

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
UNIVERSITE MOULOUD MAMMARI DE TIZI-OUZOU



FACULTE DU GENIE ELECTRIQUE ET D'INFORMATIQUE

DEPARTEMENT D'ELECTRONIQUE

Mémoire de Fin d'Etudes

De MASTER PROFESSIONNEL

Domaine : Sciences et Technologies

Filière : **Electronique**

Spécialité : **Micro-électronique**

Présenté par

Mr ALLEG Syphax

Mr ARAB Meziane

Thème

**Conception et réalisation d'un régulateur de
température et application dans un réservoir d'eau.**

Mémoire soutenu publiquement le **29 /06/2025** devant le jury composé de :

Mme BOUDIA Ouerdia	MAA	UMMTO	Présidente
Mme BOUSSOUM Ouiza	MCB	UMMTO	Examinatrice
Mr KOUSSA Mustapha	Directeur de recherche	CDER	Encadreur
Mme BOUARABA Fazia	MCB	UMMTO	Co-Promotrice

Année universitaire 2024-2025

Remerciement

Tout d'abord nous tenons à remercier Dieu, le Tout-Puissant, de nous avoir donné la force et le courage de mener à bien ce modeste travail. Tout cela n'aurait pas été possible sans sa bénédiction.

Nous tenons aussi à exprimer notre sincère gratitude envers **Monsieur KOUSSA Mustapha**, Maître de Recherche au Centre de Développement des Énergies Renouvelables (CDER) et promoteur de ce mémoire. Son encadrement, ses conseils précieux et sa disponibilité tout au long de ce travail ont été déterminants. Sa compétence, sa méthode scientifique stricte et sa bienveillance ont joué un rôle crucial dans le succès de ce projet. Nous lui sommes profondément reconnaissants pour la confiance qu'il nous a témoignée et pour l'attention qu'il a accordé à notre projet.

Nous adressons également nos sincères remerciements à **Mme BOUARABA Fazia**, enseignante à UMMTO et co-promotrice de mémoire, pour son accompagnement, sa patience et ses encouragements constants. Sa disponibilité, ses conseils avisés et son soutien moral nous ont été d'une grande aide, notamment lors des étapes les plus délicates de ce travail.

Nos remerciements s'adressent aussi à l'ensemble des **enseignants** du département d'électronique, spécialité microélectronique, de l'UMMTO, pour la qualité de leur enseignement, leur accompagnement et leur engagement à transmettre leur savoir. Nous n'oublions pas les membres du personnel administratif pour leur efficacité, et leur gentillesse tout au long de notre parcours.

Nous exprimons notre reconnaissance aux **membres de jury** d'avoir accepté de juger notre travail, le temps qu'ils lui consacrent et la pertinence de leurs remarques qui ne manqueront pas d'enrichir notre mémoire.

Nous remercions chaleureusement **nos camarades** et amis pour leur aide, leur soutien, leur bonne humeur et les moments de partage qui ont rythmé ces années d'études. Leur présence a rendu cette expérience universitaire plus agréable et enrichissante. Enfin, nous adressons une pensée particulière à nos familles, et notamment à nos parents, pour leur amour, leur patience, leur confiance et leur soutien indéfectible tout au long de notre parcours. Rien de tout cela n'aurait été possible sans leur encouragement et leur présence à nos côtés.

À tous, nous disons un grand merci du fond du cœur.

Dédicaces

Avec toute ma reconnaissance,

Je dédie ce travail à mon père **Salem**,

Qui m'a toujours encouragé à poursuivre mes rêves et à croire en moi.

À ma mère **Safia**,

Pour son amour inconditionnel, sa patience et son soutien indéfectible.

À mes trois frères **Ahmed**, **M'hend**, et **Yuva**,

Mes compagnons de route, pour leur solidarité et leur soutien inestimable.

À ma tante **Houria et ses enfants**, en particulier **Taoues**,

Pour leur générosité, leurs conseils et leur bienveillance.

À ma tante **Malika**,

Partie trop tôt, mais dont le souvenir reste vivant dans mon cœur.

Merci pour ton amour et tes précieux conseils, qui m'accompagnent chaque jour.

À tous mes cousins et cousines, en particulier **HMIMMED**,

Pour l'énergie, la joie et le soutien.

À toute **ma famille**,

Pour leur affection.

À mon cher frère et **Binôme ARAB Meziane**, et toute sa famille.

À tous **mes amis**,

Pour leur présence, leur énergie et leur bonne humeur.

Dédicaces

Avec toute ma reconnaissance,

Je dédie ce mémoire à mes chers parents

À mon défunt père **Nacer** parti trop tôt, mais toujours vivant en moi,

Qui m'avait toujours encouragé à poursuivre mes études et de croire en moi. Ce mémoire, c'est aussi pour toi en hommage à tout ce que tu m'as transmis.

À ma mère **Fazia**

Chaque linge de ce mémoire porte l'empreinte de tes prières, de tes sacrifices silencieux et de ton amour infini

À mes deux sœurs **Katia** et **ouardia**

Mes confidentes, mes piliers

Merci pour votre soutien, votre patience et votre amour m'ont donné la force d'aller jusqu'au bout de ce chemin

Ainsi nos petits anges **Yasten** et **Émilie**

Aussi mes beaux-frères **Mehdi** et **Mouloud**

Des frères que la vie m'a offerts. Merci pour votre présence, vos conseils et votre encouragement tout au long de mon parcours.

Ma tante **Houria** une seconde mère, une présence rassurante. Merci pour ton amour, ton soutien et ta générosité.

Ma grand-mère **Ourdia** partie mais toujours vivante dans mon cœur. Je t'écris ces mots avec une pensée émue, en espérant que, de là où tu es, tu sois fière de moi.

À mes Chers Amis,

Pour leur présence, leur énergie et leur bonne humeur.

