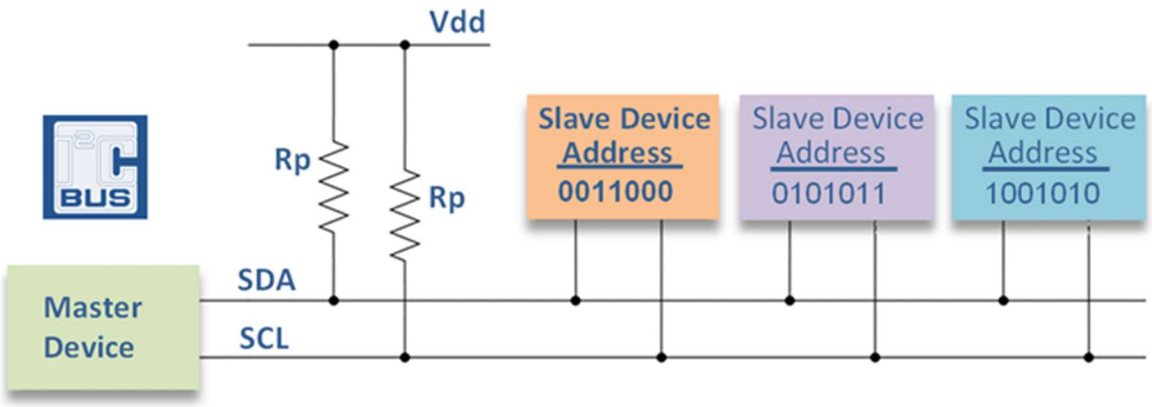
LE BUS I2C



1. Vitesse de transmission : Le bus I2C peut fonctionner à différentes vitesses, allant de 100 kHz à 5 MHz pour les versions les plus rapides.
2. Topologie : Le bus I2C utilise une topologie de bus en série, où plusieurs périphériques peuvent être connectés à deux lignes de signal : SDA (Serial Data) et SCL (Serial Clock). Il peut y avoir plusieurs maîtres et esclaves sur le bus.
3. Communication bidirectionnelle : Le bus I2C permet la communication bidirectionnelle, ce qui signifie que les données peuvent être transmises dans les deux sens entre les maîtres et les esclaves. Les maîtres envoient des données aux esclaves et reçoivent des données des esclaves.
4. Adresses : Chaque esclave sur le bus I2C a une adresse unique qui est utilisée pour identifier l'esclave lors de la communication. L'adresse peut être soit de 7 bits ou de 10 bits, selon la version du bus.
5. Protocole : Le bus I2C utilise un protocole de communication simple mais efficace. Le maître envoie un signal de départ pour initier la communication, suivi de l'adresse de l'esclave. Une fois que l'esclave a été sélectionné, les données sont transmises sur la ligne SDA, avec des signaux de synchronisation sur la ligne SCL.

Avantages :

- Il permet la communication entre plusieurs périphériques sur un seul bus.
- Il est facile à mettre en œuvre et à configurer.
- Il utilise seulement deux fils pour la communication de données et de synchronisation, ce qui limite le nombre de pins nécessaires.
- Il est largement utilisé dans l'industrie électronique, donc il y a de nombreux composants disponibles qui utilisent le protocole I2C.

Inconvénients :

- Il a une bande passante limitée par rapport à d'autres protocoles de communication.
- Il est généralement utilisé pour des distances de communication relativement courtes en raison de la limite de capacitance des bus I2C.

- Il peut être affecté par les interférences électromagnétiques (EMI) et les bruits électriques, ce qui peut perturber la communication entre les périphériques.
- Le protocole de communication ne fournit pas de vérification d'erreur intégrée, ce qui signifie que les erreurs de communication peuvent ne pas être détectées sans ajout de code spécifique dans le logiciel.

Le fonctionnement du bus I2C

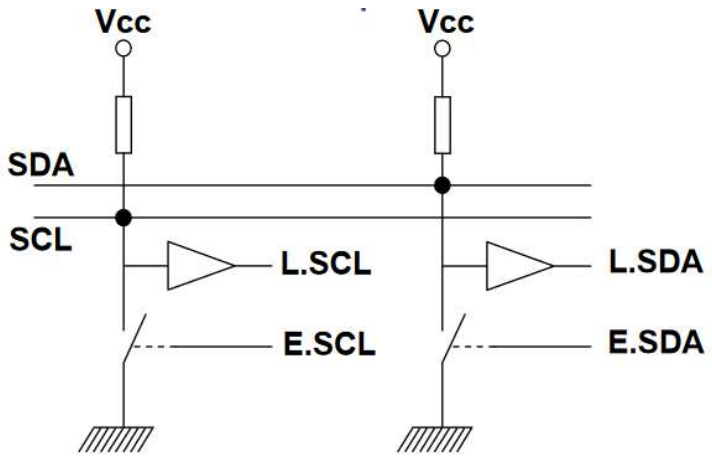
1. Nécessité de la résistance de pull up

Le bus I2C est bidirectionnel, ce qui implique que le maître et l'esclave peuvent communiquer via la ligne SDA de l'I2C. Il est crucial d'éviter tout conflit pouvant provoquer un court-circuit :

- Le maître établit un niveau logique « 1 » sur la ligne SDA => Vdd
- L'esclave établit un niveau logique « 0 » sur la ligne SDA => GND

Afin d'éviter cela, on empêche les dispositifs d'imposer un niveau logique « 1 » sur la ligne SDA. Ce niveau sera défini par défaut grâce à une résistance de tirage (pull-up). Ainsi, si deux entités tentent de communiquer simultanément sur le bus, le matériel est protégé.

