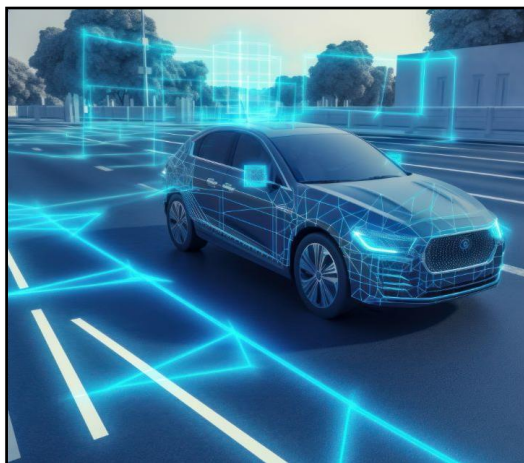


Note Technique – IMTHADULLA Adam

Rapport de validation

Remplacement de logique de protection discrète par des «
Programmable Mixed-Signal Products »



Étudiants :

Firas AMMAR

Adam-Musharraf IMTHADULLA

Clients :

M. German HULGICH

M. Victorien AUBRET

Tuteur industriel :

M. Pascal FICKINGER

Tuteur de revue :

M. Julian LAURENCE

Tuteurs académiques :

M. Alexis LANDRAULT

M. Michel JAMES

Enseignante d'EEO :

Mme. Véronique

QUANQUIN RUET

TABLE DES MATIERES

1	Introduction	3
2	Matériel et logiciel	4
3	Préparation des testes	6
3.1	Table des connexions	8
3.2	Équations régissant le circuit	8
3.3	Branchement	9
3.4	Importation du projet e2Studio	9
3.5	Connexion de la carte de communication Série PMOD USB-UART	11
4	Automatisation des tests	12
4.1	Indicateurs LED et validation des tests	13
4.2	Analyse des résultats	13
4.3	Conversion et visualisation des résultats	13
5	Interprétation des résultats	14
5.1	Scénario de Test 1 : Validation de la Sortie nOC	14
5.2	Scénario de Test 2 : Validation de la Sortie nOV	16
5.3	Scénario de Test 3 : Validation du Bloc de Latch	17
5.4	Scénario de Test 4 : Validation de la Sortie DRV_ASC_EN_ISO	18
5.5	Scénario de Test 5 : Validation de la Sortie DRV_ASC_ISO_HS	19
5.6	Scénario de Test 6 : Validation de la Sortie DRV_ASC_ISO_LS	20
5.7	Scénario de Test 7 : Vérification Temporelle des Sorties de Protection	21
6	Tests Non Réalisés	23
7	Les états de l'onduleur	24
8	Conclusion	25

TABLES DES TABLEAUX

Tableau 1 : Matériel requis pour la réalisation des tests	4
Tableau 2 : Logiciel requis pour la réalisation des tests	5
Tableau 3 : Branchement USB-UART	11
Tableau 4 : Scénario 1 - table des résultats	15
Tableau 5 : Scénario 1 - nOV_latched.....	16
Tableau 6 : Scénario 2 - table des résultats	17
Tableau 7 : ADC.....	17
Tableau 8 : Scénario 3 - table des résultats	18
Tableau 9 : Scénario 4 - table des résultats	19
Tableau 10 : Scénario 5 - table des résultats	20
Tableau 11 : Scénario 7 - table des résultats	20
Tableau 12 : Contrainte temporelle.....	21
Tableau 13 : Etats de l'onduleur	24

TABLES DES FIGURES

Figure 1 : Placement du SLG sur l'adaptateur	6
Figure 2 : Branchement Adaptateur - Carte de développement.....	6
Figure 3 : Icon Debug.....	6
Figure 4 : GreenPAK Advanced Development Platform	7
Figure 5 : Debugging Controls	7
Figure 6 : Existing Projects into Workspace.....	9
Figure 7 : Select archive file.....	10
Figure 8 : Projet - Test_SLG.....	10
Figure 9 : look_up_table.h.....	10
Figure 10 : Build All.....	10
Figure 11 : Carte USB-UART.....	11
Figure 12 : Branchement Renesas S7G2 "debug"	12
Figure 13 : Icon debug e2studio	12
Figure 14 : e2studio - lancement du debug	12
Figure 15 : Affichage des teste réaliser sur le terminal	12
Figure 16 : Synoptique circuit externe	14
Figure 17 : Branchement circuit externe	15

1 Introduction

L'objectif principal du projet est de remplacer la logique de protection discrète existante par une solution programmable permettant d'assurer des performances améliorées, une réduction des coûts et une meilleure robustesse. Ce rapport présente les étapes de validation réalisées, les outils utilisés, le fonctionnement du programme développé, ainsi que les résultats obtenus.

2 Matériel et logiciel

La réutilisation du projet pour tester et valider la puce SLG nécessite d'abord la disponibilité du matériel adéquat.

Matériel	Fonction	Lien
Puce SLG46625	Composant programmable testé pour implémenter les fonctions de sécurité de l'onduleur.	Lien
Adaptateur SLG46625-SKT	Facilite la programmation de la puce et son prototypage.	Lien
Carte de développement SLG4DVKADV	Fournit des fonctions complètes de programmation, d'émulation et de test pour les puces GreenPAK.	Lien
Carte de développement Renesas S7G2	Pilote les tests automatisés en intégrant les périphériques nécessaires comme les DAC, GPIO et UART. (Seulement pour la partie validation)	Lien
Pmod USB-UART	Permet de faire la communication série entre la carte S7G2 et l'ordinateur	Lien
Oscilloscope	Utilisé pour capturer les signaux électriques générés par le système et valider les temps de réponse. (Seulement pour la partie validation)	N/A

Tableau 1 : Matériel requis pour la réalisation des tests

La carte Renesas S7G2 permet uniquement d'automatiser les tests pour la validation de la puce SLG. D'autres méthodes peuvent être employées pour vérifier le circuit. Cependant, ce rapport détaille la mise en place des tests à l'aide de la carte Renesas S7G2.

