

**NOM : Correction**

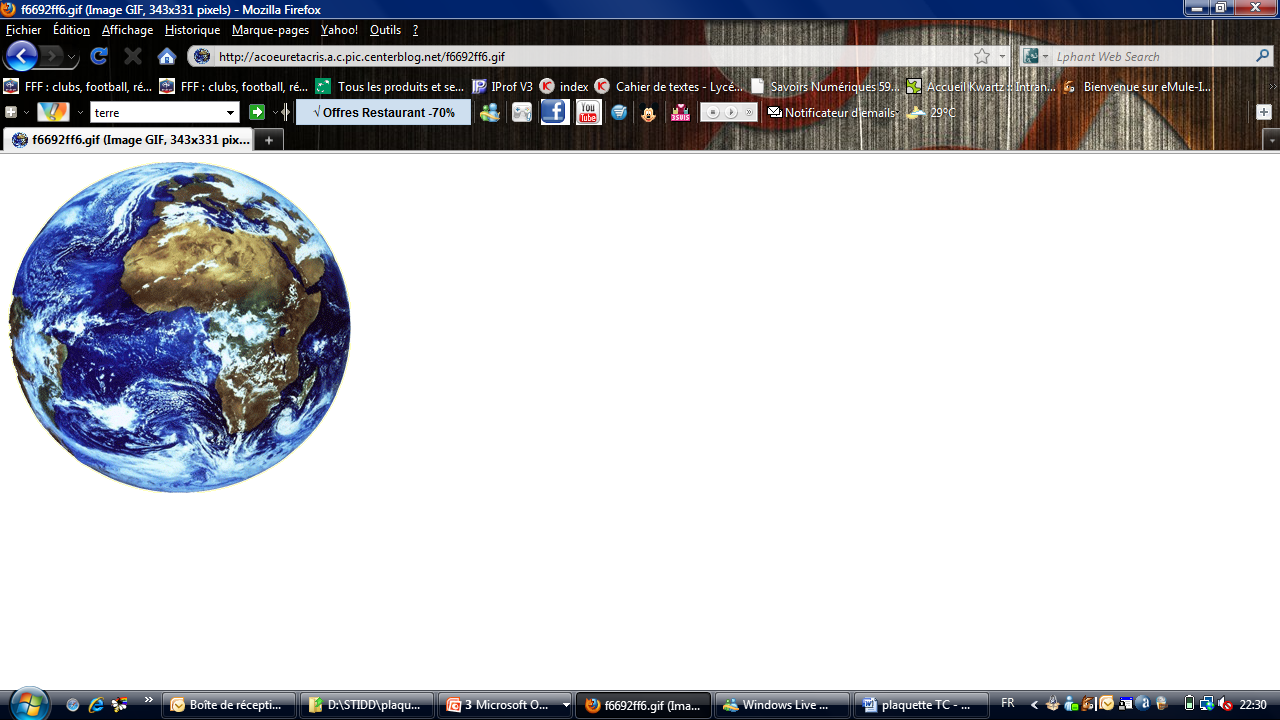
**Prénom :**

**Classe : TSTI2D\_SIN**

**Durée :**

**Note : /**

Baccalauréat STI2D



**Sciences et**

**Technologie de**

**l’Industrie**

**Développement**

**Durable**

**Et du**

Enseignement de Spécialité

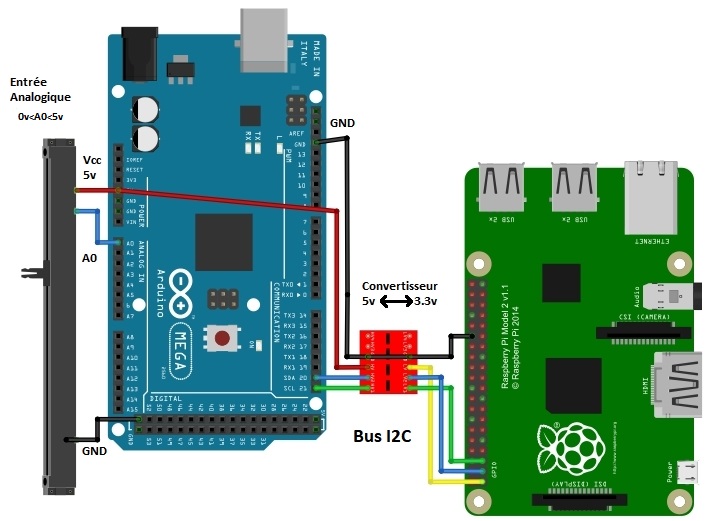
**Système d’Information et Numériques**

TP Liaison i2c

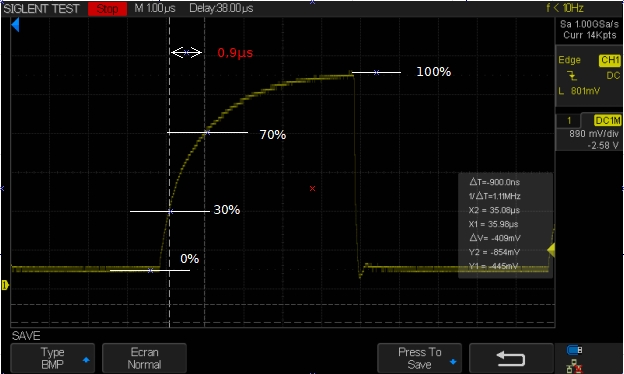
Raspberry Pi – Arduino Méga 2560 R3

Publication dans une page Web de l’entrée A0

Etude Dynamique de la communication



**4 : Étude Dynamique (Rpmax)**

****

***Relevé effectué avec un résistor Rp = 1.72K mesuré ( 10K // 10K // 2,7K = 1,753K théorique ) coté Arduino ( 5v ) ligne SCL ( câble de 5m )***

***Ouvrer le document réponse\_Etude Dynamique ele pour répondre à cette partie.***

**Les valeurs des résistors de pull-up permettront de régler les paramètres "statique et dynamique" du bus I2C.**

***L'étude dynamique permet de déterminer la valeur maximale des résistors de pull-up: Rp***

**4.1: Analyse des valeurs dynamiques du bus I2C**

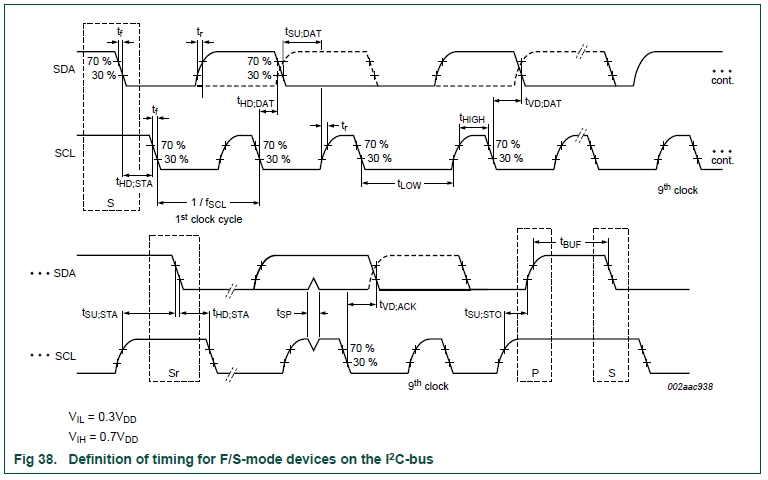
***La valeur maximale du résistor Rp dépendra de la constante de temps des lignes SDA et SCL.***

La communication sur le bus I2C doit respecter des contraintes temporelles, cela correspond aux paramètres dynamiques du bus.