Le matériel est donc protégé matériellement contre les tentatives de communication simultanées, mais cela introduit un inconvénient en raison des imperfections présentes sur les périphériques et les lignes de communication qui génèrent des éléments capacitifs. L'ajout d'une résistance de tirage crée un circuit RC sur la ligne de communication, limitant ainsi la vitesse du bus et le nombre maximal de périphériques pouvant être connectés.



2. Dimensionnement de la résistance de pull up

La valeur de la résistance de pull-up pour le bus I2C doit être choisie en fonction de la capacité de charge totale des périphériques connectés au bus, ainsi que de la vitesse de transmission des données souhaitée.

La capacité de charge totale est déterminée par la somme des capacités d'entrée des périphériques connectés au bus. En général, il est recommandé de limiter la capacité totale à 400 pF pour les vitesses de transmission de 100 kHz, et à 100 pF pour les vitesses supérieures.

Pour calculer la valeur de la résistance de pull-up, on peut utiliser la formule suivante :

R_pullup = (Vcc - V_low) / I_pullup

Où :

- Vcc est la tension d'alimentation du bus I2C
- V_low est la tension de niveau bas pour le signal SDA
- I_pullup est le courant de pull-up souhaité

La valeur typique du courant de pull-up est de 2 à 3 mA pour une vitesse de transmission de 100 kHz, et de 1 mA pour des vitesses plus élevées.

Par exemple, pour une tension d'alimentation de 5 V, une tension de niveau bas de 0,4 V, et un courant de pull-up de 2 mA, on peut calculer la valeur de la résistance de pull-up comme suit :

R_pullup = (5 V - 0,4 V) / 0,002 A = 2300 Ω

Il est important de noter que la valeur de la résistance de pull-up peut varier en fonction des conditions de fonctionnement réelles du bus I2C, donc il est recommandé de faire des tests de validation pour s'assurer que le bus fonctionne correctement avec les périphériques connectés.

Le protocole du bus I2C

Une trame I2C (Inter-Integrated Circuit) est utilisée pour la communication série synchrone entre des composants électroniques. Elle est constituée de plusieurs éléments qui permettent d'établir la communication et de transmettre les données entre les périphériques connectés au bus I2C. Voici les différents éléments d'une trame I2C :

1. Signal de départ (Start) : La trame I2C débute toujours par un signal de départ émis par le maître (ou initiateur) du bus. Il s'agit d'une transition du niveau haut (SCL et SDA à l'état haut) à un niveau bas sur la ligne de données (SDA) pendant que la ligne d'horloge (SCL) reste à l'état haut.
2. Adresse du périphérique : Après le signal de départ, le maître envoie l'adresse du périphérique avec lequel il souhaite communiquer. L'adresse est généralement composée de 7 bits, mais peut être étendue à 10 bits.

1. Les 7 premiers bits sont l'adresse réelle du périphérique
2. Le huitième bit indique si le maître souhaite écrire (0) ou lire (1) à partir du périphérique.

3. Bit d'acquittement (ACK) : Après avoir envoyé l'adresse du périphérique, l'émetteur (maître ou esclave) attend un bit d'acquittement (ACK) du récepteur. Le récepteur (périphérique cible) tire la ligne de données (SDA) bas pendant le 9e cycle d'horloge pour indiquer qu'il a reçu les données avec succès, ou la laisse à l'état haut pour indiquer un non-acquittement (NACK). Le bit d'acquittement est utilisé pour vérifier la présence du périphérique sur le bus.
4. Données : Après l'adresse et la réception de l'ACK, le maître peut envoyer des données au périphérique ou en recevoir du périphérique. Les données sont transmises sous forme de paquets de 8 bits (octets), avec le bit de poids faible (LSB) d'abord. Après l'envoi de chaque octet, le récepteur renvoie un bit d'acquittement (ACK ou NACK) pour indiquer si les données ont été reçues correctement ou non.
5. Signal d'arrêt (Stop) : Une fois que toutes les données ont été transmises ou reçues, le maître envoie un signal d'arrêt pour indiquer la fin de la trame I2C. Le signal d'arrêt est une transition du niveau bas au niveau haut sur la ligne de données (SDA) pendant que la ligne d'horloge (SCL) reste à l'état haut.

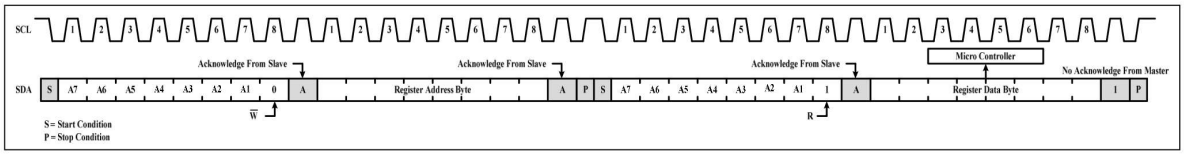


Figure 7 Reading from IS31FL3239