L'utilisation de logiciels de développement adaptés est essentielle pour mener à bien ce projet. Les outils requis sont listés ci-dessous :

Logiciel	Fonction	Lien
Go Configure Software Hub (Version : 6.46)	Logiciel de conception graphique pour la configuration des puces GreenPAK, permettant de créer et de simuler des fonctions logiques complexes sans codage manuel.	<u>Lien de téléchargement</u>
Renesas e ² studio (Version : 24.1.1)	Environnement de développement intégré pour la programmation des microcontrôleurs Renesas. Il permet d'automatiser les tests de façon rapide.	<u>Lien de téléchargement</u>
Python (Version : 3.12.7)	Il permet de lire les données sur le port série du PC et de générer un fichier CSV qui stocke les informations des tests.	<u>Lien de téléchargement</u>

Tableau 2 : Logiciel requis pour la réalisation des tests

3 Préparation des tests

La conception du circuit utilisant la puce SLG46625 est réalisée avec le logiciel Go Configure. Pour mettre en place la validation, il est nécessaire d'avoir préalablement flashé la puce SLG46625 avec le circuit conçu dans ce projet.

Étapes de programmation :

1. Placer une puce SLG46625 dans l'adaptateur SLG46625-SKT.

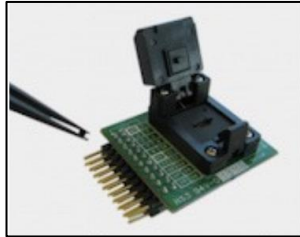


Figure 1 : Placement du SLG sur l'adaptateur

2. Brancher l'adaptateur à la carte de développement.



Figure 2 : Branchement Adaptateur - Carte de développement

3. Connecter la carte de développement SLG4DVKADV à l'ordinateur via un câble USB Mini-B.
4. Ouvrir le fichier de conception : `Dossier_technique\GreenPAK\Architecture_interne.gp4`
5. Cliquer sur « Debug ».

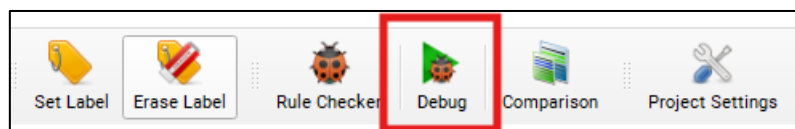


Figure 3 : Icon Debug

6. Sélectionner « *GreenPAK Advanced Development Platform* ».



Figure 4 : *GreenPAK Advanced Development Platform*

7. Vérifier que la carte de développement est reconnue et que l'option « *Program* » n'est pas grisée dans l'onglet « *Debugging Controls* ».

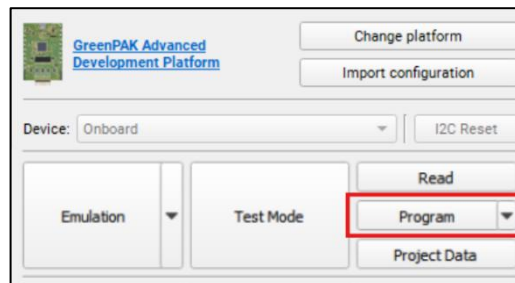


Figure 5 : *Debugging Controls*

8. Cliquer sur « *Program* » pour flasher la puce SLG46625 et attendre la fin du chargement.

Une fois la puce programmée, il est nécessaire d'ajouter la partie externe du circuit complet.

3.1 Table des connexions

PIN S7G2	PIN SLG	Signaux du circuit	Entrée/sortie	Commentaire
VDD	1	VDD	Entrée	Alimentation
010	2	SUP_DCDC_PGOOD	Entrée	
311	3	PS_nDIS_PWM_BIS	Entrée	
613	4	PS_nDIS_PWM	Entrée	
314	5	DRV_ASC_EN_ISO	Sortie	
014	6	DOUT	Entrée	Signal analogique créé par le contrôleur.
614	7	PMIC_nSS1	Entrée	
615	8	DRV_HS_nFLT1	Entrée	Indique un problème sur les transistors à l'état haut.
111	9	DRV_LS_nFLT1	Entrée	Indique un problème sur les transistors à l'état bas.
GND	10	GND	Entrée	Non utilisé dans ce projet.
GND	11	GND	Entrée	Masse
313	12	nOC_LATCHED	Sortie	
112	13	PS_asc_SEL	Entrée	
315	14	nOV_LATCHED	Sortie	
312	15	DRV_ASC_ISO_LS	Sortie	
113	16	nRESET_LATCHED	Entrée	
513	17	nOC	Entrée	Permet de simuler le bloc de détection de surintensité.
512	18	PS_nASC	Entrée	
511	19	PS_nASC_bis	Entrée	
415	20	DRV_ASC_ISO_HS	Sortie	

3.2 Équations régissant le circuit

Le comportement du circuit est défini par les cinq équations suivantes :

