



وزارة التعليم العالي والبحث العلمي  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
جامعة عبد الحميد ابن باديس مستغانم  
Université Abdelhamid Ibn Badis de Mostaganem  
كلية العلوم و التكنولوجيا  
Faculté des Sciences et de la Technologie



N° d'ordre : M...../GE/2018

## MEMOIRE DE FIN D'ETUDE DE MASTER ACADEMIQUE

Filière : Électronique

Spécialité : Electronique des systèmes embarqués

### *Thème*

**REALISATION D'UNE CARTE DE COMMANDE D'UNE  
STATION DE POMPAGE A BASE D'ARDUINO**

Présenté par :

- DERBOUZ Mahdjouba Asma
- Amara Mohammed

*Soutenu le 01 / 07 / 2018 devant le jury composé de :*

Président : Mr .BEKLOUZ.L

Examineur1 : Mr . RESFA.A

Examineur2 : Mr .HENNIS

Encadreur : Mme : A.ABBAD

Année Universitaire : 2017 / 2018

<b>Introduction générale</b> .....	9
<b>Chapitre I ; présentation de la Station de pompage</b> .....	10
1. Introduction .....	11
2. Station de pompage .....	11
2.1 Les composantes principales du système de transfert.....	11
3. Principe de fonctionnement d'une pompe .....	12
4. Variation de la consommation d'eau : .....	12
5. Présentation des tarifs de la SONELGAZ .....	12
6. Conclusion .....	14
<b>Chapitre II ; Composants de la carte</b> .....	15
1. Introduction .....	16
2. La carte ARDUINO UNO .....	17
2.1. Caractéristiques techniques .....	19
2.2 Entrées et sorties.....	19
2.3 Communication .....	20
3. Afficheur à cristaux liquides.....	20
4. Connexion de l'afficheur sur la carte .....	21
5. Horloge temps réel .....	24
5.1 Caractéristiques .....	24
5.2 Connexion du RTCDS3231 sur la carte .....	26
5.3 Connexion du RTCDS3231 sur la carte et LCD.....	26
6. Les relais .....	26
7. Les touches de commande .....	27
8. Le bus I2C.....	28
9. BUZZER.....	29
10. Conclusion .....	30
<b>Chapitre III ; Réalisation de la carte</b> .....	31
1. Introduction .....	32
2. Algorithme d'optimisation .....	32
3. La Partie simulation .....	34
4. Organigramme de fonctionnement.....	35
4.1 Sous-programme de réglage date et heure.....	36
5. Expression de la puissance consommée par une pompe .....	38
5.1 Exemple de la consommation électrique pendant une journée.....	39
6. Réalisation pratique .....	39
7. Conclusion.....	42

**Conclusion générale**.....43

**Bibliographie**.....44

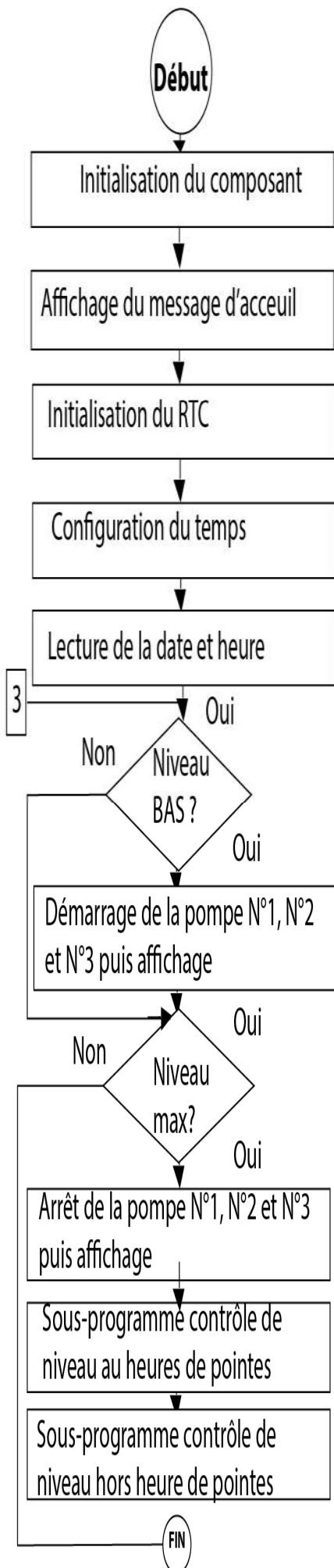


Figure 3.3 : Organigramme du programme principal

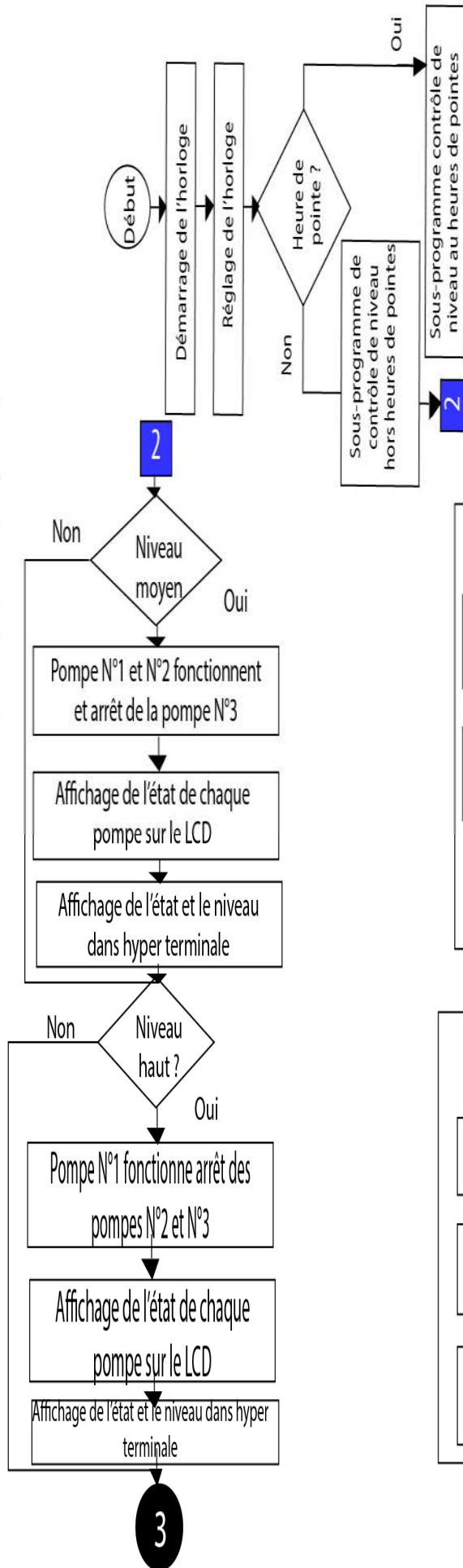


Figure 3.6 : Organigramme de Sous-programme contrôle de niveau ou d'heure de pointe

Figure 3.4 : Organigramme de Sous-programme contrôle d'heure de pointe

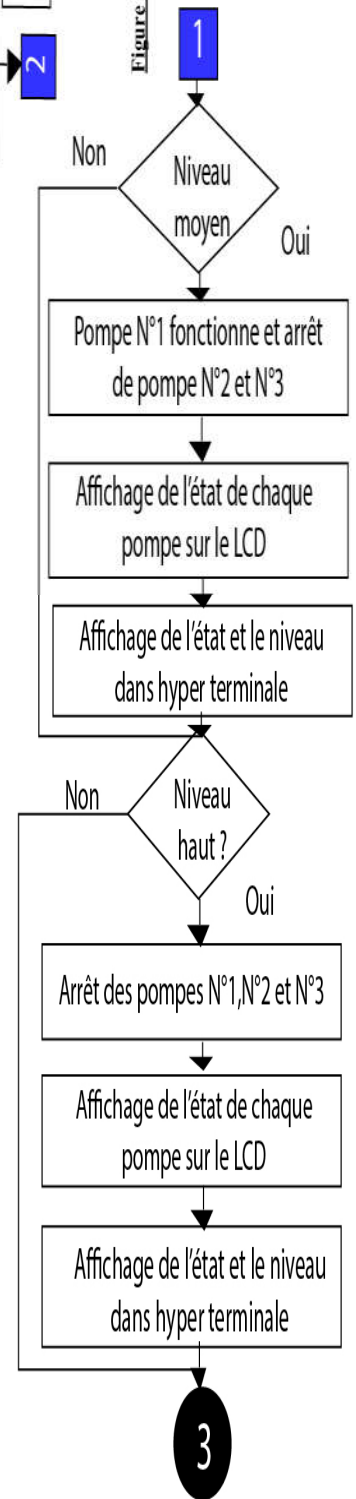


Figure 3.5 : Organigramme de Sous-programme contrôle de niveau ou d'heure de pointe

## Introduction générale

L'eau qu'on surnomme l'or bleu, ce liquide précieux, est indispensable à la vie, sans elle il n'y aurait aucune vie possible sur terre. Le constat est simple, l'homme a besoin d'eau pour son existence et pour l'accomplissement de l'ensemble de ses activités journalières telles que l'industrie ou l'agriculture, ainsi il est nécessaire d'imposer une bonne gestion de l'eau afin d'optimiser son exploitation. Par conséquent, il devient indispensable de faire appel à des techniques efficaces de supervision au niveau des installations hydrauliques et de leurs instruments.

Cependant, nous sommes face à un problème qui est le coût élevé de l'énergie (électricité) pendant les heures de pointe, c'est pour cela, il est préférable de procéder au pompage hors de cette période de temps.

En effet, notre but de cette réalisation, est de concevoir une carte qui permettra la gestion et l'optimisation du fonctionnement des stations de pompage.

L'idée de ce projet est de réaliser une carte de commande simple et surtout non coûteuse afin d'optimiser le fonctionnement des ouvrages d'extraction d'eau suivant les tarifs de SONELGAZ (Jour, Heures de Pointe et Nuit). Mais aussi notre carte permet le stockage des données de consommation durant toute une année. Ces données peuvent être transmises à un PC à l'aide d'un simple câble série (USB).

Maintenant que nous avons présenté l'idée générale de notre projet, nous allons présenter la démarche que nous avons suivie en indiquant le contenu des différentes parties qui constituent ce manuscrit.

Le premier chapitre consistera, pour un premier temps, à faire une présentation des stations de pompage utilisées à l'hydraulique

Dans le deuxième chapitre nous présentons le matériel utilisé.

Le dernier chapitre sera consacré à la simulation et la programmation ainsi que la réalisation pratique. Notre manuscrit sera conclu par une conclusion générale ainsi que des perspectives.

# *Chapitre I :*

## *« Présentation de la Station de pompage »*

### **1. Introduction ;**

Dans ce premier chapitre, nous allons donner un aperçu sur le principe de fonctionnement d'une station de pompage utilisée en hydraulique. Nous allons par la suite, présenter une étude sur la variation de la consommation d'eau pendant les saisons d'hiver et d'été ainsi que les heures et les tarifs d'énergie de l'électricité (Jour/Pointe/Nuit) qui diffèrent d'une période à une autre. Enfin, nous présenterons un algorithme qui explique le principe de fonctionnement de notre carte.

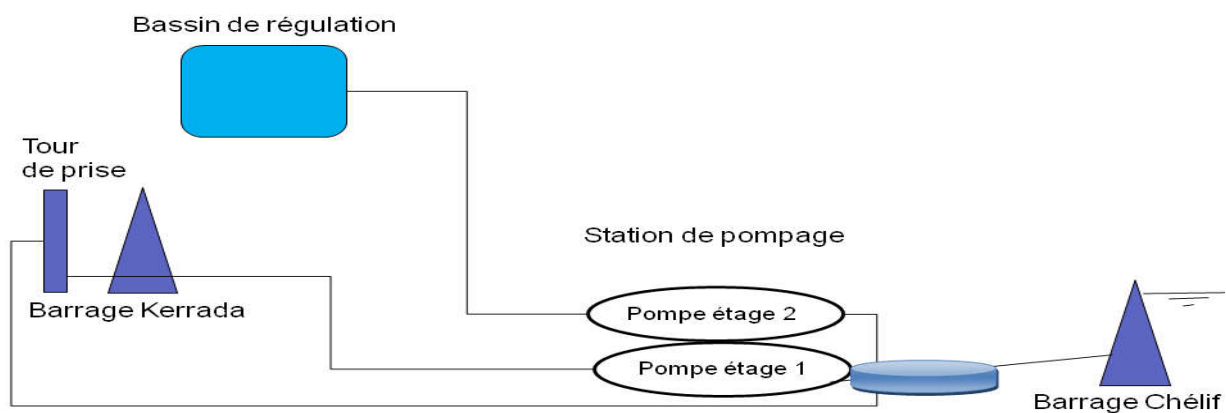
### **2. Station de pompage ;**

Une station de pompage est un ouvrage hydromécanique destiné à pomper l'eau potable

d'un niveau géographique bas à un niveau plus haut. On distingue dans une station de pompage les éléments suivants : le poste de transformation, le groupe électropompe, l'armoire de commande et de protection, le conduit de raccordement de protection, les réservoirs et les accessoires (vanne, clapets, ....) [1].

