# Bureau d’étude

[Bureau d’étude 1](#_Toc16158)

[Introduction 2](#_Toc24116)

[Matériels 2](#_Toc30307)

[Affichage de l’écran LCD 2](#_Toc9520)

[Méthodes 3](#_Toc27564)

[Résultats 3](#_Toc13503)

[Conclusion 3](#_Toc1125)

[La tension du panneaux solaire 3](#_Toc21380)

[Méthodes 3](#_Toc12260)

[Résultats 4](#_Toc29112)

[Conclusion 4](#_Toc19158)

[Le courant du panneaux solaire 4](#_Toc14685)

[Méthodes 4](#_Toc16144)

[Résultats 4](#_Toc15698)

[Conclusion 4](#_Toc19180)

[Commande du hacheur 4](#_Toc9648)

[Méthodes 4](#_Toc29854)

[Résultats 5](#_Toc27730)

[Conclusion 5](#_Toc4391)

[Annexe 5](#_Toc4741)

[Sources 5](#_Toc17351)

# Introduction

Notre projet de bureau d’étude porte sur le rechargement d’une batterie grâce à un panneau solaire. Le système disposera d’un afficheur LCD qui affichera la tension ainsi que le courant instantané au bornes du panneaux solaire.

Pour realiser le projet, nous avons décidés de la découper en plusieurs étapes :

- Afficher du texte et des variables sur l’écran LCD

- Récupérer la tension max du panneaux solaire

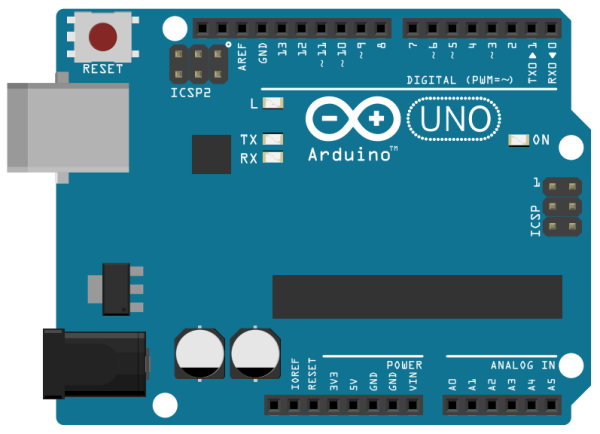
- Récupérer le courant max du panneaux solaire

- Commander le hacheur

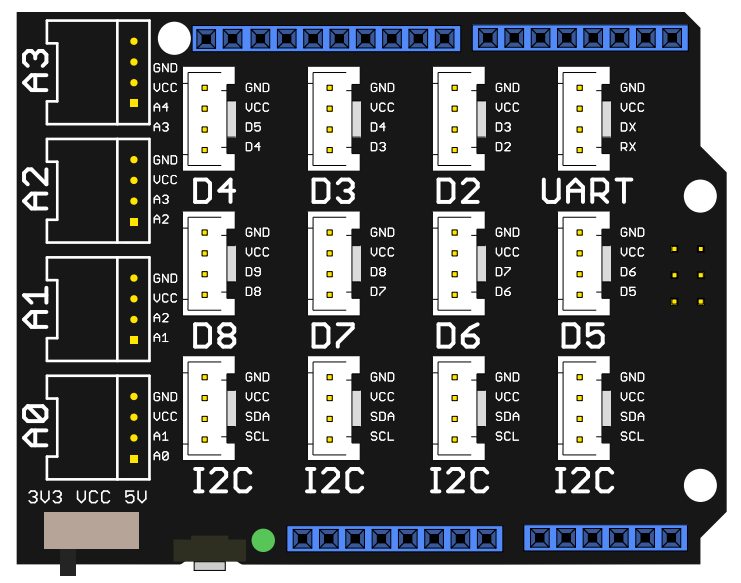
# Matériels

Pour la réalisation du projet nous disposons de panneaux solaire se situant sur le toit du bâtiment, pour notre projet nous n’avons besoin que d’un panneaux.

Pour la gestion du programme de rechargement nous disposon d’une carte Arduino Uno



Pour simplifier le câblage nous disposons d’une extension, il s’agit d’une carte shield



Le raccordement entre les deux composants est très simple il suffit seulement de branche le shield sur la carte Arduino car il possède exactement les mêmes connecteurs.