1. **DRV_ASC_EN_ISO** = $\neg (nOC_LATCHED \& nOV_LATCHED \& PMIC_nSS1 \& DRV_HS_nFLT1 \& DRV_LS_nFLT1 \& SUP_DCDC_PGOOD \& PS_nDIS_PWM \& PS_nDIS_PWM_BIS)$
2. **DRV_ASC_ISO_HS** = $ASC \& PS_ASC_SEL \& DRV_xx_nFLT1 \& SUP_DCDC_PGOOD$
3. **DRV_ASC_ISO_LS** = $ASC \& \neg (PS_ASC_SEL \& DRV_xx_nFLT1 \& SUP_DCDC_PGOOD)$

4. **nOC_LATCHED** =

- Si (!nRESET_LATCHED & nOC) → 1
- Sinon si (nRESET_LATCHED & nOC) → État de mémoire
- Sinon → 0

5. **nOV_LATCHED** =

- Si (!nRESET_LATCHED & (DOUT < 1V)) → 1
- Sinon si (nRESET_LATCHED & (DOUT < 1V)) → État de mémoire
- Sinon → 0

La création de la table de vérité a été générée avec Python, car les équations 4 et 5 impliquent des états de mémorisation. Il était donc nécessaire d'utiliser des variables intermédiaires. Le script Python utilisé est disponible dans le dossier *Dossier_technique\Python\Create_look_up_table.py*.

3.3 Branchement

Les broches de l'adaptateur SLG doivent être connectées aux broches du contrôleur Renesas S7G2, en respectant les correspondances du tableau ci-dessus.

3.4 Importation du projet e2Studio

Avant de lancer l'automatisation des tests, il faut d'abord importer le projet *e2Studio*. Voici les étapes à suivre :

1. Ouvrir *e2Studio* et cliquer sur l'onglet « *File* », puis sélectionner « *Import* »
2. Choisir « *Existing Projects into Workspace* ».

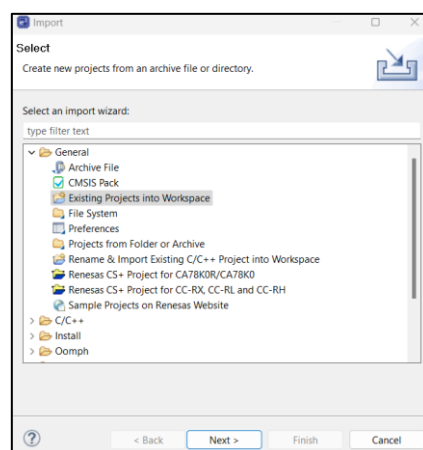


Figure 6 : Existing Projects into Workspace

3. Cocher « *Select archive file* », puis cliquer sur « *Browse...* ».

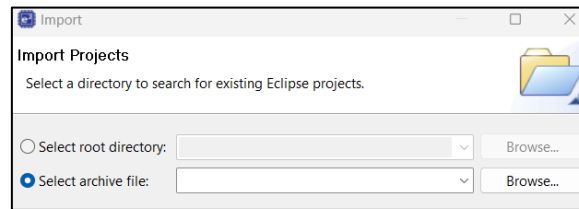


Figure 7 : Select archive file

4. Sélectionner le fichier : *Dossier_technique\e2studio\Controleur_Renesas.zip*.
5. Cocher le projet *Test_SLG* et cliquer sur « *Finish* ».

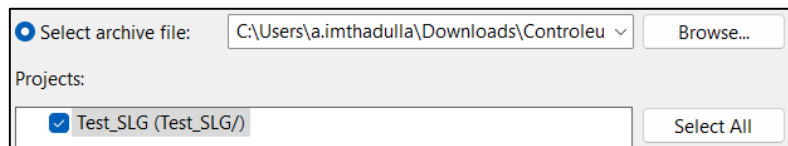


Figure 8 : Projet - Test_SLG

6. Vérifier que le fichier « *look_up_table.h* » est bien présent dans le dossier : *Test_SLG\src\synergy_gen*. Ce fichier contient la table de vérité.

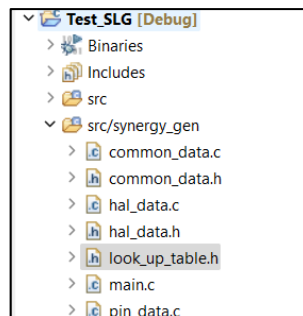


Figure 9 : look_up_table.h

7. Compiler le projet en cliquant sur « *Project* » > « *Build All* » ou en appuyant sur l'icône du marteau.

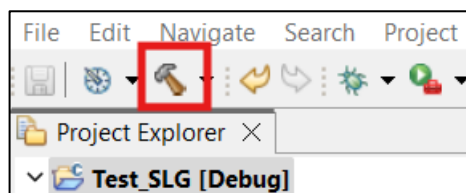


Figure 10 : Build All

L'importation est réussie s'il n'y a aucune erreur lors de la compilation.

Avant de lancer l'automatisation des tests, il est impératif de vérifier les points suivants :

- Les broches du SLG sont correctement connectées aux broches correspondantes du contrôleur Renesas S7G2.
- Les masses sont bien reliées.