## 2.1 Les composantes principales du système de transfert

Le système de transfert d'eau à partir du barrage de Cheliff vers Mostaganem, Arzew et Oran (MAO) alimente Mostaganem et la wilaya d'Oran par un volume de près de 600 000 mètres cubes/jour. Il sécurise une région autrefois confrontée aux pénuries d'eau. La station de pompage est composée de deux étages de pompes et chaque étage se compose de 4 pompes. Le bassin de régulation est un réservoir de secours dont la fonction est de faire des appoints au circuit d'alimentation en cas de nécessité. Le barrage de kerrada c'est le réservoir de stockage d'eau [2].



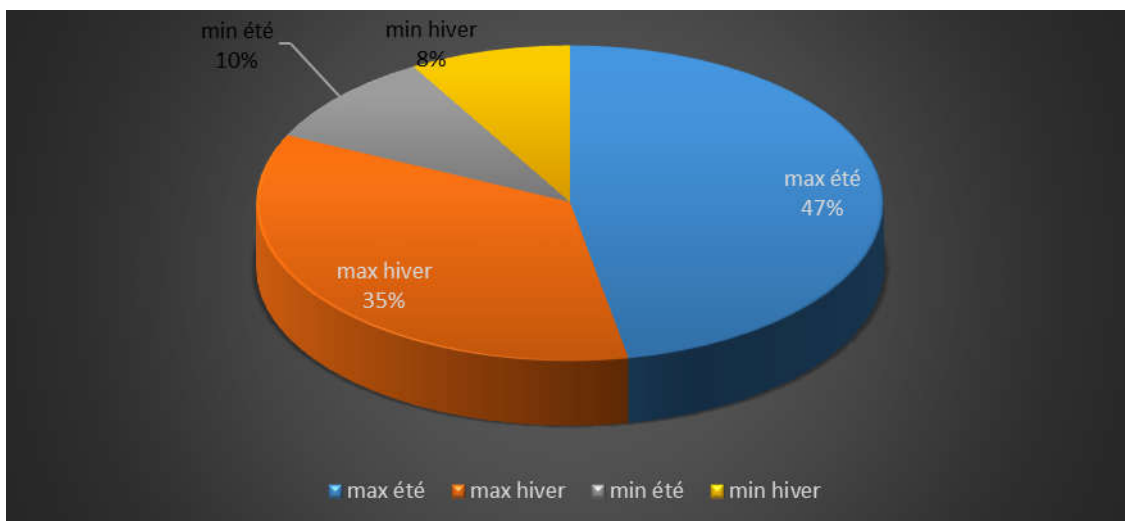
***Figure 1.1*** ; les composantes principales de système de transfert

## 3. Principe de fonctionnement d'une pompe :

Le principe essentiel de la pompe est de refouler une quantité d'eau d'un niveau bas à un niveau haut. C'est pour cela que la pompe se compose de deux parties essentielles qui sont : L'aspiration et le refoulement. Le fonctionnement des pompes diffère selon des critères bien définis tels que : le débit, la pression, la hauteur, le rendement et la puissance [3].

## 4. Variation de la consommation d'eau :

Dans ce graphe, nous présentons un document qui nous a été donné au niveau de la station de pompage de Mostaganem. Ce document présente des données de consommation d'eau enregistrées pendant les saisons d'hiver et d'été.



**Figure 1.2 :** Consommation d'eau pendant les saisons d'hiver et d'été

On remarque à partir de ce document qu'en saison d'été, on consomme beaucoup plus qu'en saison d'hiver, ce qui est tout à fait normal en raison de la chaleur qui sévit en été. On remarque aussi que la quantité d'eau consommée varie de façon considérable durant toute la journée.

### 5. Présentation des tarifs de la SONELGAZ :

Nous nous sommes aussi procurés auprès des services de la SONELGAZ, un deuxième document (présenté dans le tableau suivant) qui résume les tarifs de la consommation journalière pendant les saisons d'hiver et d'été.

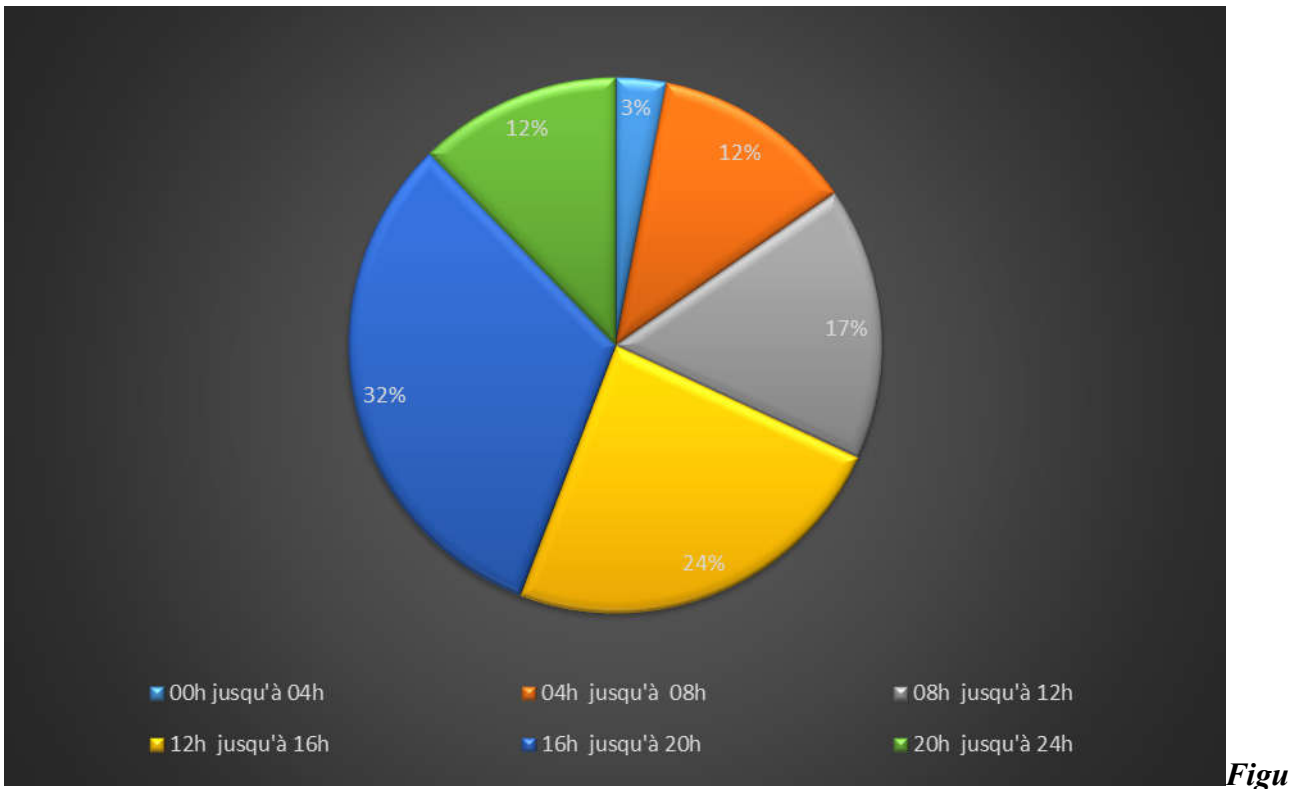
	JOUR	POINTE	NUIT	<b>Tab leau 1.1 :</b> Prés enta tion des hora ires et prix
Saison d'hiver	06.00 → 17.30	17.30 → 22.00	22.00 → 06.00	
Saison d'été	08.00 → 11.00 16.00 → 19.00	11.00 → 19.00 22.00 → 23.00	19.00 → 22.00 23.00 → 08.00	
Prix DA (KW/h)	180.64	872.02	180.64	

journaliers.

D'après ces données, nous remarquons que les heures de pointe diffèrent d'une saison à une autre et que le prix de l'énergie est clairement différent pendant les trois périodes de la journée.

### Exemple de consommation d'eau pendant une journée

La figure suivante nous montre la variation de la consommation d'eau pendant



**re 1.2** : consommation d'eau pendant une journée

D'après le graphique ci-dessus, on remarque que la consommation d'eau est élevée dans l'intervalle entre 08h jusqu'à 20h. Donc la consommation d'eau augmente pendant les heures de pointe quand le prix est cher. Notre but dans ce projet, consiste alors à optimiser le fonctionnement des pompes suivant les tarifs de la SONELGAZ, c'est-à-dire qu'on veut commander le démarrage et l'arrêt des moteurs hors des heures de pointe.

## 6. Conclusion

Dans ce premier chapitre, nous avons présenté des documents qui nous ont montré que la consommation d'eau diffère considérablement d'une saison à une autre et diffère aussi pendant la même journée. Sachant que le coût de la consommation énergétique est lui aussi nettement différent selon les périodes, nous avons choisi d'optimiser le fonctionnement des pompes en commandant leurs heures de démarrage et d'arrêt.

Pour réaliser cet objectif et répondre à tous ces problèmes, nous allons réaliser une carte électronique à base d'un ARDUINO UNO dont le programme de commande sera basé sur l'algorithme de fonctionnement qui a été établi dans ce chapitre.

# **Chapitre II :**

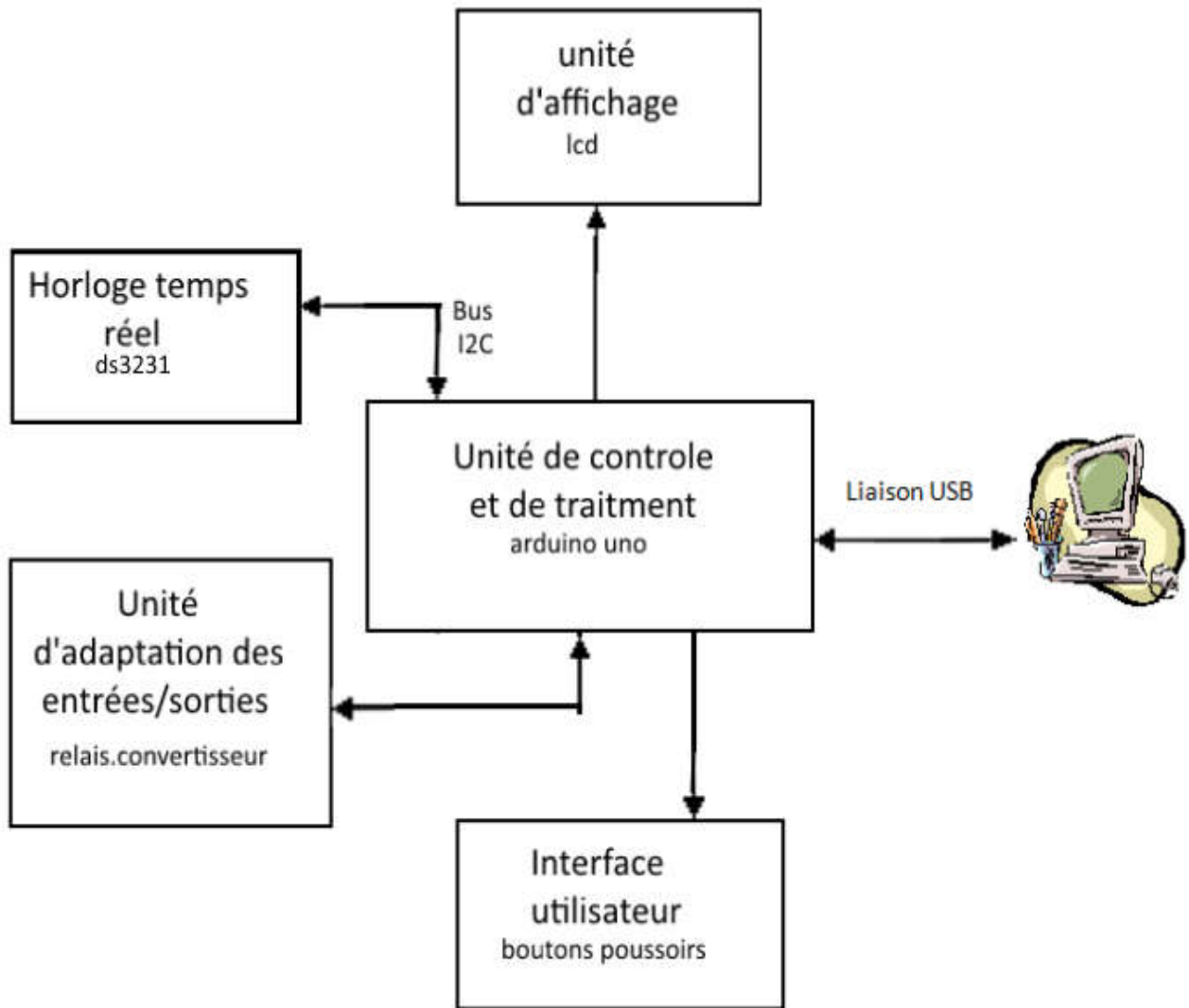
## **« Composants de la carte »**

### **1. Introduction**

Dans ce chapitre, nous présentons une description précise de la solution choisie pour répondre aux spécifications de notre cahier de charge en abordant la conception précise de chaque partie du système afin d'obtenir une schématisation complète et précise.