Meziane

Résumé

La production de l'eau chaude solaire, constitue un secteur primordial dans les applications de l'énergie solaire. Par ailleurs, généralement durant les périodes creuses, ou l'eau contenue dans le réservoir n'est pas suffisamment chauffée, un système d'appoint en matière de chauffage s'impose afin de combler le déficit énergétique résultant et d'assurer le bon fonctionnement du système, il s'agit d'un thermoplongeur. Notre travail consiste à concevoir et réaliser un régulateur de température adapté à un environnement précis. Il comprend une étude théorique des principes de régulation, notamment le fonctionnement des régulateurs tout ou rien (TOR) et proportionnel-intégral-dérivé (PID), ainsi que le choix des composants essentiels, tels que les capteurs de température et les actionneurs (relais ou éléments chauffants).

La réalisation pratique a confirmé l'efficacité et la fiabilité du système en condition réelles. Grâce à l'affichage LCD, la température est surveillée en temps réel et la régulation s'effectue automatiquement via l'activation ou la désactivation de la pompe et la résistance selon les besoins.

Anglais :

This thesis presents the design and implementation of an automatic system to regulate the temperature of a water tank, combining solar energy and an electric heating element

Following an in-depth study of control principles (ON/OFF, PID, etc.), the project details the selection of components (DS18B20 sensors, Arduino, relays, LCD), simulation on Proteus and then practical implementation on a prototype and printed circuit board. The tests show that the system is both reliable, easy to use, and energy efficient, suitable for both domestic and industrial applications. The work offers opportunities for improvement, such as integrating additional sensors and optimizing energy efficiency.

Mots clés :

- **Régulateur de température**
- **Chauffe-eau solaire**

- **Capteur DS18B20**
- **Régulateur tout ou rien**
- **Energie solaire thermique**

Table de matière

Table des figures

Liste des tableaux

Liste des abréviations

Introduction générale :1

Chapitre I: Généralité sur la régulation industrielle

I.1. Introduction	3
I.2. L'objectif global de la régulation	3
I.3. Définition de la régulation	3
I.4. Chaines de régulations	4
I.4.1 Régulation en boucle ouverte	4
I.4.2. Régulation en boucle fermée	5
I.5. Composition de la boucle de régulation	5
I.5.1 La consigne SP (Set Point)	5
I.5.2 La sortie X (system Output)	5
I.5.3 La mesure PV (Procès Variable)	6
I.5.4 La perturbation Z	6
I.5.5 L'actionneur	6
I.5.6 Le comparateur	6
I.5.7. Le régulateur	6
I.6. Les différents types de régulation	7
I.6.1. Régulateur tout ou rien (ON-OFF)	7
I.6.2 Régulateur continue	7
I.7. Critères de performance d'une régulation	11

I.7.1 La stabilité	11
I.7.2 La précision	11
I.7.3. La rapidité	12
I.8. Transition des régulateurs classiques vers la régulation numérique dans les systèmes modernes	12
I.9. Comparaison entre régulateur continue et régulateur tout ou rien	13
I.10 Conclusion	14

CHAPITRE II : Description matérielle et simulation.

II.1 Introduction	15
II.2. Présentation globale du régulateur	15
II.3 Description des composants utilisés	16
II.3.1 Capteur de température DS18B20	16
II.3.2 Afficheur LCD 20×4 avec interface I2C	18
II.3.3 Relais électrique 5VSRD-05VDC-SL-C	20
II.3.4 Le capteur de température et d'humidité (DHT11)	21
II.3.5. La Carte Arduino	22
II.3.6 Bouton Poussoir	23
II.4 Capteur solaire thermique	24
II.4.1 Fonctionnement d'un capteur solaire thermique	24
II.5 Les différents types d'installation	27
II.6. Partie logicielle	29
II.6.1. Présentation du logiciel utilisé	29
II.6.2 L'environnement Arduino	29
II.6.3 Structure d'un programme Arduino	31
II.6.4 Proteus (ISIS et ARES)	32
II.7 Conclusion	35

Chapitre III: Tests de validation et résultats expérimentaux

III.1. Introduction	36
III.2. Simulation et premiers tests	36
III.1.1. Simulation sur Proteus avec DS18B20	36
III.1.2. Simulations sur Proteus avec LM35	37
III.1.3. Mise en œuvre du régulateur complet en simulation	38
III.1.4. Organigramme de fonctionnement du régulateur de température	40
III.2. Réalisation de la maquette d'essai	42
III.2.1. Présentation de la LAB d'essai	42
III.2.2. Liste des matériaux utilisés sur notre montage	42
III.2.3. Fonctionnement général du système	45
III.3. Réalisations final sur circuit imprimé	47

III.3.1. Etapes de conception de circuit imprimé	47
III.3.2. Equivalence des broches entre Arduino et ATmega328 pour le câblage des composants	50
III.3.3. Vue d'ensemble du montage final réalisé	51
III.3.4. Tests et validation du montage	53
III.4. Conclusion	53
Conclusion générale	55
Références bibliographique	Erreur ! Signet non défini.

Liste des tableaux :

Tableau I. 1 : Table de comparaison des régulateurs continus et tout ou rien (TOR)	13
Tableau II. 1 : Caractéristiques principales de l'afficheur LCD.....	20
Tableau II. 2 : Quelques cartes Arduino et leurs caractéristiques.....	22
Tableau III.1 : rôle des composants	47

Liste des figures

Figure I. 1 La boucle de régulation [2].....	4
Figure I. 2 Système en boucle ouverte [1].....	5
Figure I. 3 Système en boucle fermé [1].....	5
Figure I. 4 Éléments constitutifs d'une boucle de régulation.....	6
Figure I. 5 Régulation tout ou rien	7
Figure I. 6 L'effet de l'action proportionnelle (P) [27]	8
Figure I. 7 L'effet de l'action intégrale (I) [6].....	9
Figure I. 8 Régulateur de type PI [27].....	9
Figure I. 9 Régulateur PID [7]	10
Figure I. 10 Schéma global du régulateur.....	15
 Boucle solaire fermée Circulation forcée [11] Capteur solaire thermique à tubes sous vide (caloduc) [24].	
Figure II. 1 Schéma global du regulateur présenté sur proteus	17
Figure II. 2 Capteur de température DS18B20 [13].....	18
Figure II. 3 Branchement du DS18B20 avec Arduino [16].....	19
Figure II. 4 Afficheur LCD20×S4 I2C	Erreur ! Signet non défini.
Figure II. 5 Composition d'un écran LCD [17].....	19
Figure II. 6 Relais électrique 5V [19]	21
Figure II. 7 Le Capteur DHT11.....	23
Figure II. 8 Arduino UNO (R3) [23]	24
Figure II. 9 Capteur solaire thermique à tubes sous vide (caloduc)	27
Figure II. 10 Capteur turbulaire sous vide [24]	28
Figure II. 11 Capteur solaire plan vitré [24].....	28
Figure II. 12 Boucle solaire fermée Circulation forcée [11]	27
Figure II.13 Boucle ouverte [11]	27
Figure II. 14 Circulation thermosiphon	28
Figure II.15 Interface de logiciel Arduino IDE	33
Figure II.16 Les boutons de l'IDE.....	33
Figure II.17 Interface Proteus	33