3.5 Connexion de la carte de communication Série PMOD USB-UART

Elle permet la communication série entre le contrôleur et l'ordinateur, il est nécessaire de connecter la carte PMOD USB-UART. Voici le branchement :

PIN S7G2	PIN PMOD USB-UART
410	TX
411	RX
GND	GND
VDD	VDD

Tableau 3 : Branchement USB-UART

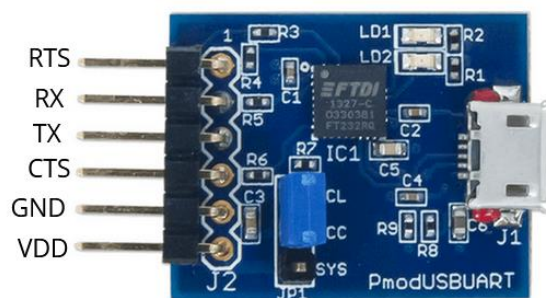


Figure 11 : Carte USB-UART

4 Automatisation des tests

Voici les étapes à suivre pour lancer l'automatisation des tests :

1. Brancher le *Pmod* à l'ordinateur avec un câble USB micro.
2. Brancher la carte Renesas S7G2 (port *Debug_USB*) à l'ordinateur avec un câble USB micro.

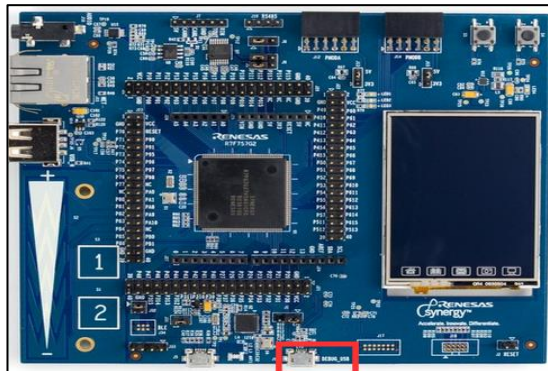


Figure 12 : Branchement Renesas S7G2 "debug"

3. Appuyer sur « *Debug* » dans e2Studio.

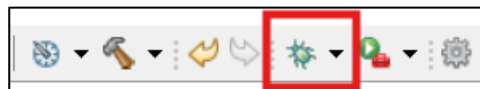


Figure 13 : Icon debug e2studio

4. Ouvrir le dossier *Dossier_technique\Python*, faire un clic droit, puis sélectionner « *Ouvrir dans un terminal* ».
5. Taper la commande suivante dans le terminal : `py.\import_serial.py`
6. Revenir sur e2Studio et lancer le débogage en appuyant sur le bouton « *Play* »



Figure 14 : e2studio - lancement du debug

7. Vérifier dans le terminal que les tests sont en cours d'exécution.

```
Test: 0, Entrée: 500 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0, Sortie: 1 0 1 1 0, Statut: Bon, Temps: 785 ticks
Test: 1, Entrée: 500 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 0, Sortie: 1 0 1 1 0, Statut: Bon, Temps: 784 ticks
Test: 2, Entrée: 500 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 1 0, Sortie: 1 0 1 1 0, Statut: Bon, Temps: 783 ticks
Test: 3, Entrée: 500 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 1 0, Sortie: 1 0 1 1 0, Statut: Bon, Temps: 782 ticks
Test: 4, Entrée: 500 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 1 1 0, Sortie: 1 1 1 1 0, Statut: Bon, Temps: 784 ticks
Test: 5, Entrée: 500 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 1 1 0, Sortie: 1 1 1 1 0, Statut: Bon, Temps: 782 ticks
Test: 6, Entrée: 500 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 1 1 0, Sortie: 1 1 1 1 0, Statut: Bon, Temps: 782 ticks
Test: 7, Entrée: 500 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0, Sortie: 1 1 1 1 0, Statut: Bon, Temps: 780 ticks
Test: 8, Entrée: 500 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 1 0, Sortie: 1 0 1 1 0, Statut: Bon, Temps: 784 ticks
Test: 9, Entrée: 500 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 1 1 0, Sortie: 1 0 1 1 0, Statut: Bon, Temps: 782 ticks
Test: 10, Entrée: 500 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 1 0, Sortie: 1 0 1 1 0, Statut: Bon, Temps: 782 ticks
Test: 11, Entrée: 500 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 0 1 1 0, Sortie: 1 0 1 1 0, Statut: Bon, Temps: 780 ticks
```

Figure 15 : Affichage des teste réaliser sur le terminal

4.1 Indicateurs LED et validation des tests

- LED orange : Test en cours.
- LED verte : Tous les tests ont été validés avec succès.
- LED rouge : Au moins un test a échoué.

4.2 Analyse des résultats

Un fichier nommé « *Uart_data.csv* » est généré automatiquement dans le même répertoire que le script Python *import_serial.py*. Ce fichier contient l'ensemble des résultats des tests et permet d'analyser en détail les validations effectuées.

4.3 Conversion et visualisation des résultats

Pour une meilleure lisibilité des données, convertir le fichier CSV en Excel en suivant ces étapes :

- Exécuter le script Python *to_excel.py* avec la commande : `py .\to_excel.py`
- Ouvrir le fichier « *excel_output.csv* » avec Excel.

Ce fichier permet de valider les scénarios 2 à 7, car tous les cas définis dans le plan de test sont inclus dans cette base de données.

5 Interprétation des résultats

5.1 Scénario de Test 1 : Validation de la Sortie nOC

Il est important de noter que notre circuit complet intègre une partie externe à la puce SLG46625. En particulier, la détection de surintensité est réalisée en dehors de la puce. Ce bloc reçoit en entrée trois signaux : I_Ph_1, I_Ph_2 et I_Ph_3, et génère en sortie le signal nOC.