Nous allons définir l'unité de contrôle et sa structure. La partie la plus importante sera celle qui traite les différents ports d'entrées/sorties, ainsi que le convertisseur qui jouera le rôle d'interface entre l'unité de contrôle et le capteur de niveau.

Notre schéma synoptique est le suivant ;



**Figure 2.1:** Schéma synoptique de la carte.

Notre carte se compose de plusieurs unités qui sont :

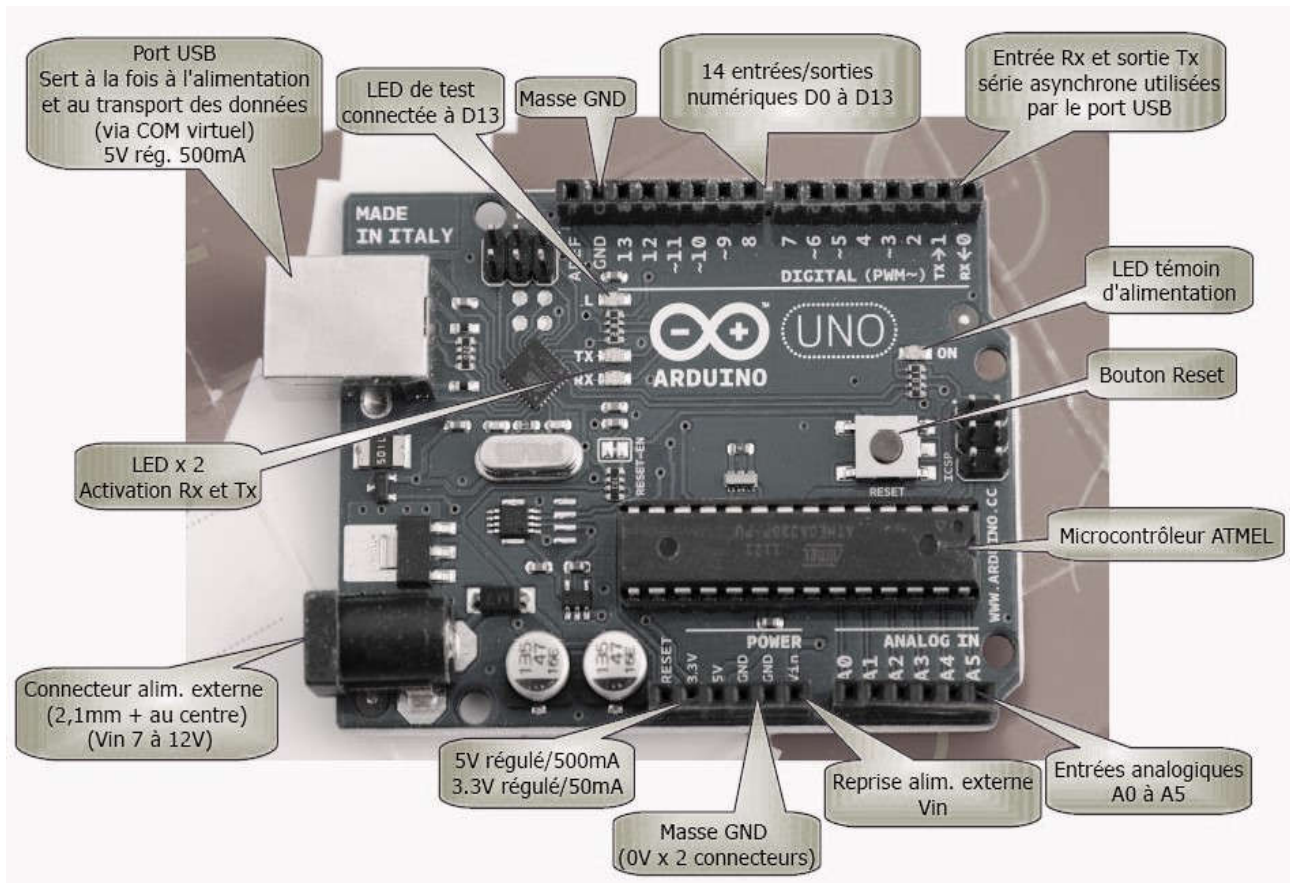
- ❖ Un afficheur LCD qui est lié directement à la carte pour afficher les données.
- ❖ Un clavier composé de trois touches qui commande le réglage de temps ainsi que la date.
- ❖ Une horloge temps réel.
- ❖ L'unité de traitement et de contrôle des données qui est arduino uno.
- ❖ Un buzzer qui joue le rôle d'une alarme.
- ❖ Des actionneurs qui assurent la commande des deux pompes (relais).

## 2. La carte ARDUINO UNO

Arduino est le nom d'un « fabricant » de circuits imprimés sur lesquels il est possible de brancher toutes sortes d'appareils. Cette carte se programme sur l'ordinateur via un câble USB (ou autre) et permet de diriger n'importe quel appareil. Pour cela, il suffit de modifier le code qu'exécute l'arduino. La particularité de ce système c'est que les plans des cartes sont disponibles gratuitement, et qu'il est possible de modifier et de réutiliser ces plans. C'est pourquoi de nombreux systèmes basés sur Arduino (freeduino, sparkfun, etc.) existent. Dans la figure suivante la structure de la carte arduino est expliquée en détail.



**Figure 2.2** : la carte de l'arduino uno.



**Figure 2.3** ; schéma électrique de la carte Arduino

Arduino simplifie la façon de travailler avec les microcontrôleurs, tout en offrant plusieurs avantages pour l'utilisateur:

- Les cartes Arduino sont relativement peu coûteuses comparant aux autres plateformes.
- Le logiciel Arduino, écrit en Java, tourne sous les systèmes d'exploitation Windows, Macintosh et Linux.
- L'environnement de programmation Arduino (le logiciel Arduino) est souple à utiliser pour les débutants.
- Le logiciel et le langage Arduino sont publiés sous licence open source, disponible pour être complété par des programmeurs expérimentés.
- Les cartes Arduino sont basées sur les microcontrôleurs Atmel ATMEGA8, ATMEGA168, ATMEGA 328, etc.

## 2.1. Caractéristiques techniques :

Microcontrôleur	ATmega328P
Tension de fonctionnement	5V
Tension d'entrée (recommandé)	7-12V
Tension d'entrée (limite)	6-20V
E / S numériques Pins	14 (dont 6 fournissent la sortie PWM*)
PWM numérique E / S Pins	6
Pins d'entrée analogique	6
DC Courant par I O Pin /	20 mA
Courant DC pour 3.3V Pin	50 mA
Mémoire flash	32 KB (ATmega328P) dont 0,5 KB utilisé par boot loader**
SRAM	2 KB (ATmega328P)
EEPROM ***	1 KB (ATmega328P)
Vitesse de l'horloge	16 MHz
Longueur	68,6 mm
Largeur	53,4 mm
Poids	25 g

## 2.2 Entrées et sorties

Chacune des 14 broches numériques sur la carte Uno peut être utilisée comme une entrée ou une sortie, en utilisant les fonctions `pinMode`, `digitalWrite`, et `digitalRead`. Elles fonctionnent à 5 volts. Chaque broche peut fournir ou recevoir 20 mA en état de fonctionnement recommandée et a une résistance de pull-up interne (déconnecté par défaut) de 20-50 k  $\Omega$ . Un maximum de 40mA est la valeur qui ne doit pas être dépassée sur toutes les broches d'Entrée/Sorties pour éviter des dommages continus du microcontrôleur.

Certaines broches ont des fonctions spécialisées:

- Série: 0 (RX) et 1 (TX). Permet de recevoir (RX) et transmettre (TX) TTL données série. Ces pins sont connectés aux pins correspondants de l'USB-TTL puce Serial ATmega8U2.
- LED: 13. Elle est équipée d'un conduit par la broche numérique 13. LED. Lorsque la broche est à la valeur HIGH, la LED est allumée, lorsque la broche est à la valeur faible, elle est éteinte.

L'UNO dispose de 6 entrées analogiques, A0 à A5, dont chacune fournit 10 bits de résolution (ou 1024 valeurs différentes). Par défaut, la tension est de 5 volts. Il est pourtant possible de changer la limite supérieure de la gamme en utilisant la broche AREF et la fonction analogique. Autres broches de la carte :

- AREF. Tension de référence pour les entrées analogiques. Pin utilisé avec analog.
- Réinitialiser.

## 2.3 Communication

Arduino a un certain nombre de moyens pour communiquer avec un ordinateur, une autre carte Arduino, ou autres microcontrôleurs. L'ATmega328 fournit UART TTL (5V) en communication série, disponible sur les broches numériques 0 (RX) et 1 (TX).

\* **PWM** En numérique, les signaux ont toujours une valeur LOW ou HIGH (0 ou 1). Le principe est de construire un signal alternativement LOW ou HIGH et de répéter cette alternance. La DEL est donc alternativement allumée et éteinte mais le cycle est tellement rapide que la persistance rétinienne nous donne l'illusion d'une DEL allumée en permanence.

\*\* **Bootloader** : il s'agit de la racine du système, à la manière du bios sous Windows, où l'on peut installer un recovery alternatif, un kernel, une image d'Android, parmi d'autres choses... après l'avoir ouvert.

\*\*\* **EEPROM** : mémoire dont les valeurs sont conservées lorsque la carte est éteinte (comme un petit disque dur).

## 3. Afficheur à cristaux liquides

*Les afficheurs à cristaux liquides LCD (Liquide Crystal Display), sont des modules compacts intelligents et nécessitent peu de composants externes pour un bon fonctionnement.*

*Ils consomment relativement peu de 1 à 5 mA, les afficheurs sont caractérisés par le nombre de caractères affichables et par le nombre de lignes disponibles. Dans notre exemple l'afficheur possède seize caractères sur deux lignes.*



**Figure 2.4** ; afficheur LCD 4\*20

L'afficheur LCD se compose de 16 broches (tableau 2.1)

Les broches	Nom	Rôle
1	VSS	Masse
2	VDD	+5V
3	VO	Réglage du contraste
4	RS	Sélection du registre (commande ou donnée)
5	R/W	Lecture ou écriture
6	E	Entrée de validation
7 à 14	D0 à D7	Bits de données
15	A	Anode du rétroéclairage (+5V)
16	K	Cathode du rétroéclairage (masse)

**Tableau 2.1** Caractéristiques de l'afficheur LCD

#### 4. Connexion de l'afficheur sur la carte :

Dans notre application, nous avons utilisé un écran LCD alphanumérique de 4 lignes et de 24 caractères. L'afficheur LCD utilise 6 à 10 broches de données ((D0 à D7) ou (D4 à D7) + RS + E) et deux d'alimentations (+5V et masse). La plupart des écrans possèdent aussi une entrée analogique pour régler le contraste des caractères. Nous brancherons dessus un potentiomètre de 10 kOhms.