Figure III. 1 Visualisation de l'erreur du capteur DS28B20 simulé sous Proteus.....	37
Figure III. 2 Visualisation de l'efficacité du capteur LM35 simulé sur Proteus	38
Figure III. 3 Visualisation du REGULATEUR complet.....	39
Figure III. 4 branchement des composant	43
Figure III. 5 Installation d'un chauffe-eau solaire	45
Figure III. 6 schéma électrique.....	48
Figure III. 7 Emplacements des différents composants sur le PCB.....	49
Figure III. 8 Equivalence entre les broches Arduino et les pattes de l'ATmega328 [29].....	51

Liste des abréviations

SP: Set Point

PV: procès variable

X: output (sortie)

Z: perturbation

TOR: tout ou rien

P: proportionnel

I: Intégrale

D: Dérivée

PID: Proportionnel, Intégrale, Dérivé

PI: Proportionnel, Intégrale

LCD: Liquid Crystal Display

I2C: Inter –Integrated Circuit

PCB: Printed Circuit Board

VCC: Voltage Common Collector

GND: Ground (Masse)

DQ: Data

PWM: Pulse Width Modulation

IDE: Integrated Development Environment

CAO: Conception Assistée par ordinateur

SRAM: Static Random Access Memory

EEPROM: Electrically Erasable Programmable Read –Only Memory

RST: Reset

SDA/SCL: Broches du bus I2C

Introduction générale

Introduction générale :

La régulation industrielle est indispensable pour gérer et améliorer les processus, particulièrement face à l'évolution rapide de l'automatisation et de l'informatique et dans un monde où le confort de tous les jours et la sauvegarde de l'énergie sont des enjeux majeurs.

La température peut être un paramètre analogique simple, mais elle est très difficile à contrôler, elle joue un rôle essentiel dans plusieurs secteurs : procédés thermiques, chauffage, gestion d'énergie à domicile ou industrielle. Il est donc essentiel de surveiller la température pour automatiser efficacement les systèmes.

Notre travail consiste à concevoir et réaliser un régulateur de température adapté à un environnement précis. Il comprend une étude théorique des principes de régulation, notamment le fonctionnement des régulateurs tout ou rien (TOR) et proportionnel-intégral-dérivé (PID), ainsi que le choix des composants essentiels, tels que les capteurs de température et les actionneurs (relais ou éléments chauffants).

La simulation du système a été effectuée en utilisant le logiciel Proteus. Le montage de base a été réalisé dans un premier temps sur un LAB d'essai afin de vérifier le bon fonctionnement de système en utilisant des composants simples ; des capteurs DS18B20 afin de mesurer la température avec précision, un écran LCD I2C pour visualiser les données, des relais pour gérer les équipements et une carte Arduino UNO.

Après avoir vérifié le bon fonctionnement sur le prototype, nous avons réalisé un circuit imprimé (PCB) pour assurer un montage plus robuste et fiable. Cette phase nous a permis d'organiser correctement les composants, de sécuriser les connexions et de préparer le système pour une utilisation de longue durée. Nous avons également veillé à connecter chaque composant aux broches appropriées du microcontrôleur.

Enfin, des tests en conditions réelles ont été effectués afin d'évaluer la précision et la stabilité du dispositif, avec une attention particulière portée à l'optimisation des performances pour garantir un fonctionnement fiable.

Le mémoire résumant le travail effectué est organisé autour de trois chapitres :

Le premier chapitre regroupe les résultats d'une recherche bibliographique mettant en évidence les bases essentielles à la compréhension de ce travail, en particulier les fondamentaux

concernant les régulateurs industrielles, leurs types, la composition de la boucle de régulation, avec une attention particulière aux régulateurs de température.

Le deuxième chapitre est consacré à la présentation des différents équipements électroniques et les circuits que nous avons mis en œuvre dans le cadre de la réalisation de notre régulateur de température, ainsi que les logiciels utilisés pour programmer le microcontrôleur et aussi le programme de simulation utiliser afin de valider le comportement du système avant sa réalisation pratique.

Le troisième chapitre présent les différentes étapes expérimentales de la mise en œuvre du projet : la réalisation pratique sur LAB d'essai et circuit imprimé, ainsi que les tests fonctionnels et l'analyse du comportement du régulateur.

Nous terminerons notre travail par une conclusion générale et des perspectives.

Chapitre I

Généralité sur la régulation industrielle

I.1. Introduction :

L'importance de la régulation dans le domaine des procédés industriels ne cesse d'augmenter, vu le développement massif des outils informatiques et technologiques, qui ont permis de mettre en œuvre des nouvelles techniques avancées de commande, d'analyse et de supervision des processus.

Le premier chapitre regroupe les résultats d'une recherche bibliographique mettant en évidence les bases essentielles à la compréhension de ce mémoire, en particulier les fondamentaux concernant les régulateurs industriels, leurs types, la composition de la boucle de régulation, avec une attention particulière aux régulateurs de température.

I.2. L'objectif global de la régulation :

L'objectif d'une régulation est d'assurer le fonctionnement d'un procédé selon les critères prédéfinis par un cahier des charges. Son objectif principal est de contrôler une ou plusieurs grandeurs mesurées quelles que soient les perturbations qui agissent sur le procédé tout en assurant la stabilité, la précision et la robustesse de la loi de commande [1].

Cependant, chaque procédé possède ses exigences propres, il est donc indispensable que la régulation soit conçue pour satisfaire aux besoins particuliers liés à la sécurité, aux impératifs de production et aux matériels.