Voici un schéma de principe :

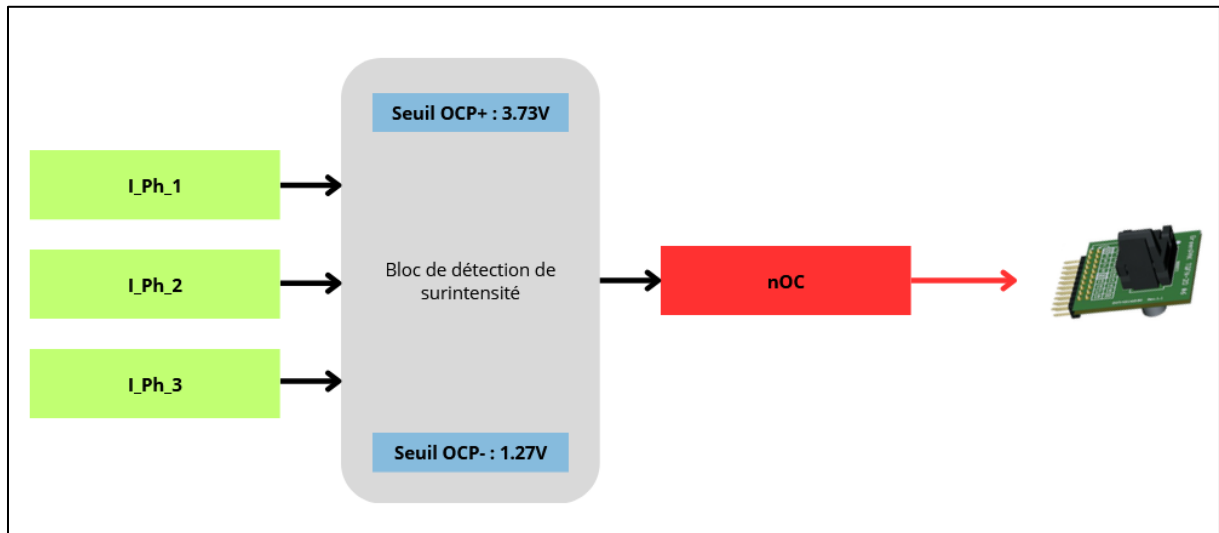


Figure 16 : Synoptique circuit externe

Si l'une des entrées I_Ph_1, I_Ph_2 ou I_Ph_3 se trouve en dehors de la plage de seuil [1.27V - 3.73V], alors la sortie nOC sera à 0. En revanche, si toutes les entrées sont comprises dans cette plage, la sortie sera à 1.

Dans cette conception, la sortie nOC du bloc externe sert d'entrée pour la puce SLG46625. C'est pourquoi les tests sur la sortie nOC peuvent être réalisés séparément.

5.1.1 Matériel nécessaire

Pour réaliser ce test, il est nécessaire de disposer des éléments suivants :

- Un générateur de tension permettant de générer :
 - Une tension constante de 5V pour alimenter les comparateurs.
 - Une tension variable entre 0V et 5V pour simuler les trois entrées.
- Un oscilloscope pour visualiser la sortie nOC.

5.1.2 Branchement

Les branchements du bloc de détection de surintensité doivent être réalisés en suivant le schéma ci-dessous ou l'architecture complète disponible dans le dossier *Dossier_technique\Architecture_complete\Circuit_complet*.

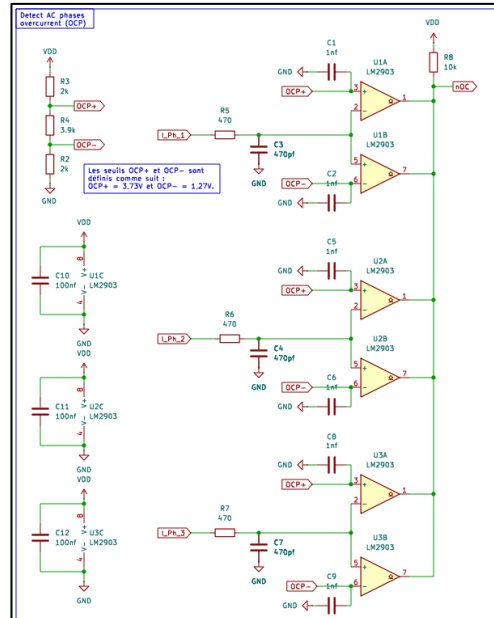


Figure 17 : Branchement circuit externe

5.1.3 Résultats Obtenue

Une fois le circuit monté, il suffit de tester les combinaisons décrites dans la table de vérité suivante et de vérifier l'état de la sortie à l'aide de l'oscilloscope.

I_Ph_1	I_Ph_2	I_Ph_3	nOC attendue	nOC obtenue
$OCP- < I_{Ph_1} < OCP+$	$OCP- < I_{Ph_2} < OCP+$	$OCP- < I_{Ph_3} < OCP+$	1	1
$OCP- > I_{Ph_1}$	$OCP- < I_{Ph_2} < OCP+$	$OCP- < I_{Ph_3} < OCP+$	0	0
$I_{Ph_1} > OCP+$	$OCP- < I_{Ph_2} < OCP+$	$OCP- < I_{Ph_3} < OCP+$	0	0
$OCP- < I_{Ph_1} < OCP+$	$OCP- > I_{Ph_2}$	$OCP- < I_{Ph_3} < OCP+$	0	0
$OCP- < I_{Ph_1} < OCP+$	$I_{Ph_2} > OCP+$	$OCP- < I_{Ph_3} < OCP+$	0	0
$OCP- < I_{Ph_1} < OCP+$	$OCP- < I_{Ph_2} < OCP+$	$OCP- > I_{Ph_3}$	0	0
$OCP- < I_{Ph_1} < OCP+$	$OCP- < I_{Ph_2} < OCP+$	$I_{Ph_3} > OCP+$	0	0
$I_{Ph_1} > OCP+$	$I_{Ph_2} > OCP+$	$I_{Ph_1} > OCP+$	0	0
$OCP- > I_{Ph_1}$	$OCP- > I_{Ph_2}$	$OCP- > I_{Ph_3}$	0	0