Les numéros des broches NOM Description

- 1 VSS Masse
- 2 VDD Alimentation positive (V+)
- 3 VO Contraste réglable par potentiomètre
- 4 RS Sélection de registre (0= instruction; 1 = donnée)
- 5 R/W ou RD Lecture ou écriture (1= lecture; 0=écriture)

6 E Enable (validation, actif au niveau haut)

7 D0 Bit 0 du bus de données

8 D1 Bit 1 du bus de données

9 D2 Bit 2 du bus de données

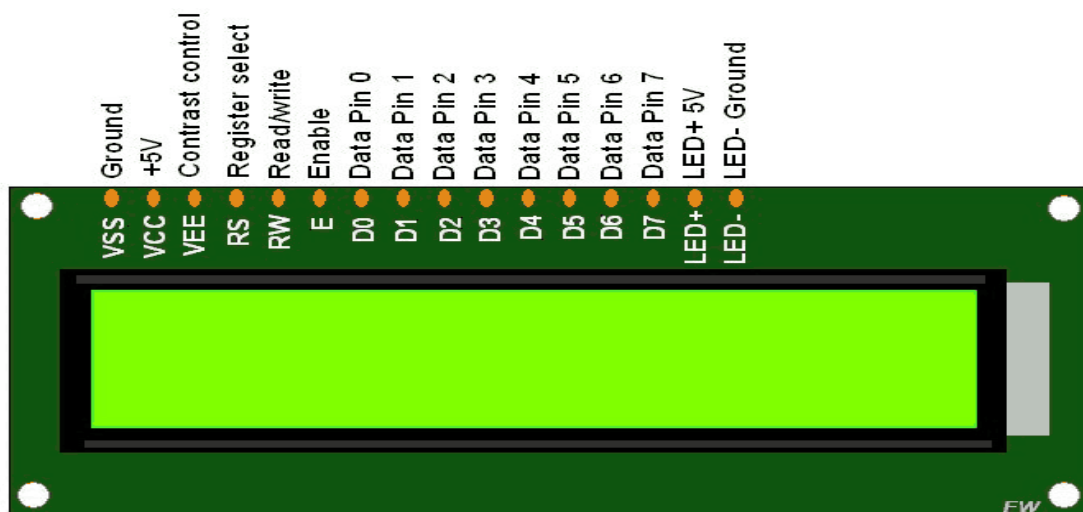
10 D3 Bit 3 du bus de données

11 D4 Bit 4 du bus de données

12 D5 Bit 5 du bus de données

13 D6 Bit 6 du bus de données

14 D7 Bit 7 du bus de données

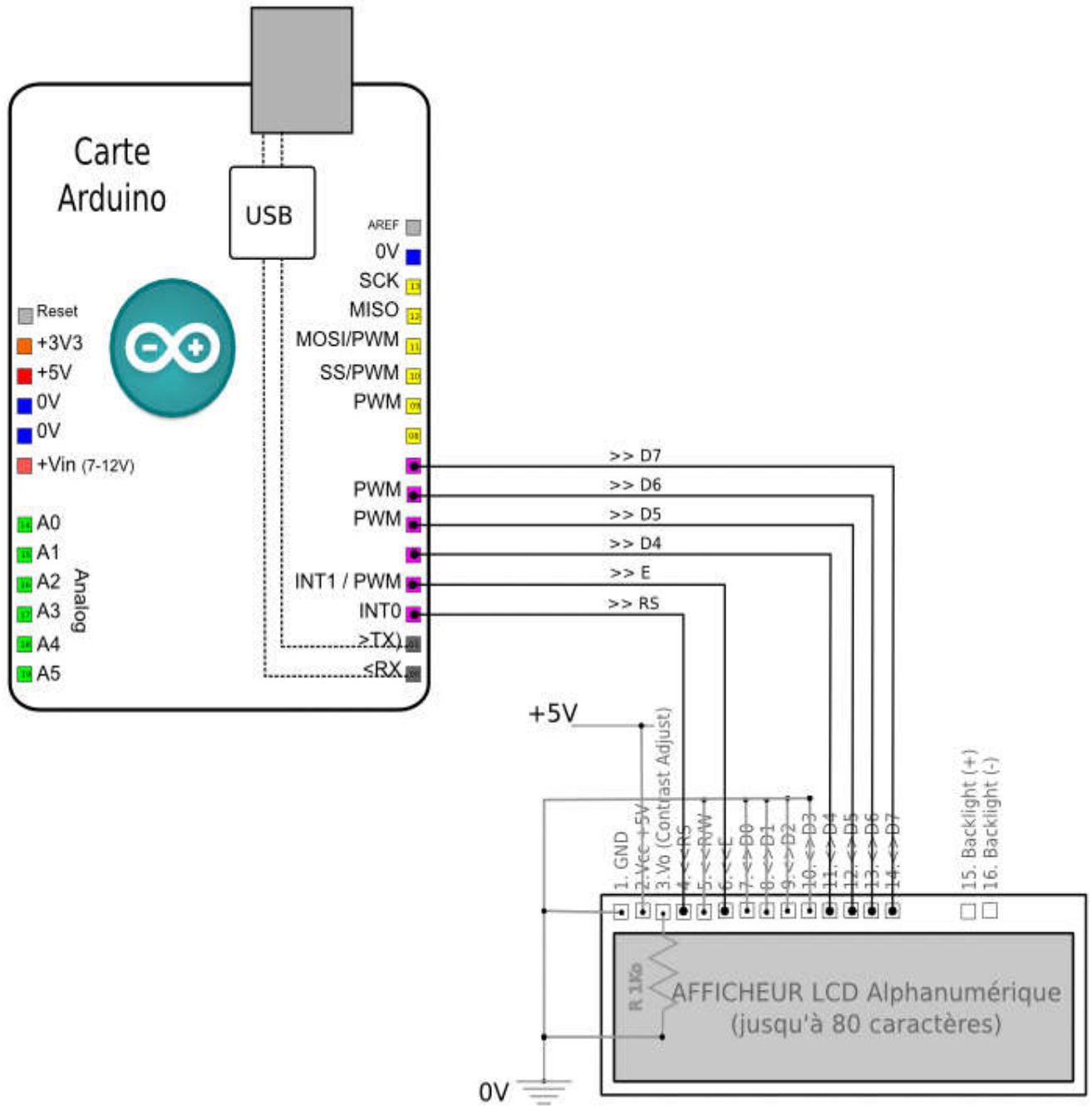


**Figure 2.5** ; les broches de l'afficheur LCD

Les 10 broches de données peuvent être placées sur n'importe quelles entrées/sorties numériques de

de

l'Arduino.



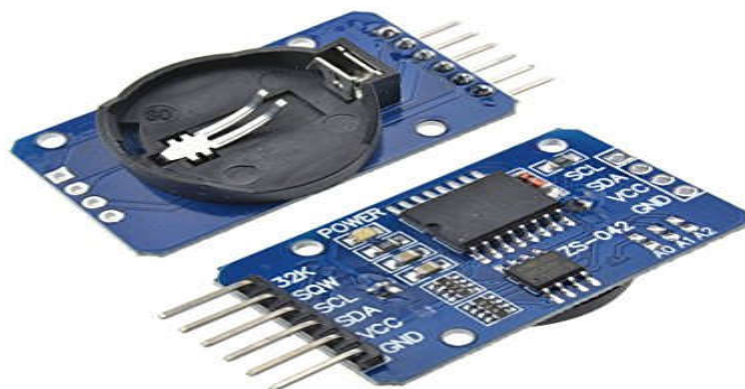
**Figure 2.6** ; Montage de l'afficheur LCD sur l'ARDUINO

## 5. Horloge temps réel :

Nous avons utilisé un RTC DS3231 (real clock time) qui servira d'horloge temps réel. Le **module horloge DS3231** est une horloge en temps réel I2C (RTC) moins coûteuse, très précise, avec un oscillateur à cristaux liquides (TCXO) intégré et un cristal. L'appareil intègre une entrée de batterie, débranche l'alimentation principale et maintient un chronométrage précis. L'oscillateur intégré améliore la précision à long terme de l'appareil et réduit le nombre de composants de la ligne de production. Le **RTC** conserve les secondes, les minutes, les heures, le jour, la date, le mois et l'année. Moins de 31 jours du mois, la date de fin sera automatiquement ajustée, y compris les corrections pour l'année bissextile. L'horloge fonctionne dans les 24 heures ou l'indication de bande / AM / PM du format de 12 heures. Il fournit deux réveils configurables et un calendrier peut être réglé sur une sortie d'onde carrée. L'adresse et les données sont transférées en série via un bus bidirectionnel I2C. Un circuit de référence et de comparaison de tension à compensation de température de précision surveille l'état de VCC pour détecter les pannes de courant, fournir une sortie de réinitialisation et, si nécessaire, passer automatiquement à l'alimentation de secours.

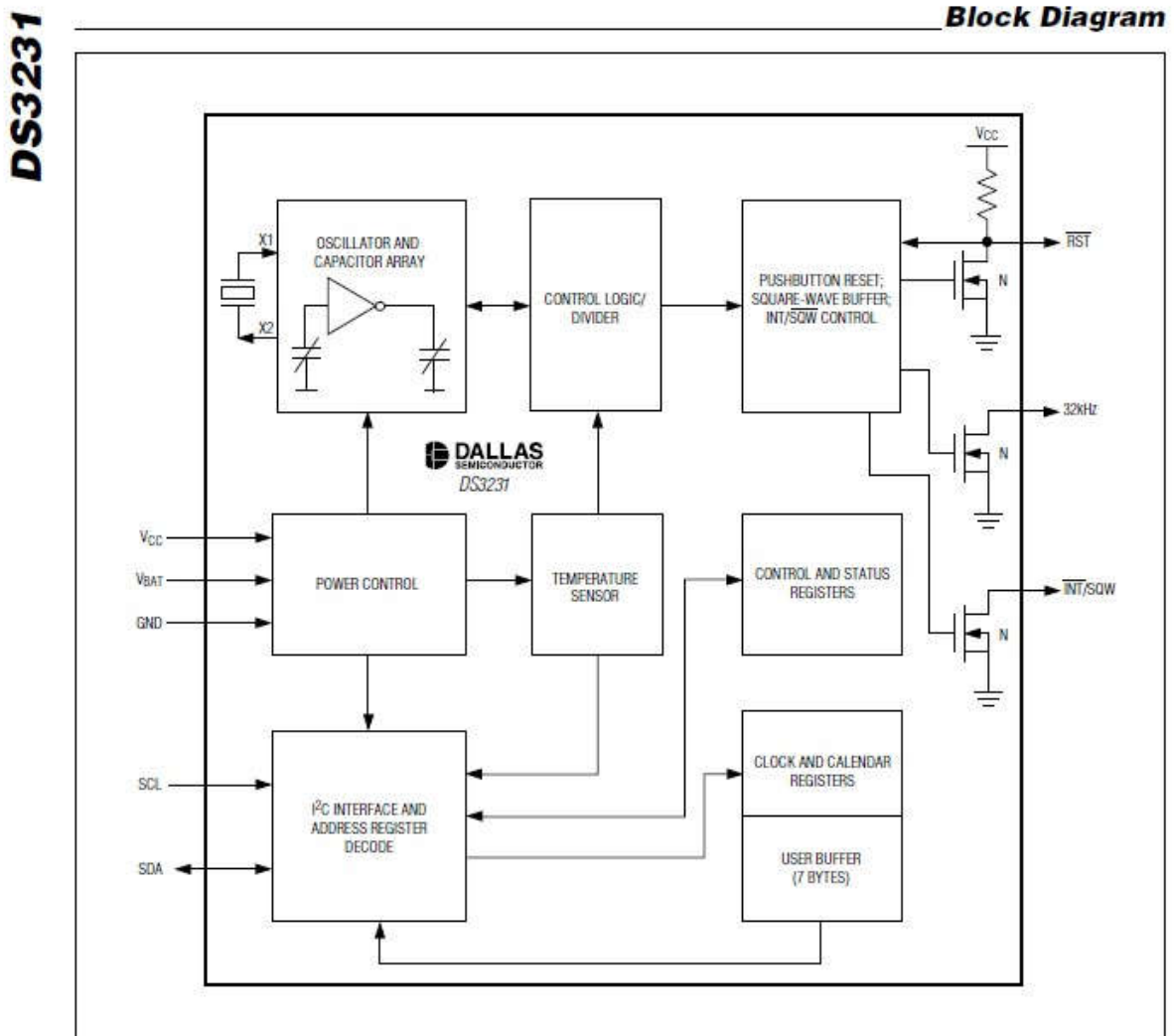
### 5.1 Caractéristiques

- Puce d'horloge : haute précision ds3231.
- Précision de L'horloge : plage de 0 à 40 ° C, précision 2ppm, erreur d'environ 1 minute.
- Batterie : CR2032.
- Tension de fonctionnement : 3.5 – 5.5V.
- Générateur d'horloge en temps réel, secondes, minutes, heures, jour, date, mois, année et calendrier.
- calendrier réveil avec deux sorties d'onde carrée programmable.
- Mémoire puces : AT24C32 (capacité de stockage 32 K).
- Dimensions : 38 x 22 x 14mm.
- Poids : 8g.



**Figure 2.7** ; Montage RTC DS3231

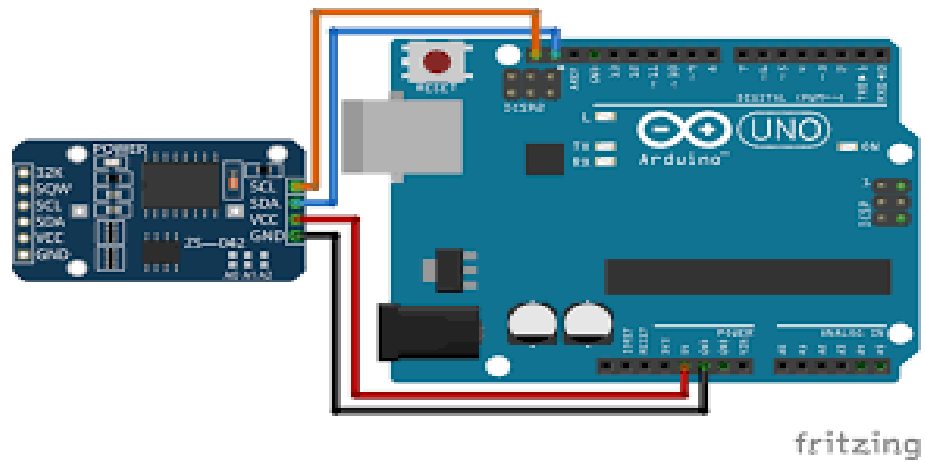
La figure 2.8 représente le montage interne du RTC DS3231.