I.3. Définition de la régulation :

La régulation est un ensemble de méthodes utilisées pour contrôler des paramètres physiques tels que la température, la vitesse ou la pression, sans intervention humaine, afin de les maintenir à une valeur prédéterminée, appelée consigne. Cette approche fait partie de l'automatisme [2].

Ces méthodes ne servent pas uniquement à gérer des processus, mais elles offrent également un moyen efficace de réduire les pertes de production, d'améliorer la qualité et la quantité des produits, d'augmenter la disponibilité des unités de production et de diminuer les coûts associés à la production.

En cas de perturbations ou de modifications de la consigne, la régulation déclenche une action corrective sur un paramètre physique du procédé, connu sous le nom de grandeur de réglage ou commande. La figure suivante représente la boucle de régulation de base.

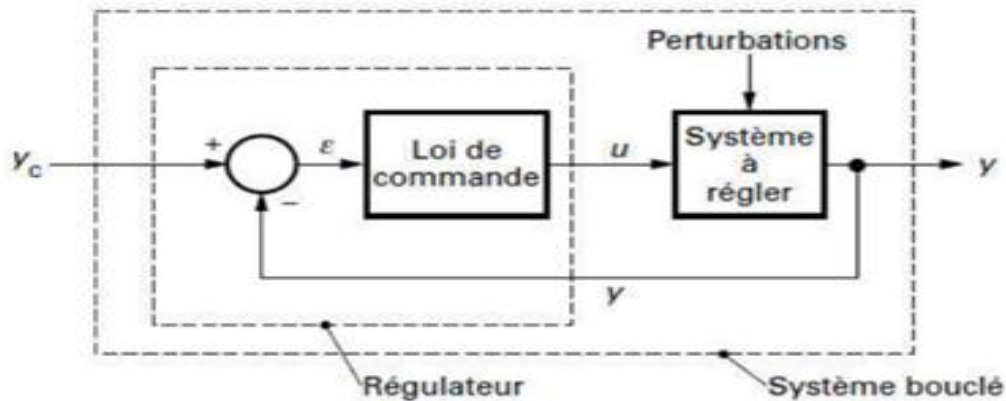


Figure I. 1 : La boucle de régulation [2].

La conduite d'un procédé industriel, qu'il soit chimique, physico-chimique, mécanique, ou électrique nécessite une bonne compréhension et un contrôle précis de plusieurs paramètres clés tels que la pression, la température, le débit et la vitesse, etc...

Pour prendre des décisions éclairées, il est essentiel de recueillir des données précises. Ces informations sont généralement accessibles dans la salle de contrôle via des indicateurs, des enregistreurs et des écrans.

Les commandes nécessaires pour diriger le processus peuvent être effectuées soit directement sur le site, près de l'équipement, soit depuis la salle de contrôle-commande. Ces commandes peuvent être manuelles ou automatiques, programmées sur des ordinateurs grâce à la régulation numérique [2].

I.4. Chaines de régulations :

I.4.1 Régulation en boucle ouverte :

Une régulation est dite en boucle ouverte lorsque les signaux d'entrées ne sont pas influencés par les signaux de sorties. Parmi les inconvénients, il est impossible de contrôler, et encore moins de compenser les erreurs, les dérives, les accidents qui peuvent intervenir à l'intérieur de la boucle, autrement dit, il n'y a pas de précision ni surtout de fidélité qui dépendent de la qualité intrinsèque des composants. Le système en boucle ouverte ne compense pas les signaux de perturbation [3].

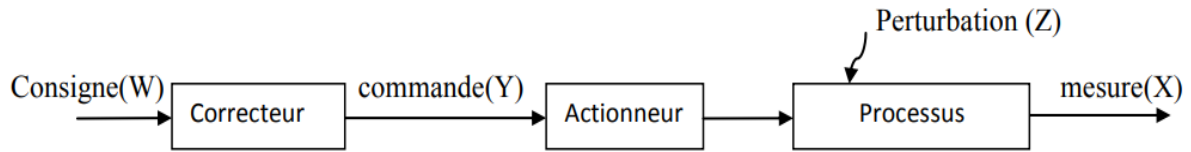


Figure I. 2 : Système en boucle ouverte [1].

I.4.2. Régulation en boucle fermée :

Une boucle est dite fermée si les signaux d'entrée sont influencés par les signaux de sorties. Elle est capable de stabiliser un système instable en boucle ouverte. Dans une régulation en boucle fermée, une bonne partie des facteurs perturbateurs externes sont automatiquement compensés par la contre-réaction à travers le procédé. L'utilisation du retour d'information est le principe fondamental en automatique. La commande appliquée au système est élaborée en fonction de la consigne et de la sortie [3].

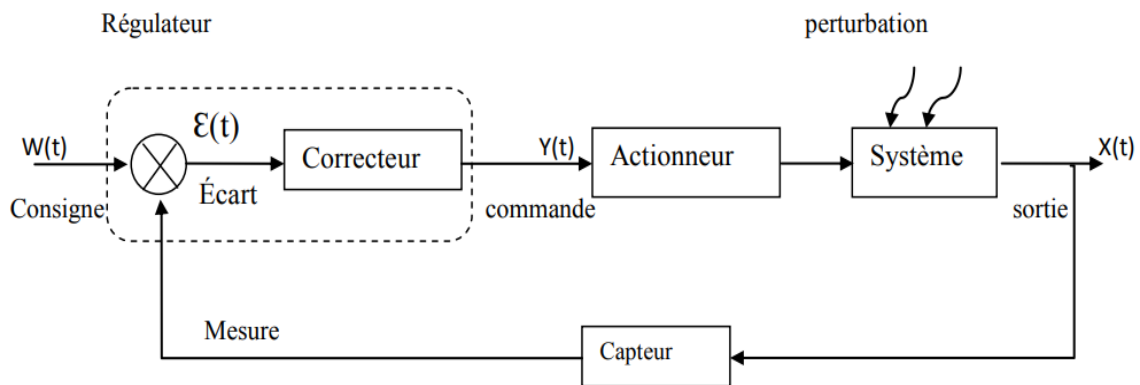


Figure I. 3 : Système en boucle fermé [1].

