisi de continuer à utiliser le régulateur +3V3 sur le nucléo pour alimenter les composants du PCB nécessitant du 3v3. Il faut par contre vérifier que consommation maximale ne dépasse pas 500mA[[1]](#footnote-2).

~~Dans le datasheet du Nucléo STM32F070RB, il est indiqué que~~

Ce projet ressemble de près à ce que pourrait réaliser un travail d’ingénieur. En effet partir de « rien » et de concevoir le système de A à Z. En se référant à la planification la première partie consiste en décider des composants principaux du système. Une fois les composants principaux choisis je ferai la partie mécanique ou du moins la réalisation 3D Du montage final. Et en dernier une fois que toues les composants principaux et la parie mécanique finie, je réaliserai les schéma de la carte de commande.

# Choix des composants principaux

Pour le PCB

Seul les composants rajoutés comme modification par la personne précédente seront rajoutés dans les documents. A part cela aucun changement n’a été planifié pour l’instant,

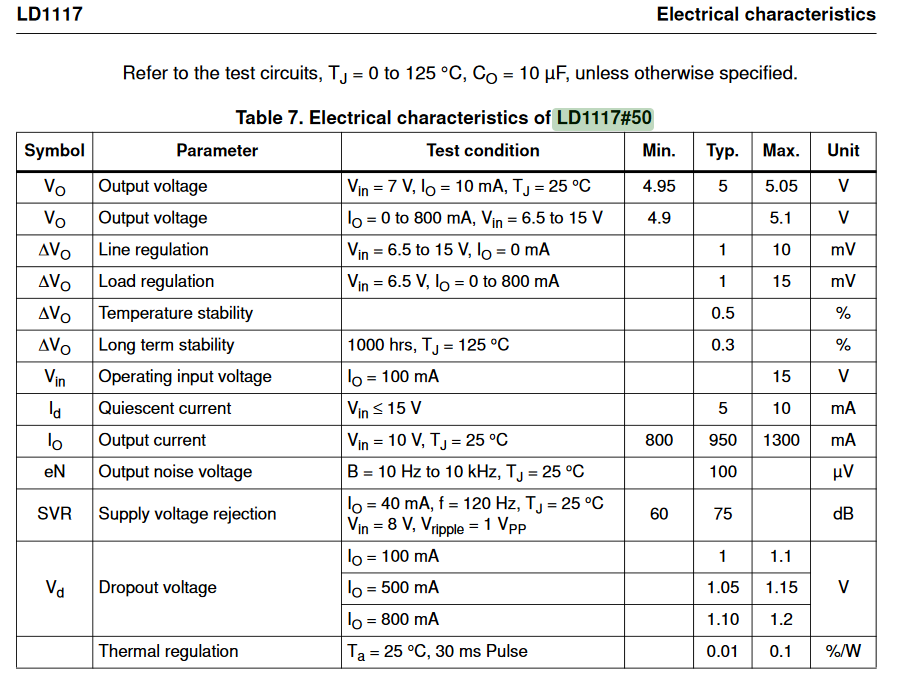


Figure 1 Datasheet régulateur 5V du Nucléo

Le premier composant est le ventilateur. Il responsable de fournir une ventilation assez puissante pour soulever un objet assez léger certes mais avec un poids non négligeable. Pour le choix de ce dernier, il s’agissait d’un « saut dans le vide ». En effet n’étant pas un connaisseur j’ai pris un premier ventilateur qui me semblait faire l’affaire, le « Noctua NF-R8 redux 1200 3-PIN » comme précise dans son nom il atteint un maximum de 1200 tours minutes pour un ventilateur de 80mm de diamètre.

Malheureusement son flux d’air d’environ 35m³/h ne suffisait pas du tout. Le but de ce ventilateur est de la piloter en PWM afin de régler la hauteur de l’objet dans le tube, si le ventilateur ne peut même pas soulever l’objet à pleine puissance il est clairement écarté des choix possibles. Le ventilateur qui sera utilisé est un autre ventilateur de chez Noctua ayant un flux d’air et un diamètre plus grand.