**Figure 2.8** ; Montage interne du RTC DS3231

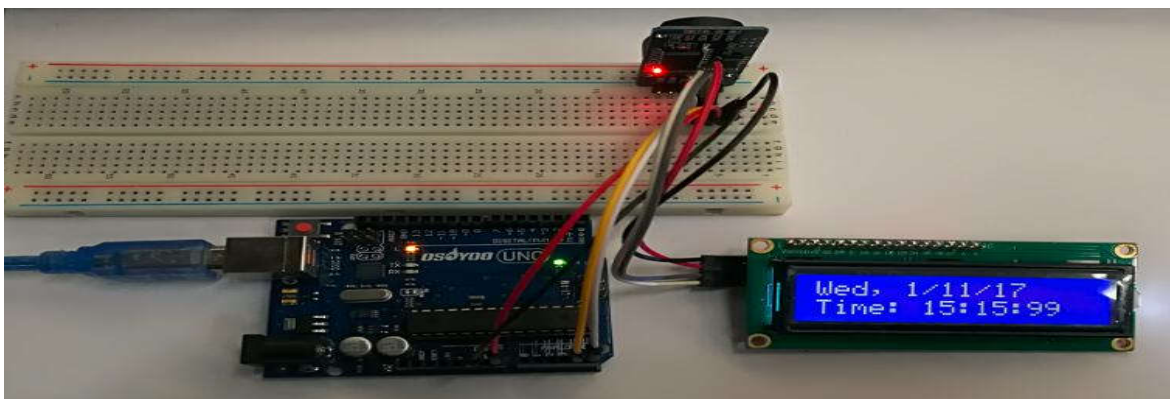
## 5.2 Connexion du RTC DS3231 sur la carte :

Dans la figure 2.9, nous représentons la connexion de l'horloge avec l'arduino



**Figure 2.9** ; connexion de l'horloge avec l'arduino

### 5.3 Connexion du RTC DS3231 sur la carte et l'afficheur LCD :



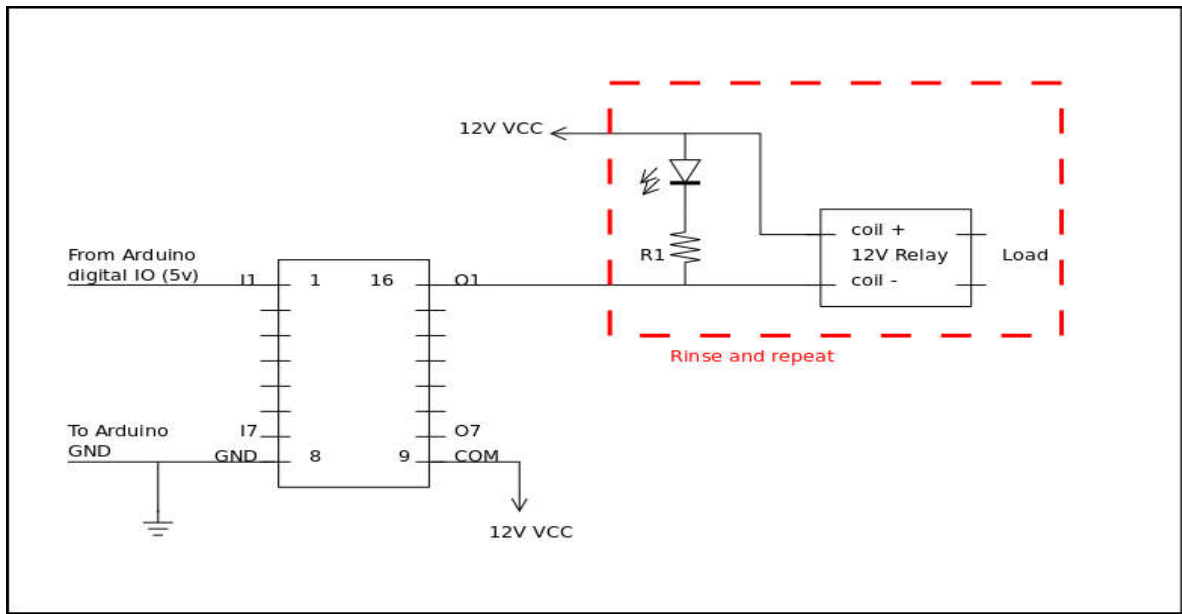
**Figure 2.10** ; horloge avec l'arduino et LCD

## 6. Les relais :

Un relais est un appareil dans lequel un phénomène électrique (courant ou tension) qui contrôle la communication ON/OFF d'un élément mécanique ou d'un élément électrique. C'est en quelque sorte un interrupteur que l'on peut actionner à distance, et /ou la fonction de coupure est dissociée de la fonction de commande. Parmi ses avantages, on peut citer ;

- ✓ Capacité de commuter aussi bien des signaux continus qu'alternatifs sur une large gamme de fréquences.
- ✓ Fonctionnement avec une dynamique considérable du signal commuté.

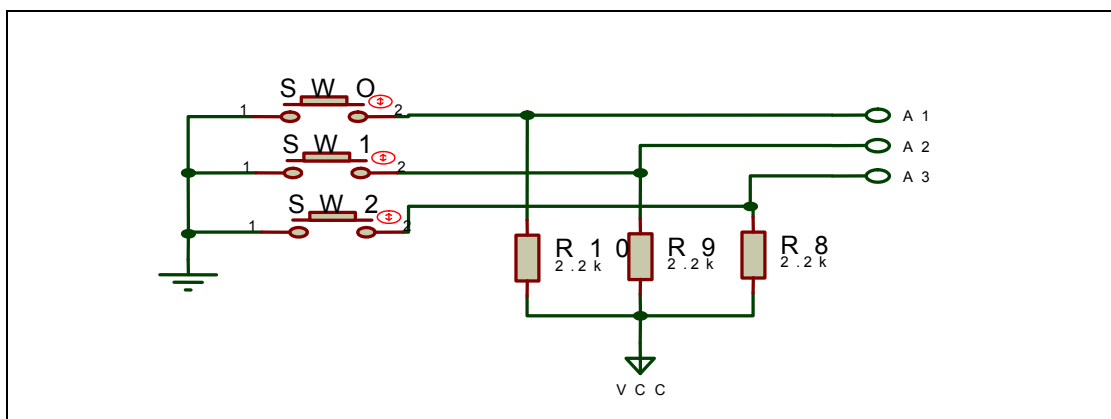
- ✓ Aucun ajout de bruit ou de distorsion.



**Figure 2.11** ; Montage du relais

## 7. Les touches de commande :

La carte dispose de trois boutons poussoirs ( $SW_0$ ,  $SW_1$  et  $SW_2$ ) qui sont reliés à trois entrées analogique de l'ARDUINO (AN1, AN2 et AN3) et reliés aux résistances.



**Figure 2.12** ; Montage des boutons poussoirs

## 8. Le bus I2C

Le bus I2C permet de faire communiquer entre eux des composants électroniques variés grâce à seulement trois fils :

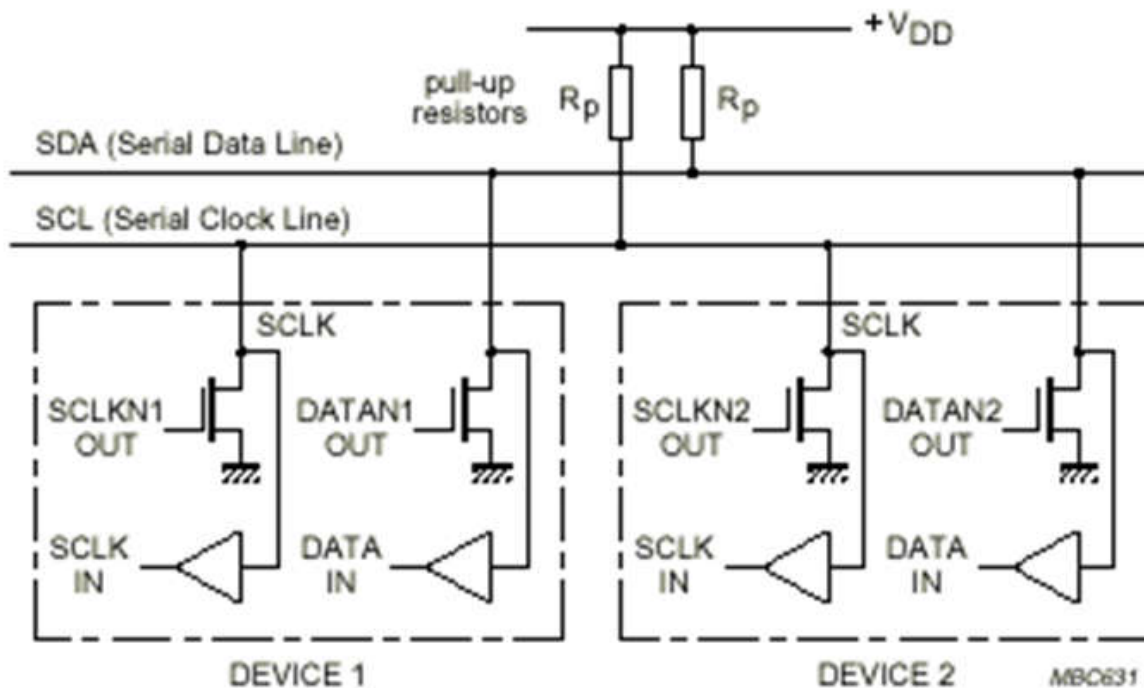
- Un signal de donnée (SDA),

- Un signal d'horloge (SCL),
- Un signal de référence électrique (Masse).

Les informations transitent sur un même fil mais jamais en même temps. Toutes les transmissions (écriture ou lecture) sont initiées par le maître et adressées à un esclave via son adresse. C'est un bus lent, à l'origine la fréquence de transmission était fixée à 100 kbits/s (mode standard). Depuis d'autres fréquences de fonctionnement sont apparues 400 kbits/s (mode rapide), 1 Mbits/s (mode rapide +), 3.4 Mbits/s (mode rapide élevé) et 5 Mbits/s (mode ultra rapide).

De nombreux fabricants ayant adopté le système, la variété des circuits disponibles disposant d'un port I2C est énorme : Ports d'E/S bidirectionnels, Convertisseurs A/N et N/A, mémoires (RAM, EPROM, EEPROM, etc.), Circuits Audio (Égaliseur, Contrôle de volume, ...) et autres drivers (LED, LCD, ...).

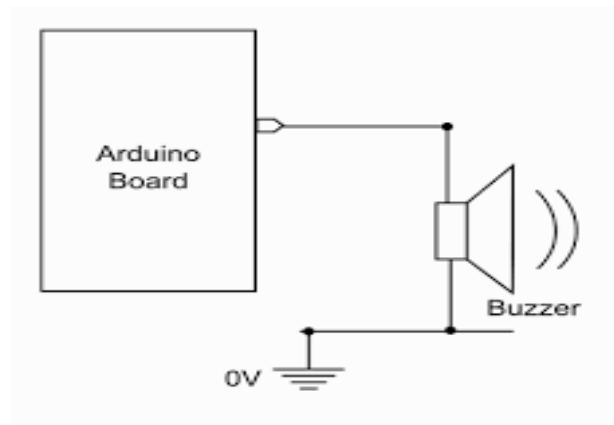
Le nombre de composants qu'il est ainsi possible de relier est essentiellement limité par la charge capacitive des lignes SDA et SCL : 400 pF.