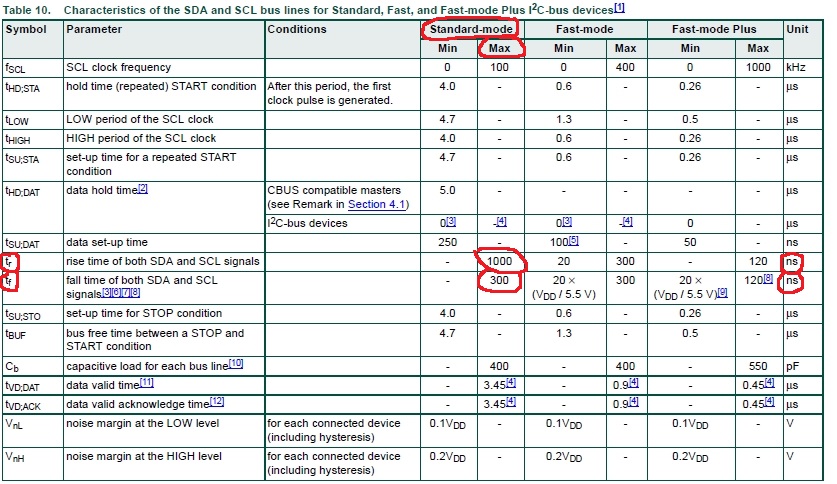
Sur les chronogrammes suivants 2 temps sont particulièrement à mesurer : tf et tr des lignes SDA et SCL.

tr = « rise time » temps de monté ou front montant

tf = « fall time » temps de descente ou front descendant

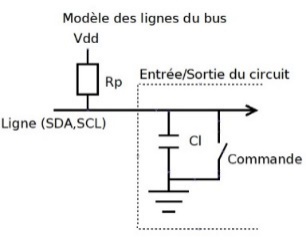
***Les temps tf et tr sont mesurés entre 30% et 70% de la valeur finale sur les lignes SDA ou SCL.***

**Relever dans la documentation du bus I2C les valeurs maximales de ces deux paramètres : tf et tr :**



Les fronts montants et descendant dépendent de la constante de temps des lignes du bus.

La constante de temps ( ) est obtenue par le produit de la valeur du résistor de pull-up **Rp** multiplié par la capacité de la ligne **Cl**.



La capacité Cl (capacité de liaison des lignes SDA et SCL) ne peut excéder 400pF !

**Hypothèse Cl = 30pF**

**Calculer** la valeur de la constante de temps ( ) des lignes SDA et SCL du montage :

Vous utiliserez les valeurs réelles mesurées obtenues dans l'étude statique !

1 Pour la carte Raspberry :

2 Pour la carte Arduino :

**4.2: Relever les fronts « montant et descendant » de la ligne SDA à l’oscilloscope.**

**P**our effectuer les relevés des temps de monté et descente de la ligne SDA, il faut dé calibrer la voie de l’oscilloscope pour faire correspondre la variation du signal de 0 à100% (souvent indiqué sur l’écran de l’oscilloscope). Les temps seront à mesurer entre 30% et 70% de la valeur finale, comme indiqué dans la documentation fig38 page précédente.

On cherche à déterminer la valeur maximale du résistor Rp qui répond aux contraintes temporelles des temps tf et tr. Pour régler la constante de temps (régler Rp), il est souhaitable d’utiliser une boite à décade afin d’obtenir rapidement une valeur maximale pour la résistance de pull-up.

Toutes les valeurs obtenues dépendent de la longueur de ligne que vous avez réalisée. Si votre ligne est très longue, que les mesures des temps ne correspondent pas aux critères temporels du bus i2c, il faudra utiliser des drivers de ligne de type exemple :  **P82B715 I2C Bus Extender (voir Activité : ressources complémentaires)**