Tableau 4 : Scénario 1 - table des résultats

Les valeurs utilisées pour ce test sont les suivantes :

- $OCP- < I_{Ph_X} < OCP+$: $I_{Ph_X} = 3.3V$
- $I_{Ph_X} > OCP+$: $I_{Ph_X} = 1V$
- $OCP- > I_{Ph_X}$: $I_{Ph_X} = 5V$

Lors des essais avec des valeurs limites comme $I_{Ph_X} = 3.73V$ et $I_{Ph_X} = 1.27V$, il a été constaté que la sortie est à 0.

5.1.4 Conclusion du scénario 1

Ce premier test a permis de valider la partie externe du circuit, chargée de la détection de surintensité. Les résultats obtenus sont conformes aux attentes, y compris pour les cas limites.

Les scénarios 2 à 7 peuvent être validés grâce à l'automatisation du test. Pour cela, la carte de contrôle S7G2 de Renesas doit être programmée. Dans la suite de la validation, nOC sera simulée comme une valeur booléenne par le contrôleur afin de simplifier la validation de la puce SLG.

5.2 Scénario de Test 2 : Validation de la Sortie nOV

5.2.1 Objectif

- Vérifier que la sortie nOV se comporte correctement en fonction de la tension d'entrée DOUT par rapport au seuil de 1V :
 - Si DOUT est inférieur ou égal à 1V, la sortie doit être 1.
 - Si DOUT est supérieur à 1V, la sortie doit être 0.

5.2.2 Résultats Attendus

Le signal nOV étant interne à la puce SLG, il a été testé indirectement via l'interprétation de la sortie nOV_latched. Si la sortie nOV_latched est à 0, cela signifie qu'une détection de surtension a eu lieu. Cette sortie reste à 0 tant que le signal de reset n'est pas activé.

DOUT (V)	Sortie nOV_latched attendue	Sortie nOV_latched Obtenue
0 - 1	1	1
> 1	0	0

Tableau 5 : Scénario 1 - nOV_latched

5.2.3 Résultats Obtenue

Seules les sorties concernées sont présentées ici. Toutes les sorties peuvent être vérifiées dans le fichier CSV en se référant aux numéros de test.

Tests	DOUT	nOV_LATCHED	nOV_LATCHED_OUT	Resultat	Temps
2047	500	1	1	Bon	766
2048	2000	0	0	Bon	786

Tableau 6 : Scénario 2 - table des résultats

DOUT prend des valeurs entre 0 et 4095, car l'ADC de la carte S7G2 est codé sur 12 bits.

Valeur du ADC	Valeur en volt
0	0
500	0.40
2000	1.61
4095	3.3

Tableau 7 : ADC

Lorsque la tension est supérieure à 1V, la sortie nOV_LATCHED est à l'état bas. Inversement, lorsque la tension est inférieure au seuil de 1V, la sortie est à l'état haut.

5.2.4 Conclusion du Scénario 2

La sortie nOV est validée de manière indirecte avec la sortie nOV_LATCHED. Les résultats obtenus sont conformes aux attentes.

5.3 Scénario de Test 3 : Validation du Bloc de Latch

5.3.1 Objectif

Vérifier que le bloc de latch se comporte correctement en fonction des entrées RESET, noc, et nov, et que les sorties noc Latched et nov Latched suivent la table de vérité spécifiée.

5.3.2 Résultats Attendus

Les sorties doivent correspondre à la table de vérité spécifiée dans l'Excel "*Plan_De_Test*" (Scénario 3) qui se trouve dans le dossier *\Dossier_technique\Plan_de_test* .

5.3.3 Résultats Obtenue

Le tableau ci-dessous montre que la sortie nOC_LATCHED est bien à l'état bas jusqu'à ce que le signal de reset soit activé et que nOC soit à 1 (pas de surintensité).

Tests	nRESET_LATCHED	nOC	nOC_LATCHED	nOC_LATCHED_OUT	Resultat	Temps
7	0	1	1	1	Bon	780
8	1	0	0	0	Bon	784
9	1	0	0	0	Bon	782
10	1	0	0	0	Bon	782
11	1	0	0	0	Bon	780
12	1	1	0	0	Bon	782
13	1	1	0	0	Bon	780
14	1	1	0	0	Bon	780
15	1	1	0	0	Bon	778
16	0	0	0	0	Bon	784
17	0	0	0	0	Bon	782
18	0	0	0	0	Bon	782
19	0	0	0	0	Bon	780

Tableau 8 : Scénario 3 - table des résultats

5.3.4 Conclusion du scenario 3

Le bloc de latch fonctionne comme attendu. L'état de mémorisation est bien pris en compte. Le même raisonnement peut être appliqué à nOV_LATCHED.

5.4 Scénario de Test 4 : Validation de la Sortie DRV_ASC_EN_ISO

5.4.1 Objectif

- Vérifier que la sortie DRV_ASC_EN_ISO se comporte correctement selon l'équation logique définie :

DRV_ASC_EN_ISO = !(nOC_LATCHED & nOV_LATCHED & PMIC_nSS1 & DRV_HS_nFLT1 & DRV_LS_nFLT1 & SUP_DCDC_PGOOD & PS_nDIS_PWM & PS_nDIS_PWM_BIS)

- Assurer que la sortie est correctement activée ou désactivée en fonction des états des entrées associées.