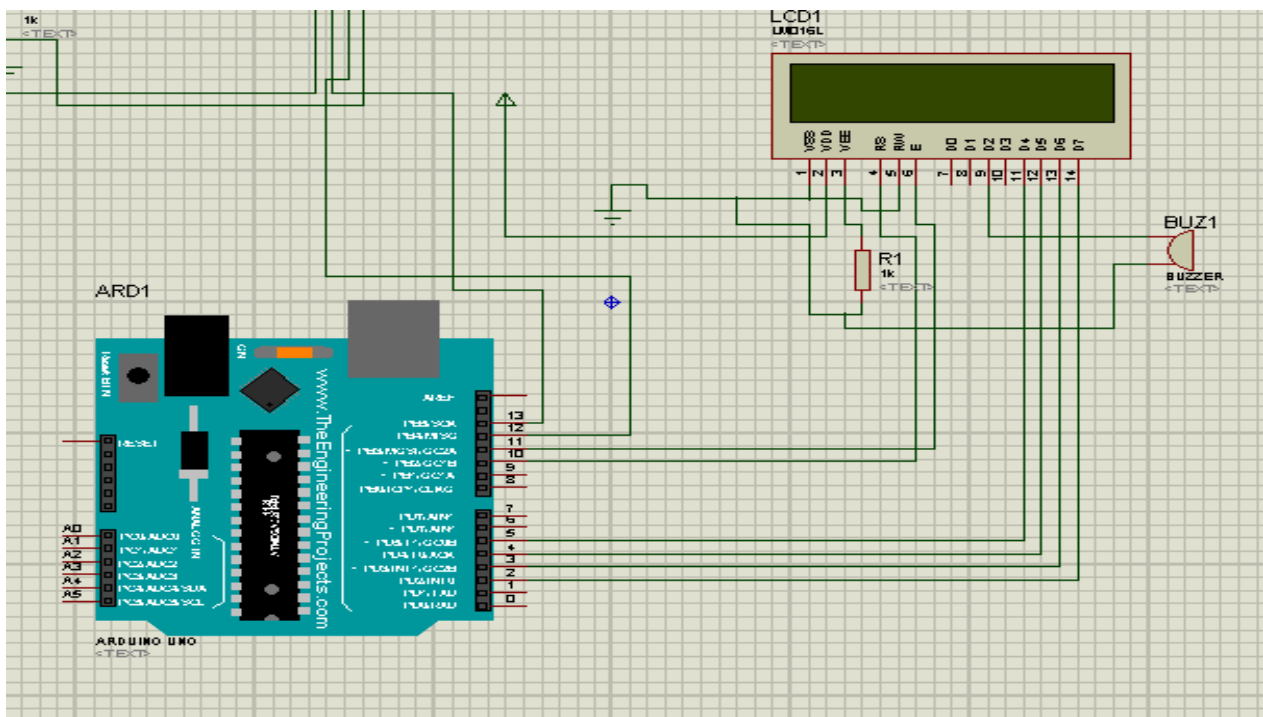
**Figure 2.13;** architecture du bus I2C

## 8. BUZZER :

Il s'agit d'un petit haut-parleur qui est relié à la broche I/O 38 et sera très appréciable pour orner les programmes de confirmations sonores. La tâche de programmation des signaux d'avertissement sonore est très facile. Dans la figure suivante, nous présentons le montage buzzer et la figure 2.15, représente l'interconnexion entre arduino buzzer et écran LCD



**Figure 2.14** ; Montage du BUZZER



**Figure 2.15** ; L'interconnexion entre arduino, buzzer et écran LCD

## 9. Conclusion

Dans la partie conception nous nous sommes basés sur la simplicité des montages avec le moindre coût, tout en assurant le bon fonctionnement de notre carte. Pour réduire le nombre de connexions et de circuits nous avons utilisé trois boutons poussoirs au lieu d'un clavier matriciel. Dans ce chapitre, nous avons présenté brièvement ARDUINO UNO, et nous remarquons qu'avec sa simplicité nous pouvons l'utiliser comme une unité de contrôle pour notre système.

**Chapitre III ;**  
**« Réalisation de la carte »**

Dans ce chapitre seront développées les différentes étapes suivies depuis la mise sur pied du schéma électrique jusqu'à l'assemblage et le test final du système. Deux parties seront étudiées dans ce chapitre ; une partie de simulation puis une partie qui sera consacrée à l'étude pratique de la carte de commande de la station de pompage.

## 2. Algorithme d'optimisation :

Comme nous l'avons déjà mentionné en introduction générale, notre but est de concevoir une carte qui permet la gestion et l'optimisation du fonctionnement des stations de pompage. L'idée consiste dans la réalisation d'une carte de commande simple et notamment non coûteuse afin d'optimiser le fonctionnement des ouvrages d'extraction d'eau suivant les tarifs de SONELGAZ (Jour, Pointe et Nuit). En plus notre carte permet le stockage des données de consommation durant toute une l'année. Nous allons développer un algorithme qui satisfait ces conditions. Tout d'abord, nous allons définir quelques notions :

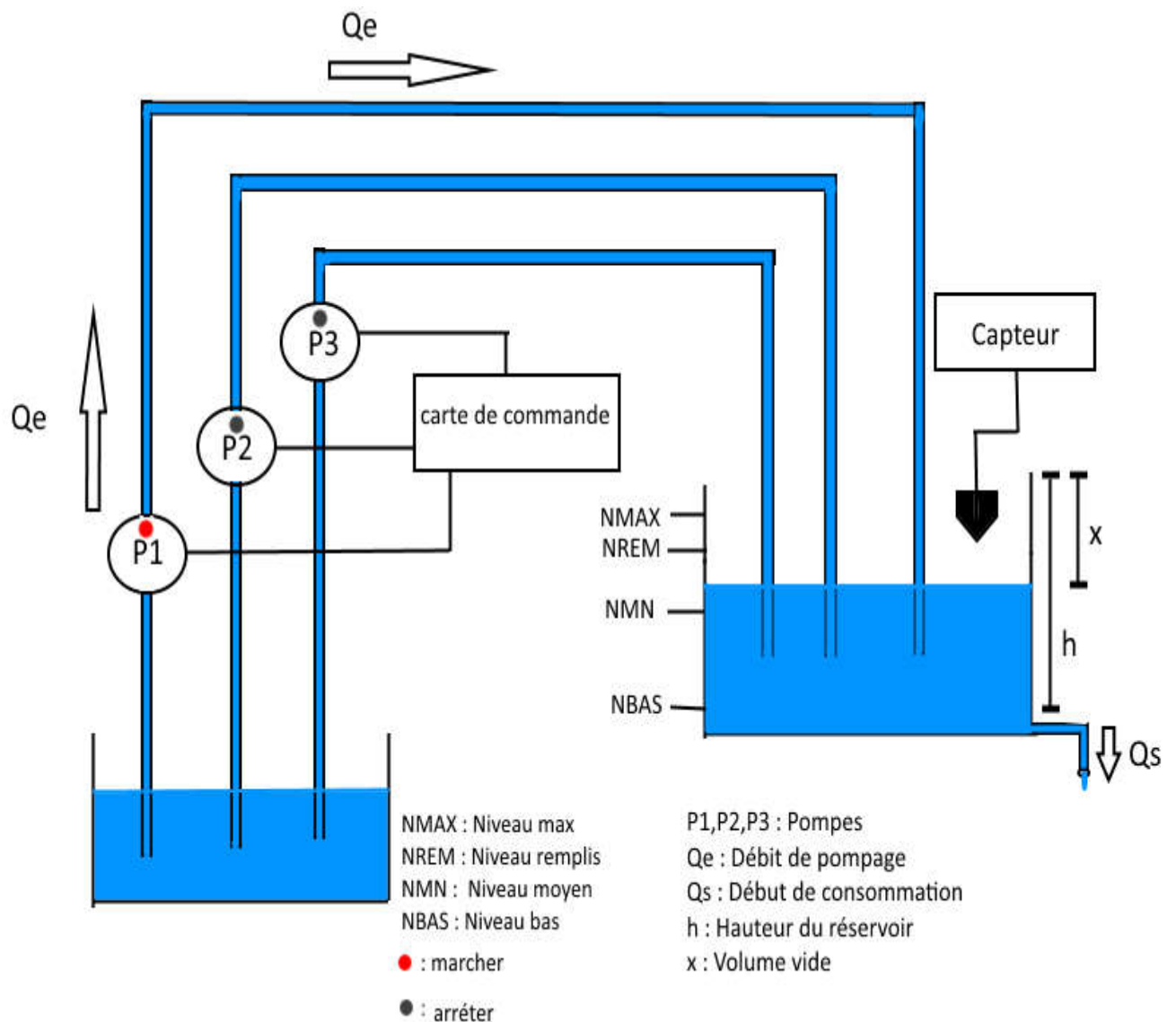
- Nb : Représente le niveau bas.
- Nm : Représente le niveau moyen.
- NMAX : Représente le niveau maximal
- NRM : Représente le niveau rempli (entre 60% et 80% du réservoir).
- Nbp : Représente le niveau bas pendant l'heure de pointe.
- Nmp : Représente le niveau moyen pendant l'heure de pointe.
- RMp : Représente le niveau rempli (entre 60% et 80% du réservoir) pendant l'heure de pointe.
- P1, P2 et P3 : Représente les 3 pompes.

### Les conditions :

- ◆ Si Nb=1 alors P1=1 & P2=1 et P3=1 ;  
Si Nb=0 on vérifie les heures de pointe. Si on est dans la période des heures de pointe ;
- ◆ Si Nmp=1 alors P1=1 et P2=0 & P3=0 ;
- ◆ Si RMp=1 alors P1=0 & P2=0 et P3=0 ;
- ◆ Si NMAX alors P1=0 & P2=0 et P3=0 ;  
Si on est dans la période hors des heures de pointe
- ◆ Si Nm=1 alors P1=1 et P2=1 & P3=0 ;

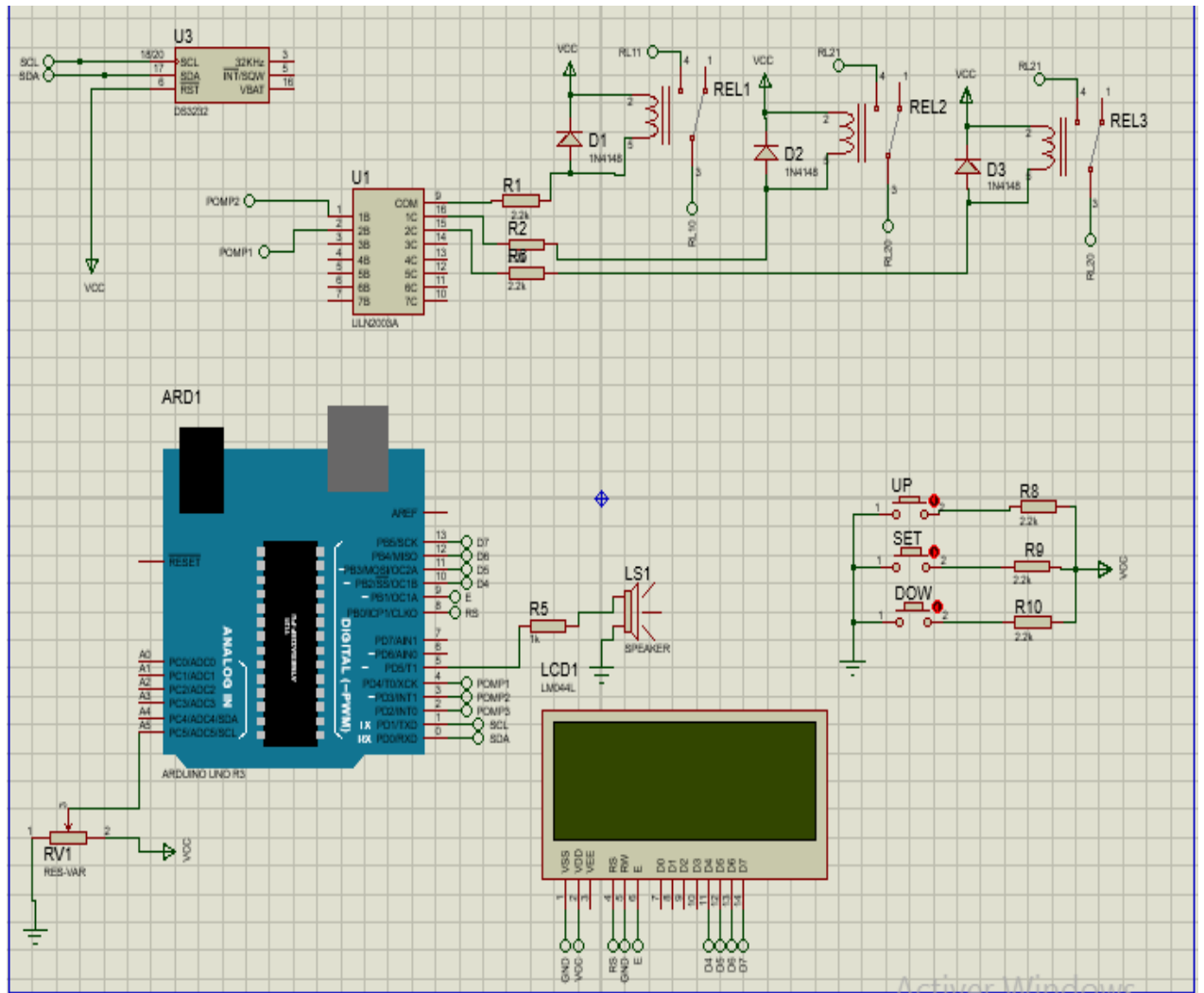
- ◆ Si  $R_{Mp}=1$  alors  $P1=1$  &  $P2=0$  et  $P3=0$  ;
- ◆ Si  $N_{MAX}$  alors  $P1=0$  &  $P2=0$  et  $P3=0$  ;
- ◆ Si le niveau d'eau atteint son niveau bas, alors les trois motopompes doivent forcément fonctionner en même temps pour répondre aux besoins des consommateurs.
- ◆ Si le niveau moyen est atteint, on vérifie si on est hors des heures de pointe, si oui les 2 motopompes (1 et 2) fonctionnent.
- ◆ Si le niveau rempli est atteint et qu'on est hors des heures de pointe, seule la motopompe 1 fonctionne.
- ◆ Si le niveau haut est atteint, c'est-à-dire que le réservoir est plein, alors il n'y aura pas de pompage.