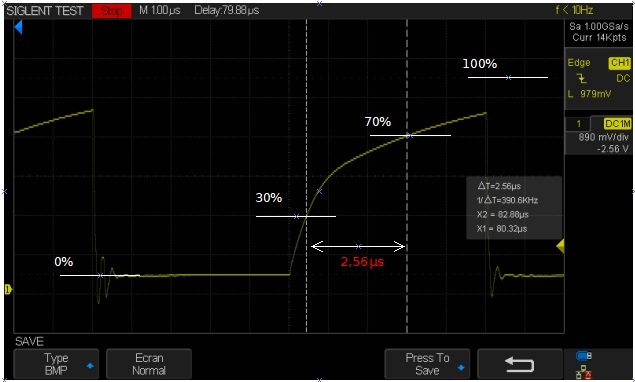
***Coté Arduino 5V***

***Coller une image (par essai) de l’écran de votre oscilloscope pour chaque relevé***

**4.21 : Mesure de tr (rise time) front montant**

**Premier essai : Sans modification du résistor Rp soit : Rp=5KΩ (RN1C//R10)**

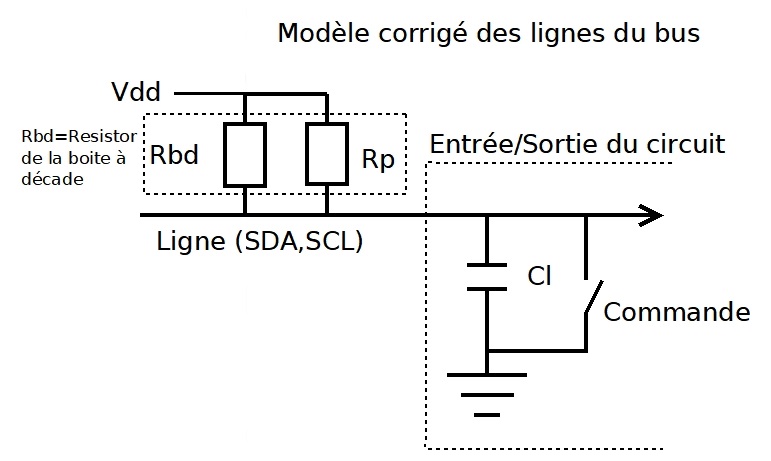
***Remplacer cette image par celle de l’écran de votre oscilloscope***



**Conclusion : avec t r=2,56µs on n’est pas assez rapide (tr<1µs)**

**Si le relevé précédent ne correspond pas aux attentes, proposer une autre valeur de Rp.**

**Pour déterminer la valeur adéquate du résistor Rp on va utiliser une boite à décade montée en parallèle sur ce résistor.**

****

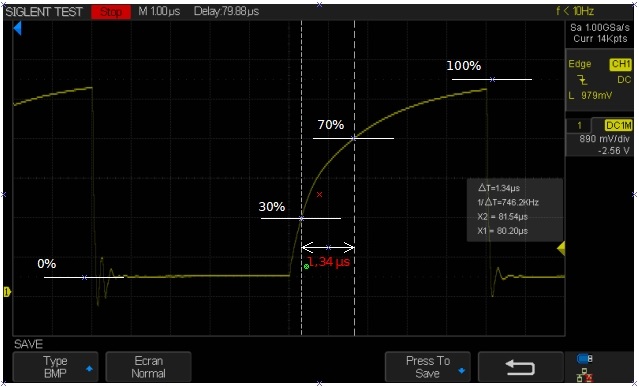
**Modifier la valeur de Rp pour répondre au contrainte temporelles de tr en ajustant vote boite à décade.**

**Le résistor Rp= ? KΩ ( RN1C // R10 // ??) avec ??=valeur de la boite à décade**

**Deuxième essai : (facultatif)**

**On ajoute en parallèle sur le résistor Rp un résistor de 4,7KΩ**

**soit (10K // 10K // 4,7K)**

**On obtient un résistor théorique de valeur Rp=2,42KΩ**

**Avec t r=1,34µs on n’est toujours pas assez rapide tr<1µs**

**On doit diminuer la valeur du résistor Rp=2,42KΩ**

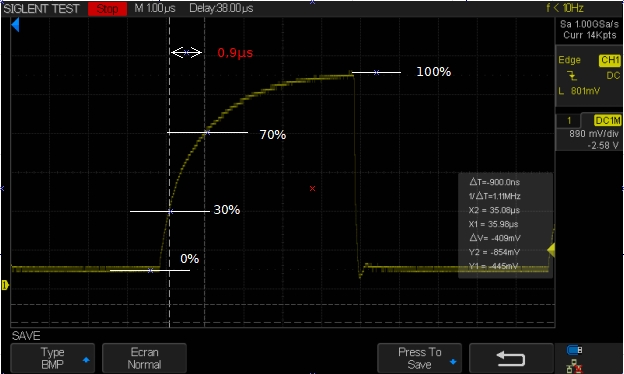
**Troisième essai : Réglage boite à décade 2,85 KΩ (remplacé par un résistor de 2,7KΩ Normalisé)**