5.4.2 Résultats Attendus

- La sortie DRV_ASC_EN_ISO doit être 0 uniquement lorsque toutes les entrées sont 1.
- La sortie doit être 1 pour toutes les autres combinaisons d'entrées.

Voir Excel « *Plan_De_Test* » (scénario 4)

5.4.3 Résultats Obtenue

Tes ts	SUP_DCDC_ PGOOD	PS_nDIS_P WM_BIS	PS_nDIS_ PWM	PMIC_ nSS1	DRV_HS_ nFLT1	DRV_LS_ nFLT1	DRV_ASC_E N_ISO	DRV_ASC_EN_I SO_OUT	Resul tat	Te mps
20 36	1	1	1	1	1	1	0	0	Bon	772
20 37	1	1	1	1	1	1	0	0	Bon	770
20 38	1	1	1	1	1	1	0	0	Bon	770
20 39	1	1	1	1	1	1	0	0	Bon	768

Tableau 9 : Scénario 4 - table des résultats

Lorsque tous les signaux nOC_LATCHED, nOV_LATCHED, PMIC_nSS1, DRV_HS_nFLT1, DRV_LS_nFLT1, SUP_DCDC_PGOOD, PS_nDIS_PWM et PS_nDIS_PWM_BIS sont à 1, la sortie est bien à 0. Dans les autres cas, ils sont à 1, voir le fichier CSV.

5.4.4 Conclusion du scénario 4

Les résultats obtenus sont conformes à ceux attendus. La sortie DRV_ASC_EN_ISO répond aux équations définies.

5.5 Scénario de Test 5 : Validation de la Sortie DRV_ASC_ISO_HS

5.5.1 Objectif

- Vérifier que la sortie DRV_ASC_ISO_HS se comporte comme prévu en fonction de l'équation logique donnée :

$$\text{DRV_ASC_ISO_HS} = \text{ASC} \& \text{PS_ASC_SEL} \& \text{DRV_xx_nFLT1} \& \text{SUP_DCDC_PGOOD}$$

$$\text{ASC} = \neg(\text{nOC_LATCHED} \& \text{nOV_LATCHED} \& \text{PMIC_nSS1} \& \text{DRV_HS_nFLT1} \& \text{DRV_LS_nFLT1} \& \text{SUP_DCDC_PGOOD} \& \text{PS_nASC} \& \text{PS_nASC_BIS})$$

- S'assurer que la sortie est activée ou désactivée correctement selon l'état des signaux d'entrée.

5.5.2 Résultats Attendus

- La sortie DRV_ASC_ISO_HS doit être 1 uniquement si toutes les entrées (ASC, PS_ASC_SEL, DRV_xx_nFLT1, SUP_DCDC_nUVLO) sont 1.
- La sortie doit être 0 si l'une des entrées est 0.

Voir Excel « Plan_De_Test » (scénario 5)

5.5.3 Résultats Obtenue

Tests	DRV_ASC_ISO_HS	DRV_ASC_ISO_HS_OUT	Resultat	Temps
1583	0	0	Bon	774
1584	1	1	Bon	780

Tableau 10 : Scénario 5 - table des résultats

Pour des raisons de place et de visualisation, seules les sorties sont représentées ici. Cependant, avec le numéro de test, il est possible de voir toutes les valeurs d'entrée dans le fichier CSV.

Les deux tests montrent que la sortie DRV_ASC_ISO_HS correspond bien aux équations logiques définies.

5.5.4 Conclusion du scénario 5

Les résultats obtenus avec l'automatisation des tests sont tous corrects. L'analyse du fichier CSV permet de confirmer cela.

5.6 Scénario de Test 6 : Validation de la Sortie DRV_ASC_ISO_LS

5.6.1 Objectif

- Vérifier que la sortie DRV_ASC_ISO_LS fonctionne correctement selon l'équation logique définie :

$$\text{DRV_ASC_ISO_LS} = \text{ASC} \& \text{ ! (PS_ASC_SEL \& DRV_xx_nFLT1 \& SUP_DCDC_PGOOD)}$$

- Confirmer que la sortie est activée ou désactivée selon l'état des signaux d'entrée.

5.6.2 Résultats Attendus

- La sortie DRV_ASC_ISO_LS doit être 1 lorsque ASC est 1 et que l'un des signaux (PS_ASC_SEL, DRV_xx_nFLT1, SUP_DCDC_PGOOD) est 0.
- Dans tous les autres cas, la sortie doit être 0.

Voir Excel « *Plan_De_Test* » (scénario 6)

5.6.3 Résultats Obtenue

Tests	DRV_ASC_ISO_LS	DRV_ASC_ISO_LS_OUT	Resultat	Temps
1103	1	1	Bon	776
1104	0	0	Bon	782

Tableau 11 : Scénario 7 - table des résultats

Les vecteurs d'entrée des tests 1103 et 1104 montrent les deux cas possibles de la sortie DRV_ASC_ISO_LS. Le test n'a pas d'erreur, ce qui signifie que la sortie correspond bien à ce qui était attendu.

5.6.4 Conclusion du scénario 6

Les résultats obtenus avec l'automatisation des tests sont tous corrects. L'analyse du fichier CSV permet de confirmer cela.

5.7 Scénario de Test 7 : Vérification Temporelle des Sorties de Protection

Ce scénario vise à vérifier la réponse temporelle des sorties de protection en fonction des entrées de détection de surtension, sous-tension DCDC et autres défauts, conformément aux exigences du cahier des charges.