Le deuxième composant essentiel est le capteur de distance. Le composant sélectionné est le suivant : Le capteur IR distance Click de chez Mikroe. Ce capteur nous permet de faire une mesure précise entre 0 et 80 cm. Grace a ce capteur je peux déterminer la hauteur de l’objet dans le tube. Bien que le capteur présent sur le petit PCB soit obsolète étant donne sa taille et sa précision je ne me voyais pas changer de capteur.

Le troisième composant nécessaire est le servomoteur. Ce composant permettra d’ouvrir une portion du tube pour contrôler l’échappement du flux d’air. Le servo moteur utilisé pour mes tests et pour la version finale du projet sera surement un SG92R, un Servo moteur basique permettant de contrôler précisément l’ouverture du tube.

Tout ces composants ont été testés avec de petits bouts de code afin de me familiariser a ces derniers. Cette partie programmation est décrite plus loin dans le rapport.

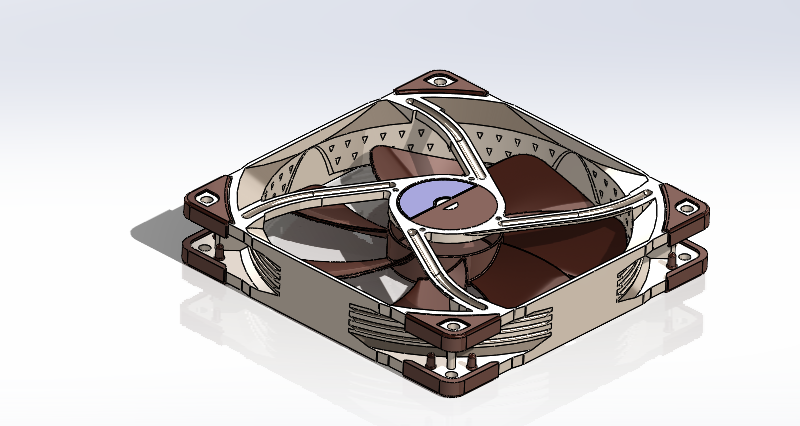
# Mécanique

Pour mon TPI, il n’y aura pas d’aspect mécanique autre que les jambes du PCB

Une fois les composants principaux testé et choisis il me fallait travailler un peu sur la partie mécanique et la conception du projet. J’ai donc décider de réaliser en 3D le projet afin de me donner une idée de comment allait être fixer tous les composants

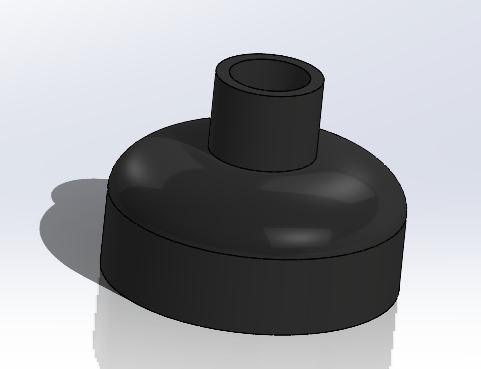
Principaux entre eux. Le design 3D effectuer sur SolidWorks est le suivant :

Ventilateur (design non-effectué par mes soins)



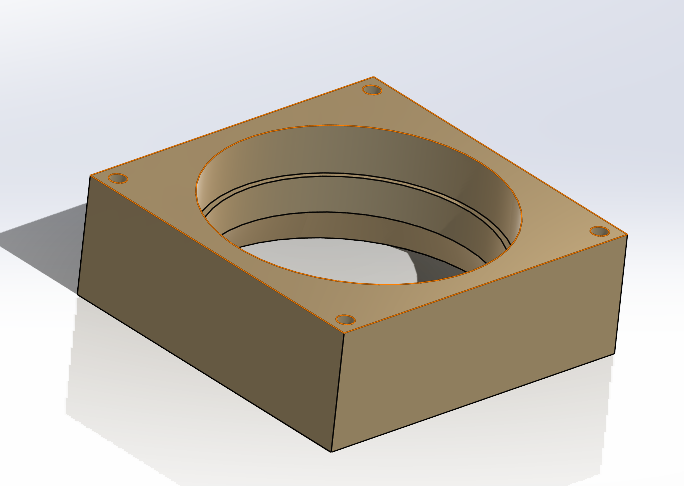
Ce ventilateur le Noctua NF-A14 PPC 3000 est un ventilateur de 140 mm de diamètre nous permettant d’avoir un flux d’air assez conséquent pour soulever l’objet à travers le tube.