**On ajoute en parallèle sur le résistor Rp un résistor de 2,7KΩ**

**soit (10K // 10K // 2,7K)**

**On obtient un résistor théorique de valeur Rp=1,753KΩ**

***Remplacer cette image par celle de l’écran de votre oscilloscope***



**Avec t r=0,9µs on respecte la condition tr<1µs**

**4.22 : Mesure de tf (fall time) front descendant**

***Remplacer cette image par celle de l’écran de votre oscilloscope***



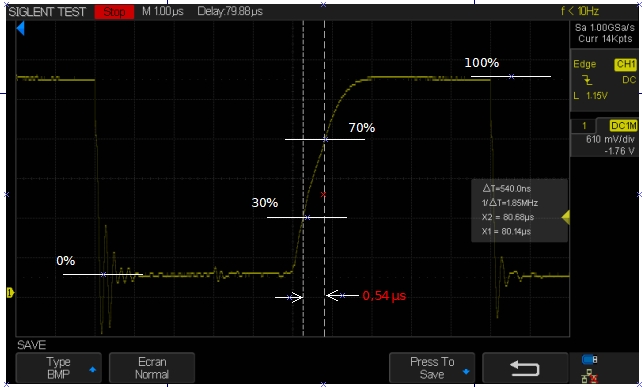
**Avec t f=0,031µs on respecte la condition tf<0,3µs**

***Coté Raspberry 3,3V***

**4.23 : Mesure de tr (rise time) front montant**

**Le résistor Rp=1,52KΩ (R2//R24)**

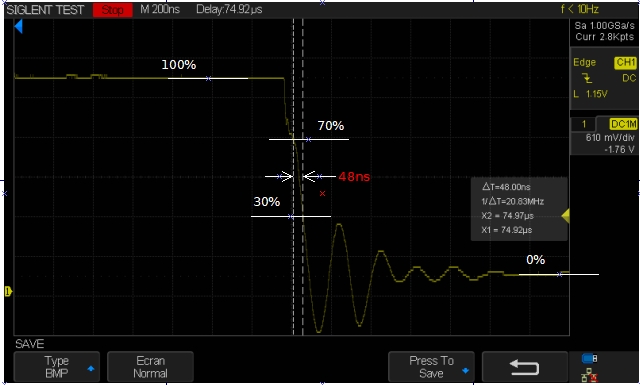
***Remplacer cette image par celle de l’écran de votre oscilloscope***



**Avec t r=0,54µs on respecte la condition tr<1µs**

**4.24 : Mesure de tf (fall time) front descendant**

***Remplacer cette image par celle de l’écran de votre oscilloscope***



**Avec t f=0,048µs on respecte la condition tf<0,3µs**

**4.3 : Compléter le tableau suivant avec la valeur des fronts montant et descendant de la ligne SDA et conclure.**

|  |  |  |
| --- | --- | --- |
| **Ligne SDA** | **Coté Arduino (5v)** | **Coté Raspberry (3,3v)** |
| **Rp\_Theorique\_cal** | **1,7** | **1,07** |
| **Rp\_mesuré** | **1,6** | **1,52** |
| **Front Montant tr** | **840 ns** | **540 ns** |
| **Front Descendant tf** | **31 ns** | **48 ns** |

**Conclusions :**

**Conclure** sur les valeurs temporelles tf et tr obtenues.

Si vos relevés présentent une incompatibilité temporelle, proposer une solution pour y remédier.

Pour obtenir un temps tr suffisamment faible coté Arduino sous 5v, il a fallu baisser la valeur du résistor à 1,6KΩ mesuré (ajout de 2 résistance de 4,7KΩ chacune), valeur légèrement inférieure à Rpmin = 1,7KΩ. Le temps de descente tf et respecté.

Pour le coté Raspberry il n’y a pas de problème, les contraintes temporelles tf et tr sont respectées