Caractéristique	Valeur à visée (Niveau)
La logique de protection doit déclencher la sortie ASC très rapidement suivant la réception de l'entrée de détection de surtension.	< 10 μ s
La logique de protection doit définir la sortie de déclenchement FW suivant la réception de l'entrée de détection de sous-tension DC/DC.	< 100 μ s
La logique de protection doit correctement définir les sorties de déclenchement LS-ASC ou HS-ASC	< 100 μ s

Tableau 12 : Contrainte temporelle

5.7.1 Résultats Attendus :

- Le temps de déclenchement de la sortie ASC doit être inférieur à 10 μ s, conformément aux spécifications.
- Le temps de déclenchement de la sortie FW doit être inférieur à 100 μ s, conformément aux spécifications.
- Le temps de déclenchement des sorties LS-ASC ou HS-ASC doit être inférieur à 100 μ s, conformément aux spécifications.

5.7.2 Résultats Obtenue

L'automatisation du test a permis de relever tous les temps de réponse des sorties pour chaque test. Cela permet de valider rapidement les contraintes temporelles. En effet, le temps de réponse indiqué pour chaque test est en *ticks*. Le S7G2 fonctionne à 240 Hz, ce qui donne un *tick* = 8,33 ns. Dans le fichier CSV, il suffit de prendre la valeur maximum de la colonne temps, ce qui donne 801 *ticks* = 6 672 ns = 6,672 μ s. Les contraintes temporelles sont largement respectées.

5.7.3 Conclusion du scénario 7

Les contraintes temporelles sont vérifiées à l'aide de la carte S7G2 de Renesas. Les temps de réponse relevés sont bien inférieurs à ceux fixés dans le cahier des charges. De plus, le

déclenchement de la sortie ASC est mesuré indirectement en mesurant les temps de réponse des sorties DRV_ASC_ISO_LS, DRV_ASC_ISO_HS et DRV_ASC_EN_ISO.

6 Tests Non Réalisés

Malgré les avancées significatives, certains tests n'ont pas pu être effectués :

- **Résilience aux Températures** : Les tests sous différentes conditions thermiques (de -40°C à 85°C) n'ont pas été réalisés faute de temps et d'infrastructure adaptée. Ces tests auraient permis de valider le comportement du circuit dans des environnements extrêmes.
- **Variation de la Tension d'Alimentation** : L'analyse des performances en cas de fluctuations importantes de la tension n'a pas été menée. Ces tests sont cruciaux pour garantir la stabilité de la solution dans des conditions de fonctionnement réelles.

7 Les états de l'onduleur

Le signal ASC permet de contrôler les trois signaux qui régissent l'état du système. Ces signaux sont les suivants :

- DRV_ASC_EN_ISO : Ce signal indique l'état de l'onduleur. Si ce signal est à l'état bas, les signaux de commande sont autorisés. Si ce signal est à l'état haut, les signaux de commande sont ignorés.
- DRV_ASC_ISO_HS : Ce signal détermine l'état des transistors haute tension (High-Side ou HS). S'il est à 0, les transistors HS sont ouverts. S'il est à 1, les transistors HS sont fermés.
- DRV_ASC_ISO_LS : Ce signal détermine l'état des transistors basse tension (Low-Side ou LS). S'il est à 0, les transistors LS sont ouverts. S'il est à 1, les transistors LS sont fermés.

Voici tableau récapitulatif des états de l'onduleur :

État de l'onduleur	DRV_ASC_EN_ISO	DRV_ASC_ISO_HS	DRV_ASC_ISO_LS
PWM	0	X	X
Roue libre	1	0	0
ASC_LS	1	0	1
ASC_HS	1	1	0
Interdit	1	1	1

Tableau 13 : Etats de l'onduleur

Voici une explication des états possible de l'onduleurs :

- PWM : Les signaux de commande (PWM) sont actifs. L'onduleur fonctionne normalement.
- Roue libre : Le mode ASC (Active Short Circuit) est activé, mais tous les transistors sont ouverts. Les signaux de commande sont ignorés et aucun courant ne circule.
- ASC_LS : Le mode ASC est activé. Tous les transistors basse tension (Low-Side ou LS) sont fermés, tandis que les transistors haute tension (High-Side ou HS) restent ouverts. Les signaux de commande sont ignorés.
- ASC_HS : Le mode ASC est activé. Tous les transistors haute tension (High-Side ou HS) sont fermés, tandis que les transistors basse tension (Low-Side ou LS) restent ouverts. Les signaux de commande sont ignorés.
- Interdit : Tous les transistors haute tension (High-Side ou HS) et basse tension (Low-Side ou LS) sont fermés simultanément, ce qui provoque un court-circuit destructif pour l'onduleur. Cet état est interdit.

L'automatisation des tests a permis de vérifier que l'onduleur n'est jamais dans l'état interdit. Le fichier CSV ne contient pas cette combinaison, ce qui est déjà un bon point.

8 Conclusion

Ce rapport de validation permet de confirmer que les contraintes définies dans le cahier des charges sont respectées. L'automatisation des tests a permis de réaliser une vérification efficace et reproductible. Les états interdits de l'onduleur n'ont jamais été observés, validant ainsi la sécurité du système. Certains tests, comme la résistance aux températures extrêmes et les variations de tension d'alimentation, n'ont cependant pas pu être menés faute de moyens et de temps. Ces aspects mériteraient une étude complémentaire pour assurer une validation complète du dispositif.