L'algorithme présenté ci-dessus illustre le fonctionnement du système présenté sur la figure suivante :



**Figure 3.1** : schéma de fonctionnement.

### 3. La Partie simulation



**Figure 3.2 ;** schéma expliquant la simulation de la carte sur Isis Proteus

En revenant au schéma synoptique de notre carte, nous avons élaboré le schéma électrique de notre circuit comme illustré dans la figure précédente. La simulation a été faite à l'aide du logiciel ISIS PROFESSIONNEL. Pour simuler notre circuit, on doit implémenter le programme dans ARDUINO

### 4. Organigramme de fonctionnement

Pour que notre projet soit fonctionnel, nous avons créé le programme suivant (voir figure 3.3). En suite nous avons développé les trois sous-programmes :

1. Sous-programme de réglage date et heure
2. Sous-programme de control de niveau hors heures de pointe.

3. Sous-programme de control de niveau aux heures de pointe.

#### 4.1 Sous-programme de réglage date et heure

Ce programme va tester si nous sommes dans la période de pointe ou non. Dans les deux cas on va tester les niveaux mais les actions sont différentes.

#### 5. Expression de la puissance consommée par une pompe

L'expression de la puissance consommée par une pompe pendant 1 heure est donnée par la relation suivante ;

$$P = \frac{Q \cdot h}{n \cdot 367} \quad (01)$$

P : la puissance (KW)

Q : quantité d'eau /heure (m<sup>3</sup>/h)

h : hauteur. Nous avons pris la valeur de 100 m comme exemple.

n : rendement donné par le constructeur de la pompe. Nous avons pris la valeur de 0.5.

$$367 = 3600s / 9.81$$

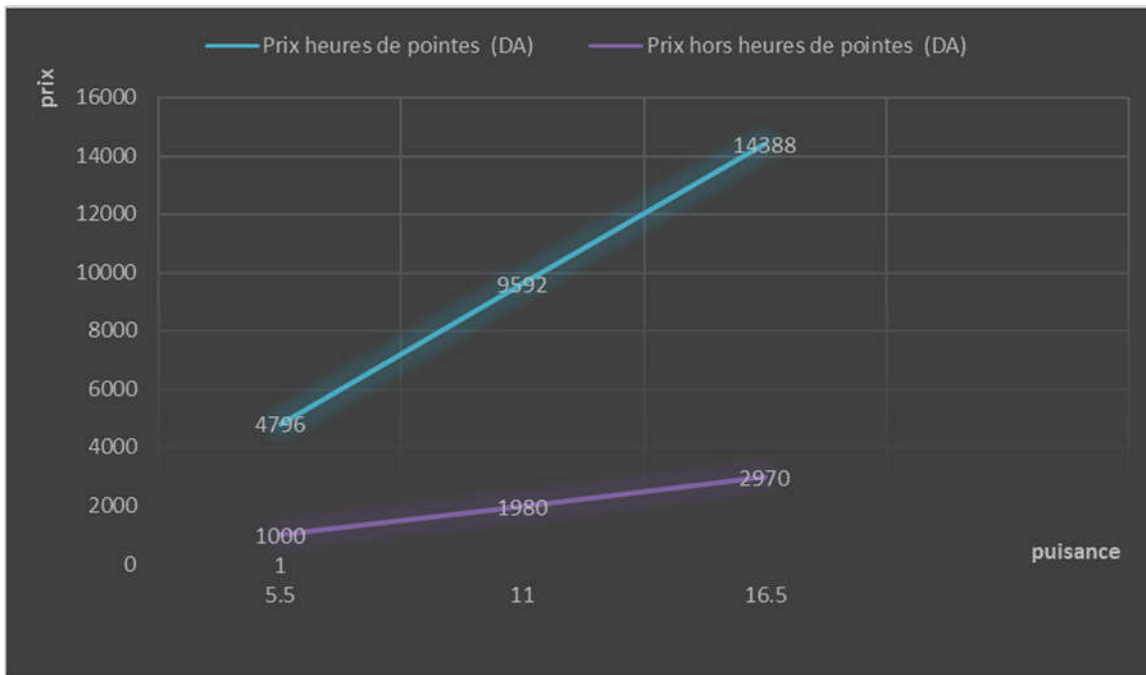
g = c'est la gravité (9.81N/kg).

##### 5.1 Exemple de la consommation électrique pendant une journée

Dans le tableau 3.1, nous donnons un exemple de la consommation électrique d'une pompe en se référant au tableau 1.1.

Q (m <sup>3</sup> )	P(KW)	Prix heures de pointes (DA)	Prix hors heures de pointes (DA)
10	5.5	4796	1000
20	11	9592	1980
30	16.5	14388	2970

Tableau 3.1 ; Exemple de la consommation électrique.

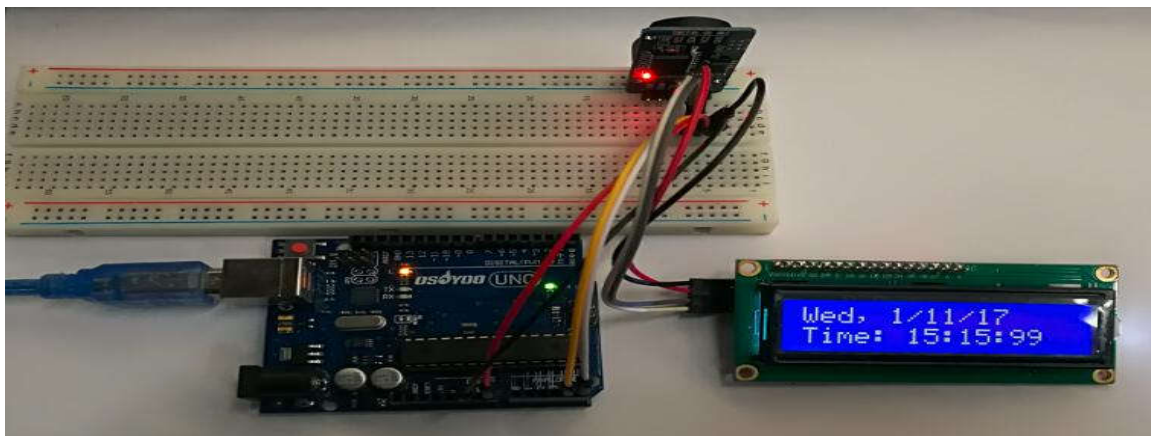


*Figure 3.7;* les prix pour 2 périodes

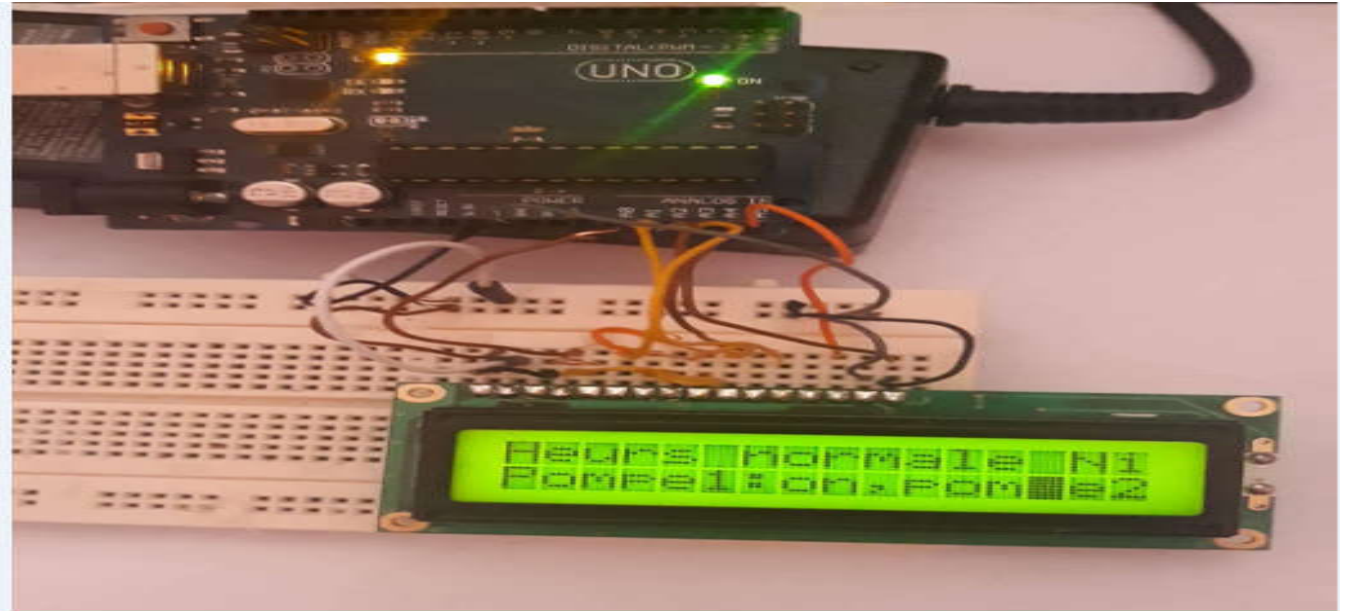
D'après ces données nous remarquons qu'il y a une grande différence entre les couts pondant les heures de pointes et hors les heures de pointes.

## 6. Réalisation pratique :

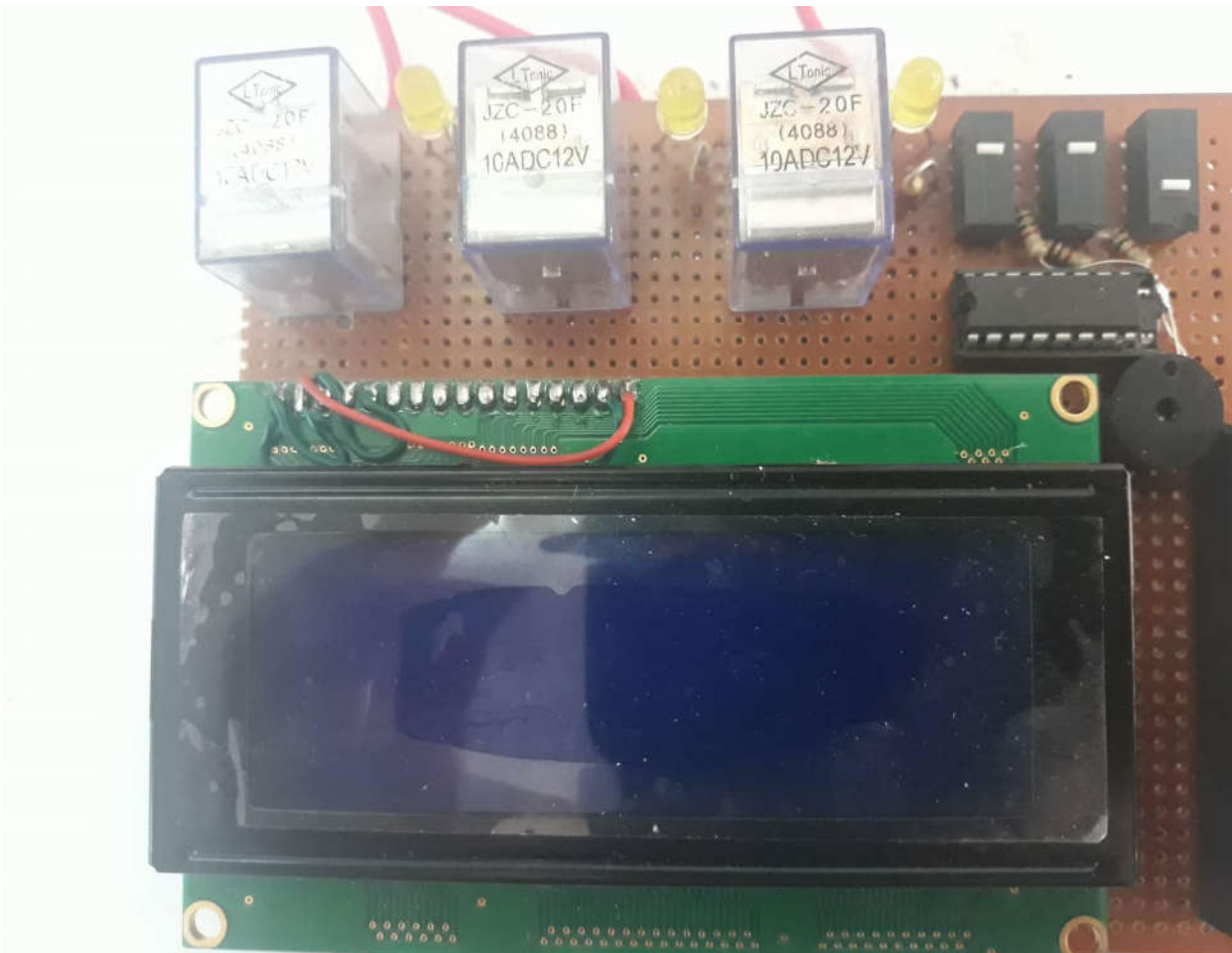
Pour la concrétisation de notre projet qui consiste à la réalisation du circuit d'une carte de commande d'une station de pompage d'eau, nous avons entamé en premier lieu notre travail sur plaque d'essais, en testant le circuit dans des conditions réelles regroupant les éléments requis pour l'obtention du dispositif final. Les figures 3.8, 3.9 et 3.10 représentent l'horloge sur plaque d'essai, le bloc de pompes sur plaque d'essai et la carte de commande finalisée, respectivement.