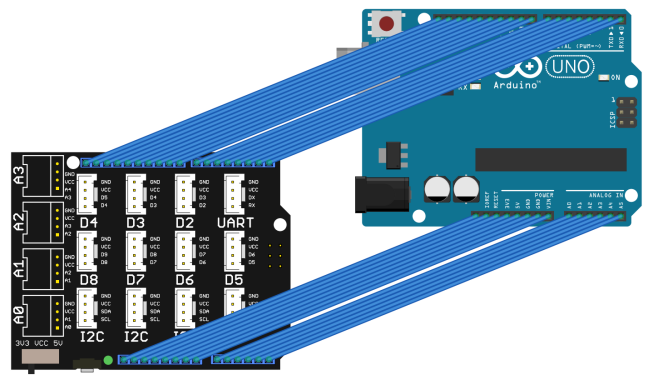


Figure X : Schéma de raccordement de la carte Arduino et le Shield

# Affichage de l’écran LCD

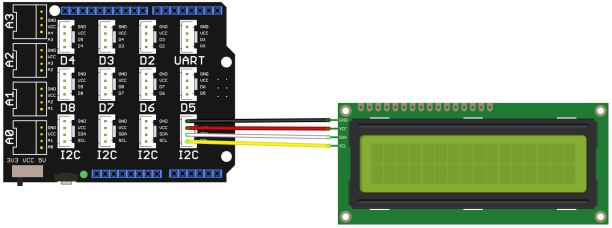
Dans notre système, nous allons afficher des informations instantanée du panneaux solaire tel que la tension et le courant maximale que peut fournir le panneaux solaire. Pour ce faire nous allons utiliser un afficheur LCD qui peut afficher 16 caractère sur 2 lignes de couleur noir sur fond jaune. L’afficheur se connecte en I2C ce qui simplifie grandement le câblage.

## Méthodes

Pour afficher du texte sur l’écran il nous faut raccorder l’afficheur à la carte Arduino, étant donné que nous avons la carte d extension Shield, nous avons simplement représenté le raccordement au niveau de l’extension. Puisque que l’afficheur se raccorde en I2C, nous utilisons que 4 broches comme indiqué dans le tableau de la figure X.

|  |  |  |
| --- | --- | --- |
| Port Arduino | Broche | Couleur |
| GND (I2C) | GND | Noir |
| Vcc (I2C) | Vcc | Rouge |
| SDA (I2C) | SDA | Blanc |
| SCL(I2C) | SCL | Jaune |

La figure X elle représente le raccordement du point de vue graphique avec la représentation des deux composants.



Pour ce qui en est de la programmation le principe de la communication I2C est de réduire le nombre de fils et d’utiliser le système de bue de communication. Nous devons donc ajouter une librairie qui a déjà était créer pour ce système d’affichage.

## Résultats

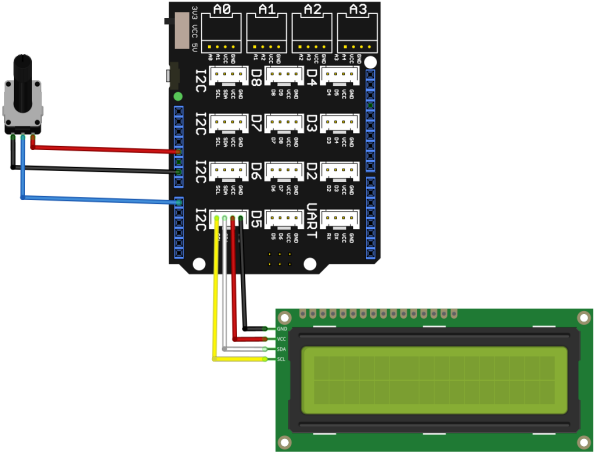
L’affichage est clair et le texte est suffisamment lisible, les lettres qui sont en majuscules sont facilement differentiable de celles en minuscules. Le premier programme réaliser pour afficher du texte et des variables étant long et pas pratique pour afficher différents textes et variables à la suite, nous avons créé deux fonctions externe à la fonction void loop afin de simplifier et alléger la fonction principale.