Guide d’air :



Piece permettant de guider l’air et de le concentrer a travers le tube. Il se fixe a la pièce de raccord.

Piece de raccord :



Piece convertissant le diamètre du ventilateur au diamètre du guide d’air.

De 140mm a 120mm.

Piece de support pour

le capteur de distance :

Une image contenant texte

Description générée automatiquement

Cette pièce permet de tenir le capteur de distance en hauteur afin de viser l’intérieur du tube. La bague viendra se serrer autour du tube et l’oblong dans la colonne permet de régler la hauteur du capteur.

Une fois la modélisation 3D elle ressemble à la photo présente en couverture de ce rapport.

# Schémas

L’aspect des schémas restera plus ou moins le même qu’à la dernière version de ce projet. Donc je vais utiliser ces schémas comme base et faire les modifications nécessaires.

La première modification était de corriger les footprints des potentiomètres. Il m’a fallu refaire car j’ai fait l’erreur de ne que changer les pins des composants existants, car cela n’avait pas changé le footprints des composants dans l’éditeur de PCB.

La première feuille ci-dessous est celle du schéma bloc. Elle comporte la listes des feuilles du schéma et des interactions entres elles. Dans ce schéma bloc se trouve également les symboles de normes respectée pour la production de la carte et de son entretiens, RoHS, ESD, WEEE. Il y a 4 points test de GND pour pouvoir faire plusieurs mesures en simultané ainsi que 4 trous non-métallisés pour fixer la carte sur son support.

Une image contenant diagramme, schématique

Description générée automatiquement

Schéma Bloc

Ensuite le bloc le plus simple du projet et celui de l’alimentation, en effet la carte est alimentée en 12V par un transformateur 230V-12v de chez tracopower pour ensuite être transformée en 5V par un régulateur. Le 12V sera utilisé pour commander le ventilateur, le 5V pour alimenter une bonne partie de la carte ainsi que le NUCLEO F070RB qui lui fournira notre tension 3V3. Les condensateurs sont de valeur normalisée pour ce genre de régulateur.

Une image contenant diagramme, schématique

Description générée automatiquement

Alimentation

Le schéma de régulation est composé de 4 potentiomètres chacun relié sur un ADC du microcontrôleur afin de mesurer le réglage de la consigne ou des réglages PID. Les condensateurs sont des condensateurs de déparasitage afin de nettoyer le signal arrivant sur l’ADC. Il y a également un bouton au cas ou une erreur se produis qui pourrait générer une interruption.

Une image contenant diagramme, schématique

Description générée automatiquement

Régulation PID

La partie microcontrôleur est le cerveau de la carte. Il contrôle les ADC pour la régulation ainsi que le capteur de distance, les sorties pour l’affichage ainsi que le PWM du servomoteur et du ventilateur. Pour la commande du ventilateur j’ai laissé la possibilité de choisir entre deux types de ventilateur, 3PIN ou 4PIN de chez Noctua. Pour cela il me fallait mettre un jumper pour la sélection ainsi qu’un buffer en open collector afin de pouvoir piloter le ventilateur de la bonne manière comme le recommande Noctua. Le connecteur pour aller jusqu’au différents périphériques est un connecteur

SUB-D25.

Une image contenant diagramme, schématique

Description générée automatiquement

Microcontrôleur

Pour l’affichage LEDS, le principe est très simple, il y a 8 sorties du microcontrôleur qui pilote 32 leds simulant un bargraphe pour indiquer le niveau de la consigne par rapport a celui de la mesure.

Une image contenant diagramme, schématique

Description générée automatiquement

Affichage LEDs

## Dimensionnement de composants

Le dimensionnement des composants ayant déjà été faite, je ne vais que vérifier qu’elles sont correctes.

Le dimensionnement de composant a été plutôt simple en effet il ma suffit de dimensionner seulement les résistances de limitations de courant pour les 32 leds de l’affichage.

Le calcul est le suivant :

# Programmation

La partie de programmation a déjà été faite en partie. Il faudra juste que je fasse quelques modifications pour l’orientation des LEDs. Ceci sera fait après l’assemblage du PCB.