*Figure 3.8 ;* L'horloge sur plaque d'essai.



**Figure 3.9** ; Le bloc de pompes sur plaque d'essai

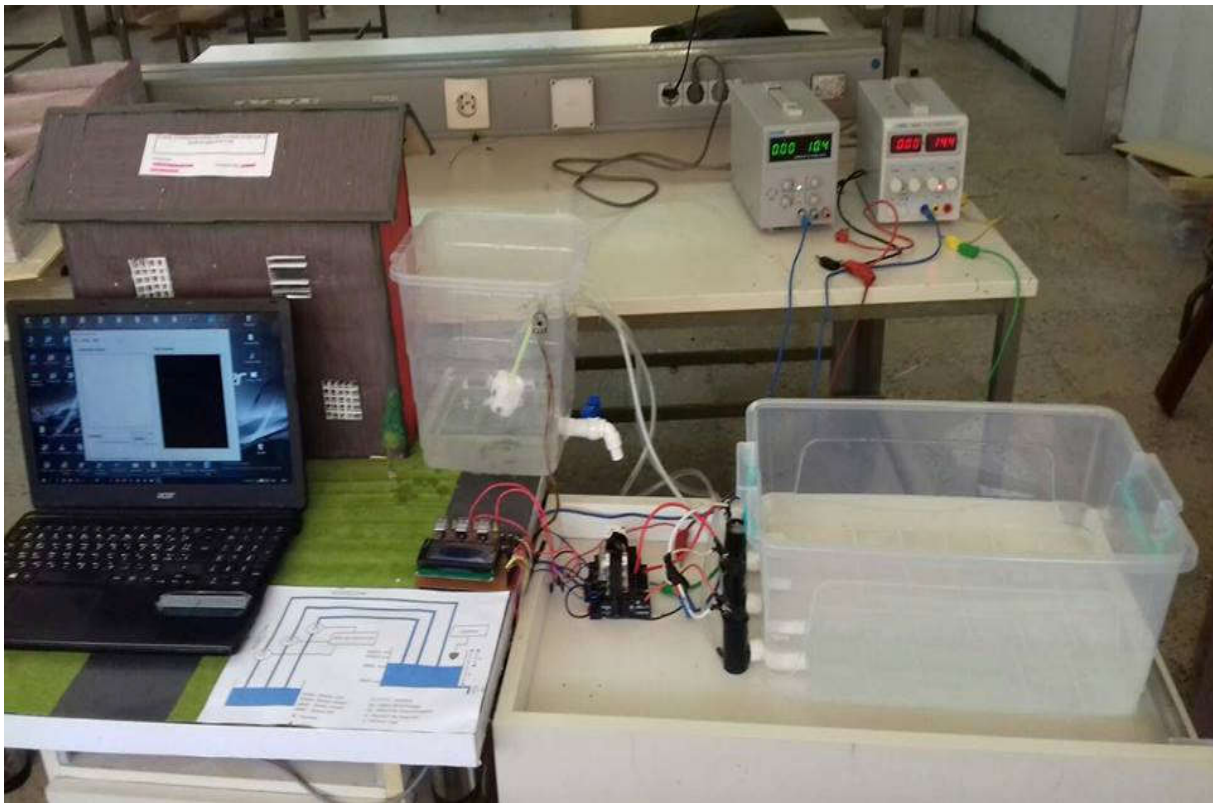


**Figure 3.10** ; carte de commande finalisée

Le tableau 3.2 regroupe tous les éléments électroniques utilisés dans la réalisation de notre projet.

R1.R2.R3	330Ω
RTC	DS3231
LCD	4*20
D1. D2. D3	Diode LED
BZZU	BUZZER
Sw1.sw2.sw3	Boutons poussoirs
RELAIS	//

**Tableau 3.2** ; liste de composants



**Figure 3.11**; marquette finale

## 7. Conclusion

Pour cette dernière étape de réalisation finale d'une carte de commande d'une station de pompage d'eau, nous avons commencé avec l'algorithme d'optimisation, ensuite nous sommes passés à la partie simulation, puis la mise des différents composants sur la plaque d'essai, en utilisant la programmation sous logiciel ARDUINO, et enfin la plaque du circuit était réalisée.

La réalisation pratique du montage était pour nous une expérience très avantageuse du fait que no

### Conclusion générale

Ce projet de fin d'étude rentre dans le cadre de l'optimisation de la consommation d'eau pour maîtriser la consommation électrique. Nous avons appris beaucoup de choses, notamment

l'importance du travail en équipe, la répartition des tâches et la résolution d'un problème. Il faut préciser que notre travail ne s'est pas déroulé sans aucun problème, néanmoins nous avons réussi à mettre en œuvre un projet fonctionnel avec l'aide de nos professeurs que nous tenons à remercier pour tout le savoir et l'aide qu'ils nous ont fournis.

L'accomplissement de ce projet nous a permis de nous familiariser profondément avec beaucoup de dispositifs électroniques innovants tels que l'ARDUINO UNO, *les afficheurs à cristaux liquides LCD*, le **module horloge** RTC DS3231 et le bus I2C.

Dans un futur proche, nous espérons que ce travail sera développé par l'introduction de nouveaux capteurs susceptibles d'améliorer le fonctionnement de notre système.

## Liste des figures

Figure 1.1 ; les composantes principales de système de transfert (page : 11)

Figure 1.2: Consommation d'eau pendant les saisons d'hiver et d'été (page : 12)

Figure 1.3 : Consommation d'eau pendant une journée (page : 13)

Figure 2.1; Schéma synoptique de la carte (page : 16)

Figure 2.2 : la carte de l'arduino uno (page : 17)

Figure 2.3 : schéma électrique de la carte Arduino (page : 18)

*Figure 2.4* : *afficheur LCD 4\*20* (page : 20)

Figure 2.5 : les broches de l'afficheur LCD (page : 22)

Figure 2.6 ; Montage de l'afficheur LCD sur l'ARDUINO (page : 23)

Figure 2.7 ; Montage RTC DS3231 (page : 25)

Figure 2.8 ; Montage interne RTC DS3231 (page : 25)

Figure 2.9; connexion d'horloge avec l'arduino (page : 26)

Figure 2.10; horloge avec l'arduino (page : 26)

Figure 2.11; horloge avec l'arduino et LCD (page : 26)

Figure 2.12; Montage du relais (page : 27)

Figure 2.13 ; Montage des boutons poussoirs (page : 27)

Figure 2.14; architecture-bus-I2C (page : 28)

Figure 2.15; Montage du BUZZER (page : 29)

Figure 2.16 ; L'interconnexion entre arduino buzzeur et écran LCD (page : 29)

Figure 3.1 : schéma de fonctionnement (page : 34)

Figure 3.2 ; schéma expliquant la simulation de la carte sur Isis Proteus (page : 35)

Figure 3.3; Organigramme du programme principal (page : 36)

Figure 3.4; Organigramme de Sous-programme de control des heures de pointe (page : 37)

Figure 3.5 ; Organigramme de Sous-programme de control de niveau aux heures de pointe (p : 37)

Figure 3.6; Organigramme de Sous-programme de control de niveau hors heures de pointe (p : 37)

*Figure 3.7*; les prix pour 2 périodes (page : 39)

Figure 3.8; L'horloge sur plaque d'essai. (page : 39)

Figure 3.9 ; Le bloc de pompes sur plaque d'essai (page : 40)

Figure 3.10; carte de commande finalisée (page : 40)

*Figure 3.11*; marquette finale (page :41)

Liste des tableaux

Tableau1.1 : Présentation des horaires et prix journaliers (page :13)

Tableau2.1 ;Caractéristiques de l'afficheur LCD (page :21)

Tableau3.1 ; Exemple de la consommation électrique. (page :38)

Tableau 3.2 ; liste de composants (page : 41)



# ULN2001A-ULN2002A ULN2003A-ULN2004A

## SEVEN DARLINGTON ARRAYS

- SEVEN DARLINGTONS PER PACKAGE
- OUTPUT CURRENT 500mA PER DRIVER (600mA PEAK)
- OUTPUT VOLTAGE 50V
- INTEGRATED SUPPRESSION DIODES FOR INDUCTIVE LOADS
- OUTPUTS CAN BE PARALLELED FOR HIGHER CURRENT
- TTL/CMOS/PMOS/DTL COMPATIBLE INPUTS
- INPUTS PINNED OPPOSITE OUTPUTS TO SIMPLIFY LAYOUT

### DESCRIPTION

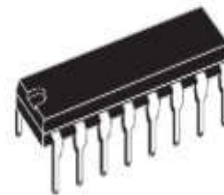
The ULN2001A, ULN2002A, ULN2003 and ULN2004A are high voltage, high current darlington arrays each containing seven open collector darlington pairs with common emitters. Each channel rated at 500mA and can withstand peak currents of 600mA. Suppression diodes are included for inductive load driving and the inputs are pinned opposite the outputs to simplify board layout.

The four versions interface to all common logic families :

ULN2001A	General Purpose, DTL, TTL, PMOS, CMOS
ULN2002A	14-25V PMOS
ULN2003A	5V TTL, CMOS
ULN2004A	6-15V CMOS, PMOS

These versatile devices are useful for driving a wide range of loads including solenoids, relays DC motors, LED displays filament lamps, thermal print-heads and high power buffers.

The ULN2001A/2002A/2003A and 2004A are supplied in 16 pin plastic DIP packages with a copper leadframe to reduce thermal resistance. They are available also in small outline package (SO-16) as ULN2001D/2002D/2003D/2004D.



DIP16

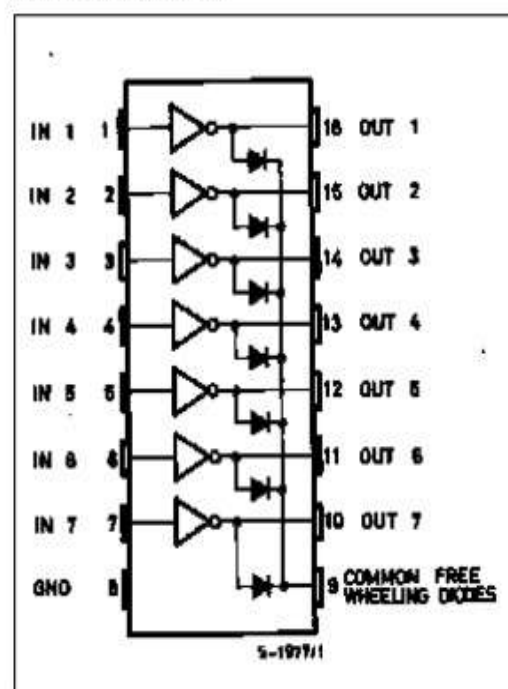
ORDERING NUMBERS: ULN2001A/2A/3A/4A



SO16

ORDERING NUMBERS: ULN2001D/2D/3D/4D

### PIN CONNECTION



# Bibliographie

## Les sites :

- [1] : [https://fr.wikipedia.org/wiki/Station\\_de\\_pompage](https://fr.wikipedia.org/wiki/Station_de_pompage)
- [3] : <https://fr.wikipedia.org/wiki/Pompe>
- [4] : [http://www.microsann.com/images/Atelier\\_Joomla/Fiches\\_PDF/La\\_carte\\_Arduino\\_UNO](http://www.microsann.com/images/Atelier_Joomla/Fiches_PDF/La_carte_Arduino_UNO)
- [5] : <https://www.aurel32.net/elec/lcd.php>
- [6] : <https://youpilab.com/home-page/module-horloge-ds3231-rtc.html>.
- [7] : <http://www.hager.be/produits-e-catalogue/distribution-d-energie/commande-signalisation-et-mesure/commande-signalisation/les-avantages-des-contacteurs-et-des-relais/8864.htm>.
- [8] : [https://les-electroniciens.com/sites/default/files/cours/cours\\_i2c.pdf](https://les-electroniciens.com/sites/default/files/cours/cours_i2c.pdf).

## Les mémoires :

- [2] : Projet de fin d'étude : Optimisation des opérations de pompage d'un réseau de distribution d'eau selon les tarifs de l'électricité 2016/2017.