I.5. Composition de la boucle de régulation :

Une boucle de régulation doit comporter au minimum les éléments suivants :

I.5.1 La consigne SP (Set Point) :

La consigne c'est la valeur donnée à l'entrée et reçu à la sortie, si la consigne est constante, la sortie doit le rester (comportement statique).

I.5.2 La sortie X (system Output) :

La sortie c'est la variable que le système essaie de contrôler en la maintenant constante. Cette variable représente la valeur réelle de la grandeur mesurée.

I.5.3 La mesure PV (Procès Variable) :

Dans une boucle de régulation, il est possible de réagir à toutes les variations indésirables car la sortie est constamment mesurée.

I.5.4 La perturbation Z :

Les perturbations sont des facteurs externes qui affectent négativement le résultat d'un système, l'éloignant de la valeur souhaitée. On utilise une régulation statique pour maintenir le système stable lors de la présence de ces perturbations [5].

I.5.5 L'actionneur :

C'est l'organe exécutif de la boucle de régulation, il reçoit le signal de commande du régulateur et le transforme en une action physique sur le procédé, il prend aucune décision il exécute directement les ordres reçus. Sa rapidité, sa précision et sa fiabilité sont essentiels pour garantir l'efficacité de la régulation.

I.5.6 Le comparateur :

C'est l'endroit où la valeur mesurée et la valeur de la consigne se rencontrent, il effectue la différence entre ces deux valeurs pour déterminer l'erreur, cette dernière représente l'écart à corriger.

I.5.7. Le régulateur :

Le régulateur joue un rôle clé, il reçoit l'écart entre la valeur mesurée et la consigne, puis décide de la meilleure action qu'il doit prendre pour mettre le système à l'équilibre.

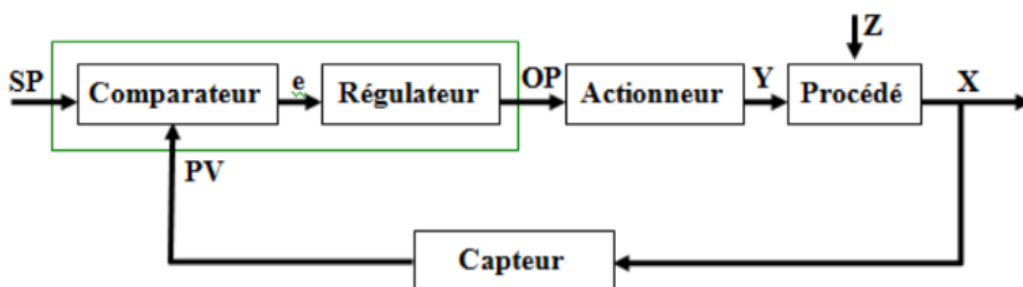


Figure I. 4 : Éléments constitutifs d'une boucle de régulation [1]

I.6. Les différents types de régulation :

I.6.1. Régulateur tout ou rien (ON-OFF) :

L'intelligence de ce type de régulation est limitée au sens de l'écart. En effet, le régulateur en tout ou rien ne perçoit que la position de la variable à réguler par rapport à la consigne ; il peut donc distinguer si la valeur de l'écart $e = (SP-PV)$ est positive ou négative. Le signal de sortie du régulateur ne peut donc prendre que deux valeurs extrêmes ou limites minimale et maximale soit 0 % et 100 % qui correspondent, dans le cas d'une vanne, à sa position complètement fermée et complètement ouverte on peut alors définir, dans le cas de l'action directe : 100% si $e > 0$ ou 0 % si $e < 0$ [6].

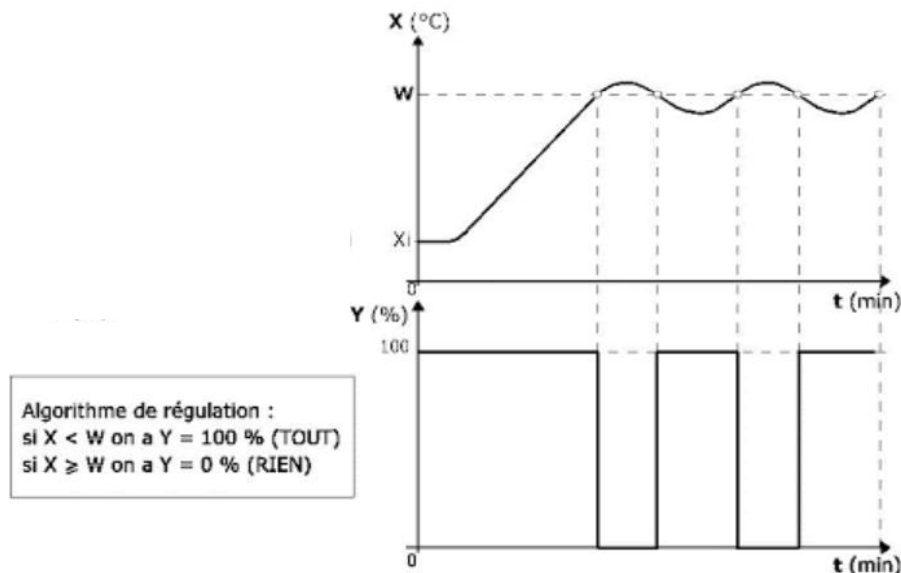


Figure I. 5 Régulation tout ou rien [6]

I.6.2 Régulateur continue :

La plupart des régulations industrielles sont effectuées par les modes d'action continus, où à chaque valeur de l'écart à l'intérieur de l'étendue de régulation, correspond une valeur déterminée et unique du signal de sortie du régulateur. Le signal de sortie ou de correction peut prendre toutes les valeurs comprises entre la portée maximale et la portée minimale de son étendue. Le mode d'action continue, qui peut imposer au signal de sortie une variation continue entre deux limites déterminées, est aussi appelé progressif. L'intelligence ou la perception des régulateurs à action continue va au-delà du seul signe de l'écart ($sp-pv$) et, selon cette perception, les modes en continu sont classés en trois catégories :

- L'action proportionnelle P.
- L'action intégrale I.
- L'action dérivée D.

Chacun de ces modes se distingue par la manière dont le signal d'entrée affecte le signal de sortie. Notons que d'une façon générale les régulateurs sont à action combinée, de deux ou trois des actions P, I et D [6].