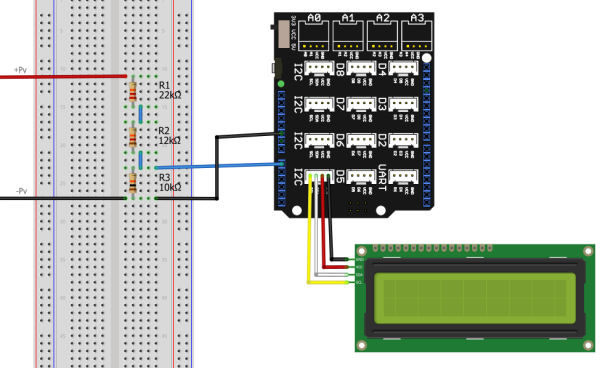
## Conclusion

# La tension du panneaux solaire

## Méthodes

Le panneaux solaire fournit en permanence une tension que va allumé la batterie, pour calculer la tension maximale qui peut être produite par le panneaux solaire nous avons mit en place un ont diviser de tension. Le panneaux solaire peut fournir une tension maximale de 22V mais la carte Arduino ne peut réceptionné maximum 5V au borne de ses ports Analogique, nous devons donc abaisser la tension pour que la tension maximale du panneaux solaire soit perçu comme 5V pour la carte Arduino. D’où le système de pont diviseur de tension. Au niveau du calcul il faut faire VCC\*(R2/(R1+R2)) où R1 est la résistance en amont du système alimenté par la tension VC et R2 celle reliée à la masse. Entre les deux résistance, nous tirons un fil jusqu’à l’entrée A0 de la carte Arduino.



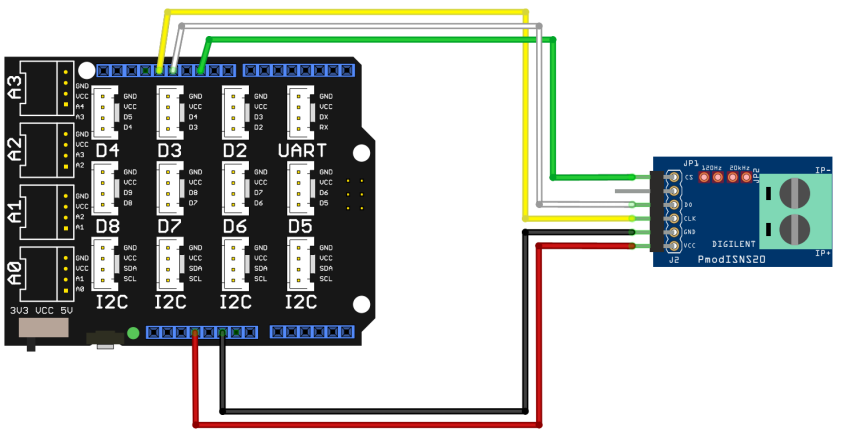


## Résultats

## Conclusion

# Le courant du panneaux solaire

## Méthodes



|  |  |  |
| --- | --- | --- |
| Port Arduino | Broche | Couleur |
| +3.3V | Vcc | Rouge |
| GND | GND | Noir |
| 13 | Clock | Jaune |
| 12 | Miso | Blanc |
| 10 | Chip Select | Vert |

Figure X : Tableau de câblage du capteur de courant

## Résultats

## Conclusion

# Commande du hacheur

## Méthodes

Pour la commande du hacheur nous allons la commander grace à la commande du transistor qui se trouve dans le hacheur. On le commandera à une fréquence supérieur à 20kHz pour ne pas entendre de bruit, de ce fait nous avons décider de l’utiliser à 50kHz minimum selon ce que peut fournir la carte Arduino.

Au niveau de la carte Arduino nous allons utiliser la fonction PWM (Pulse Width Modulation) ce qui signifie Modulation de largeur d’impulsion, elle nous permettra de créer le signal de commande du hacheur. En fonctionnement normale la fréquence du PWM est de 500Hz, nous avons donc du effectuer des modifications pour augmenter sa fréquence au maximum soit 62 500Hz.

Pour cela il faut augmenter la fréquence de l’horloge des sorties PWM et cette modifications ne peut s’applique que pour certaine sorties appelés Timers. La carte Arduino en possèdes 3, Timer 0 est pour les broches 5et 6, Timer 1 pour les broches 9 et 10 et le Timer 2 pour les broches 3 et 11.

## Résultats

## Conclusion

# Annexe

# Sources

1. <https://arduino-france.site/pwm-arduino/>
2. <https://discordapp.com/channels/@me/1037786304852787281/1084463527621828669>
3. <https://www.youtube.com/watch?v=uBVdJnP87ho&t=75s>
4. <https://www.youtube.com/watch?v=YBNEXWp-gf0>
5. <https://www.locoduino.org/spip.php?article89>