Il faudra cependant que je vérifie les ports qui sont utilisé ne change pas sans adaptation au niveau du logiciel.

Dans ce rapport la partie programmation ne concerne pas la carte de commande mais les tests réalisés sur breadboard avec une carte NUCLÉO F070RB (La même qu’utilisé sur la carte commande).

### Paramétrages du μC

Pour les paramètres du microcontrôleur j’ai utilisé une entrée ADC pour avoir les données de m on capteur de distance et deux sortie PWM, une pour contrôler le ventilateur et une autre pour contrôler le servo moteur.

### Astuces de codage

J’ai donc testé mes composants de la manière suivante :

Cette capture d’écran ci-dessous permet de voir comment est utilisé mon capteur de distance.

Une image contenant texte

Description générée automatiquement

Cette capture d’écran ci-dessous permet de voir comment est utilisé le servo moteur.

Une image contenant texte

Description générée automatiquement

Pour la gestion du servo moteur la durée du rapport cyclique varie entre 0,5 et 2,5 ms il suffit de paramétrer la fréquence du PWM de la bonne manière et varier le rapport cyclique pour faire bouger le servo moteur.

Pour la gestion du ventilateur le PWM et utilise pour faire varier le rapport cyclique d’un signal sur la grille d’un mosfet connecté au ventilateur pour faire varier sa vitesse de rotation.

# Évaluation du projet

## État d'avancement du projet

Les schémas sont terminés les composants ont été sélectionnés. Le cahier des charges du pré TPI est complété :

1) Recherche et analyse des problèmes existant

2) Trouver des solutions aux problèmes

3) Recréer/corriger le schéma électrique

4) finir le R&D pré-TPI

5) ???

6) Parler aux experts

1) Imaginer à l’aide d’un schéma bloc les types de signaux et l’interface utilisateur ;

2) Déterminer la connectique (et les signaux) entre la carte commande et le démonstrateur (balle dans le tuyau) ;

3) Choisir un microcontrôleur répondant aux différents besoins ;

4) Choisir un ventilateur ainsi que son système de commande (PWM de préférence) ;

5) Choisir un servo-moteur pour le clapet régulant le flux d’air ;

6) Dessiner le schéma sur Altium.

## Travaux restants à effectuer

Le travail restant a effectué est plutôt logique il s’agit du TPI en lui-même avec toutes les tâches suivantes :

1) Réalisation du routage et de la liste de pièce (BOM) ;

2) Commander le matériel avant les vacances de Pâques ;

3) Monter/braser le circuit imprimé ;

4) Mettre en service et tests à l’aide d’un programme de test (n’incluant pas la régulation !).

5) Réaliser une ébauche de mode d’emploi.

# Conclusion

Pour conclure même si le projet est loin d’être fini le cahier des charges pour le pre TPI est complété.

Jai pu me familiariser avec les composants que j’allais utiliser et préparer tout ce qu’il fallait pour le début du TPI, c’est-à-dire les footprint des composants et une bonne partie du design de la mécanique. Au niveau du temps, cela pris plus ou moins le temps estimé en tout, même si le temps estimé pour chaque tâche ne correspondait pas vraiment. Je vais donc faire une planification plus précise pour le TPI afin de mieux gérer mon temps. Ce début de projet c’est très bien passé.

1. Planification

🡪C:\Users\thoblatti\Education Vaud\ETML\_XLO-19-23 - Blatti\Projets\1391-Demonstrateur\_de\_regulation\_PID\_TPI\3-Documents

Journal de travail 🡪 Journal-Blatti-20230323\_1346.pdf

Planification 🡪 Planification-Blatti-20230323\_1345.pdf

1. Documents de production

Schémas C:\Users\thoblatti\Education Vaud\ETML\_XLO-19-23 - Blatti\Projets\1391-Demonstrateur\_de\_regulation\_PID\_TPI\5-Hardware\1391-Demonstrateur\_PID\Assembly\1391-Démonstrateur\_PID\_SCH.PDF

1. Data Sheets

Data Sheets C:\Users\thoblatti\Education Vaud\ETML\_XLO-19-23 - Blatti\Projets\1391-Demonstrateur\_de\_regulation\_PID\_TPI\7-DataSheets

1. Datasheet Nucléo UM1724 - **6.3.4** [↑](#footnote-ref-2)