I.6.2.1. Le régulateur proportionnel (P) :

Dans le cas d'un régulateur P, la variable régulée est toujours à action proportionnelle à l'erreur calculée. Il en résulte qu'un régulateur P réagit sans temps de retard à tout écart entre la sortie et la consigne, et seul dans ce cas sera générée une valeur régulée : $y = K.e$ [6]

K : Gain proportionnel du régulateur

e: erreur de régulation

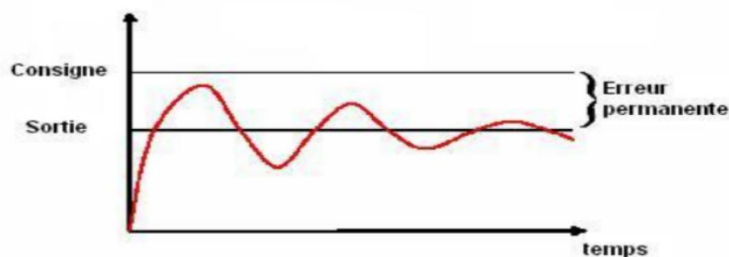


Figure I. 6 L'effet de l'action proportionnelle (P) [27]

I.6.2.2 Le régulateur intégral (I) :

Un régulateur de type intégral permet de complètement compenser une erreur de régulation constante. Tant que l'erreur n'est pas nulle, la valeur de la variable régulée est ajustée. La régulation se termine lorsque la sortie a atteint la valeur de la consigne ou que la variable régulée a atteint un seuil maximal fixé par les propriétés du système (U_{max} , P_{max} ...etc.). La formulation mathématique de ce comportement intégral est : la variable régulée est proportionnelle à l'intégrale par rapport au temps de l'erreur [6].

$$y = Ki \int edt \quad \text{ou} \quad Ki = \frac{1}{Ti}$$

La vitesse avec laquelle la variable régulée augmente (ou diminue) dépend de l'erreur de régulation et de la constante de temps d'intégration choisie.

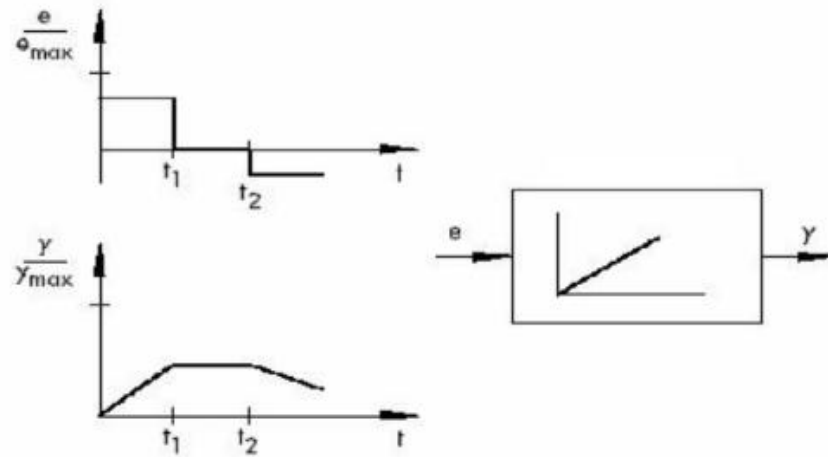


Figure I. 7 L'effet de l'action intégrale (I) [6]

I.6.2.3 Régulateur de type PI :

Le régulateur de type PI est l'un des plus utilisés dans la pratique. Il s'agit d'un couplage parallèle entre un régulateur proportionnel et un régulateur intégral. S'il est correctement paramétré, le régulateur PI cumule les avantages des deux types de régulateurs (stable, rapide, faible erreur résiduelle), tout en compensant leurs désavantages respectifs [27].

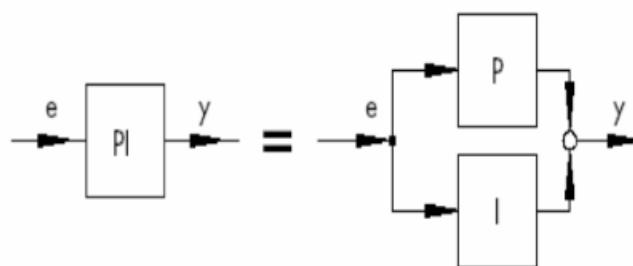


Figure I. 8 Régulateur de type PI [27]

Le comportement dans le temps est caractérisé par le facteur proportionnel K_p et la constante d'intégration T_i . Grâce à la partie proportionnelle, le régulateur réagit immédiatement à toute erreur de régulation, alors que la partie intégrale ne fera effet qu'après un certain temps. La constante d'intégration T_i est le temps qu'il faut à la partie intégrale (I) pour avoir la même

amplitude que prend la partie **P** de façon instantanée. Si l'on veut renforcer la partie intégrale, il faut comme pour un régulateur intégral réduire la constante **T_i**.

Domaine d'application du régulateur P.I : boucle de régulation rapide qui ne tolère aucune erreur résiduelle (ex : régulation de pression, de débit, température...etc.).

I.6.2.4 Le régulateur dérivé (D) :

Le régulateur dérivé établit une valeur régulée en fonction de la vitesse de variation de l'erreur et pas en fonction de l'amplitude comme pour le régulateur P. C'est pour cette raison qu'il réagit beaucoup plus rapidement qu'un régulateur P. Même face à une petite erreur il va générer une grosse valeur régulée dès lors qu'il y a une variation d'amplitude de l'erreur. Le régulateur D sera inefficace face à une erreur résiduelle permanente, quel que soit sa valeur puisque celle-ci reste constante (pas de variation d'amplitude donc pas de réaction du régulateur). C'est pourquoi ce type de régulateur sera rarement utilisé seul dans la pratique, il est couramment associé à un régulateur de type [27].

I.6.2.5 Le régulateur PID :

Un régulateur PID est obtenu par l'association de ces trois actions et il remplit essentiellement les fonctions suivantes :

- Fonction proportionnelle donne un système plus précis et plus rapide.
- Fonction intégrale utilisée pour améliorer la précision et éliminer l'erreur statique.
- Fonction dérivée principalement utilisée pour améliorer la stabilité, anticiper les variations de la sortie et accélérer la correction [7].

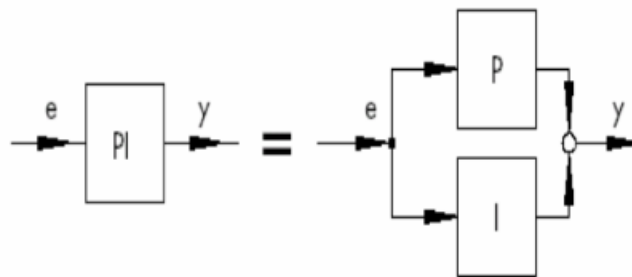


Figure I. 9 Régulateur PID [7]

I.7. Critères de performance d'une régulation :

Dans un système de régulation, les spécifications sont souvent assez floues, principalement parce que les problèmes à résoudre sont très variés. Les critères à respecter dépendent surtout du type de processus à contrôler [8].

Prenons l'exemple d'un ascenseur : pour assurer le confort des passagers, il ne faut pas que l'ascenseur accélère brusquement. De même, dans un réacteur nucléaire, la pression ne doit jamais dépasser certains seuils pour éviter de déclencher les soupapes de sécurité. Il est important de noter que lorsqu'on règle un système de régulation, on se base souvent sur la réponse à un changement de consigne (c'est-à-dire une modification volontaire de la consigne). Pourtant, en réalité, le système doit surtout faire face à des perturbations imprévues.

Les performances d'une régulation peuvent se définir à partir de l'allure du signal de mesure suite à un échelon de consigne. Notons toutefois que les critères de performances classiques peuvent se résumer comme suit :

I.7.1 La stabilité :

La qualité essentielle pour un système de régulation est la stabilité. Dans une approche simplifiée, un système est considéré comme stable si pour une grandeur à maîtriser se stabilise à une valeur finie. Plus le régime transitoire d'un système soumis à une telle variation est amorti plus il est stable. Le degré de stabilité est alors caractérisé par l'amortissement de ce régime transitoire [1].

I.7.2 La précision :

- **Précision statique :** Il est naturel d'évaluer la précision d'un système régulé en comparant l'objectif atteint par rapport à celui exigé. La précision d'un système régulé se mesure donc à l'écart entre la consigne demandée et la mesure en régime permanent ; on parle alors de précision statique. Plus l'écart statique est petit, plus le système est précis. L'évaluation de la précision statique s'effectue en réalisant une variation rapide de consigne en amplitude et en mesurant la variation d'amplitude finalement obtenue de la mesure [5].
- **Précision dynamique :**

La précision dynamique est caractérisée par le dépassement "D" lors du régime transitoire de la réponse de la grandeur réglée suite à un échelon de consigne ou de perturbation. Cette

précision est liée directement au degré de stabilité du procédé ; c'est un critère de performance qui peut être défini par les marges de gain et de phase. Elle est donc à prendre en compte lors des réglages des régulateurs [5].

I.7.3. La rapidité :

On demande en pratique que le système soit capable rapidement de compenser les perturbations et de bien suivre la consigne. La rapidité d'un système régulé s'évalue par le temps nécessaire à la mesure pour entrer dans une zone $\pm 5\%$ de sa valeur finale (soit entre 95% et 105%). Le système régulé est d'autant plus rapide que le temps de réponse à 5% est court [8].

I.8. Transition des régulateurs classiques vers la régulation numérique dans les systèmes modernes :

Les avancées technologiques dans le domaine de l'automatisation industrielle ont profondément transformé les méthodes de régulation. Autrefois, les procédés étaient contrôlés par des dispositifs simples, comme les régulateurs Tout Ou Rien (TOR), aussi appelés régulateurs ON/OFF, qui ne permettent qu'un contrôle binaire du système. Aujourd'hui encore, le régulateur TOR reste très utilisé pour des applications simples et robustes, comme dans les chauffe-eaux, les réfrigérateurs ou les systèmes de chauffage domestique. Son fonctionnement est basé sur un seuil fixe : lorsque la température dépasse la valeur souhaitée, il coupe l'alimentation ; quand elle redescend en dessous, il la remet en marche. Bien que ce type de régulation soit économique et facile à mettre en œuvre, il présente l'inconvénient d'avoir des cycles de marche/arrêt fréquents et une précision limitée [9].

Avec l'augmentation des besoins en précision et en stabilité, les industriels se sont tournés vers des régulateurs analogiques comme les régulateurs P, PI, PID, qui offrent une commande continue et proportionnée à l'erreur mesurée. Le régulateur PID, par exemple, combine une réaction immédiate (action proportionnelle), une suppression de l'erreur permanente (intégrale) et une anticipation des variations (dérivée), permettant ainsi un contrôle plus fin et dynamique du procédé [10].

Cependant, les régulateurs analogiques sont eux-mêmes limités par les tolérances des composants, le bruit électrique, et l'impossibilité d'évolution sans modifications matérielles. Cela a conduit à l'émergence de la régulation numérique, facilitée par les progrès des microcontrôleurs comme l'Arduino. Grâce à une régulation numérique, les paramètres de

commande (K_p , K_i , K_d) peuvent être ajustés en temps réel via un logiciel embarqué, ce qui offre une souplesse et une évolutivité très recherchées dans les installations modernes. [11].

Dans une régulation numérique, la valeur mesurée (PV) est lue par un capteur (thermistance, thermocouple ou capteur numérique), traitée par le microcontrôleur, et la commande est envoyée via un relais TOR ou un signal PWM à l'actionneur (résistance, moteur, vanne, etc.). Cette méthode permet d'utiliser la logique binaire du régulateur TOR tout en profitant de la logique conditionnelle et du traitement numérique pour ajuster dynamiquement la réponse du système. Ce type de régulation est aujourd'hui très courant dans les systèmes embarqués : chauffe-eaux solaires, incubateurs, contrôle de température dans les serres ou encore dans des chaudières intelligentes [12].

I.9. Comparaison entre régulateur continue et régulateur tout ou rien :

Critère	Régulateur tout ou rien	Régulateur continue
Sortie	Binaire (0 ou 100 %)	Continue (0 à 100 %)
Précision	Moins précis	Plus précis, stable
Temps de réponse	Rapide, mais avec cycles fréquents.	Plus lent mais stable.
Cout	Moins cher (équipements simples).	Plus élevé (capteurs et actionneurs plus complexes).
Complexité	Simple, économique, facile à installer	Plus complexe, nécessite réglage
Application typique	Chauffe-eau domestique, thermostat de radiateur, frigo.	Chauffage central, contrôle de vitesse, processus industriels précis.
Facilité de maintenance	Très facile	Plus complexe
Compatibilité automatisation	Facile pour systèmes simples	Idéal pour automatisation avancée

Tableau I. 1 : Table de comparaison des régulateurs continus et tout ou rien (TOR)

La régulation TOR est appropriée pour des systèmes simples où une précision élevée n'est pas critique. C'est un bon choix dans le cas des chauffe-eau domestiques ou d'autres systèmes qui doivent être de conception facile et la plus bon marché possible. En effet, la

régulation TOR fonctionne selon un système binaire, activant ou désactivant la commande en fonction de la valeur mesurée par rapport à une consigne. En comparaison, la régulation PID, encore plus coûteuse et exigeant une complexité de l'utilisation de contrôle. Cela signifie que c'est un bon choix pour les systèmes industriels basés sur les exigences en matière de contrôle les plus précis. Le type de régulateur utilisé dans ce projet est un régulateur Tout Ou Rien (TOR).

Ce choix est adapté à notre installation car elle nécessite une solution simple, fiable et économique pour maintenir la température autour d'une valeur de consigne.

I.10 Conclusion :

Dans ce chapitre nous avons présenté brièvement quelques notions de bases des systèmes de régulations, notamment les différents types de régulateur (TOR, P, I, D, PID), qui jouent un rôle essentiel dans le maintien de la stabilité et de la performance des procédés industriels. La compréhension de ces régulateurs est fondamentale pour concevoir des systèmes de contrôle efficace capable de répondre aux exigences spécifique de chaque application industrielle.

CHAPITRE II

Description matérielle et simulation

II.1 Introduction :

Ce chapitre est consacré à la présentation des différents équipements électroniques et les circuits que nous avons mis en œuvre dans le cadre de la réalisation de notre régulateur de température Différentielle. Cette étape est essentielle car elle constitue la base matérielle du système permettant de nous assurer la mesure, le traitement et aussi le contrôle de la différence de température entre deux points de mesure de manière fiable.

L'un des composants essentiels de notre montage est le capteur de température numérique DS18B20, reconnu pour sa précision, sa simplicité d'intégration et sa communication via le protocole 1-Wire ; nous aborderont ses caractéristiques techniques en détail.

Enfin, nous décrivons également les logiciels utilisés pour programmer le microcontrôleur aussi bien que le logiciel de simulation utiliser pour valider le comportement du système avant sa mise en œuvre pratique.

II.2. Présentation globale du régulateur :

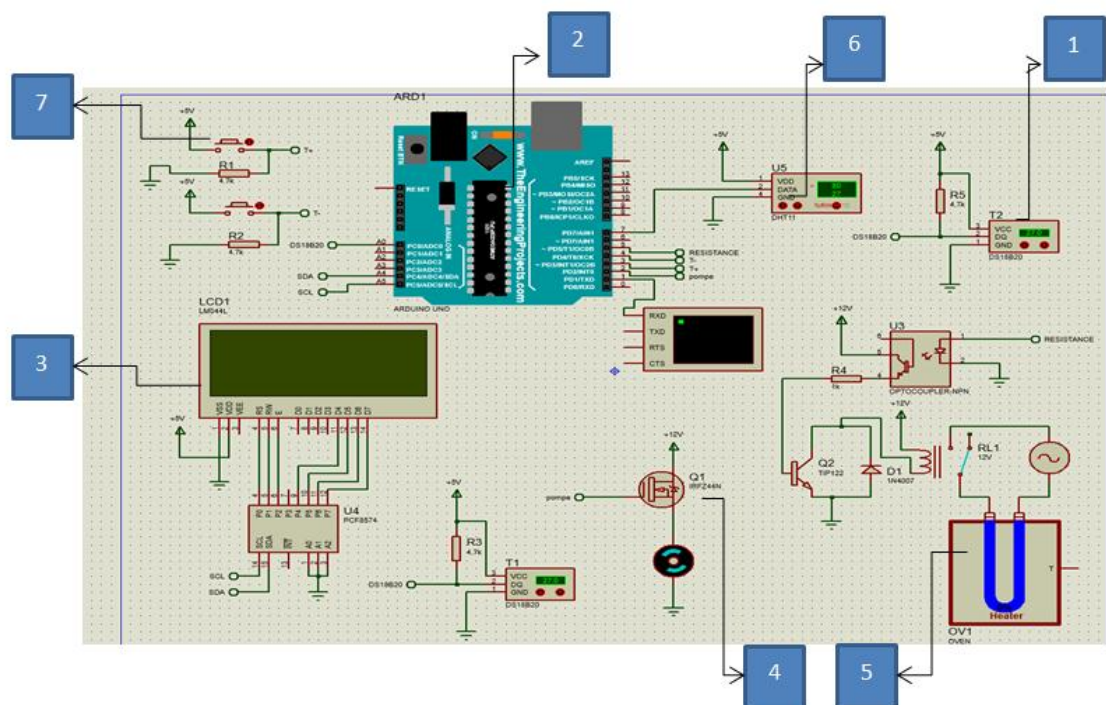


Figure II. 1 : Schéma global du régulateur présenté sur proteus.

CHAPITRE II Description matérielle et simulation

Ce schéma donne la vue complète sur l'ensemble des constituants de notre système de régulation. Et ci-dessous, on présente les différents composants utilisés notre projet :

- 1 Capteur de température DS18B20
- 2 Carte Arduino UNO R3
- 3 Afficheur LCD *LCD 20×4* avec interface I2C
- 4 Pompe
- 5 Plaque chauffante
- 6 Capteur de température dht11
- 7 Boutons poussoirs

II.3 Description des composants utilisés :

II.3.1 Capteur de température DS18B20 :

Le DS18B20 est un capteur de température numérique utilisant une interface 1-Wire dont son étendu de mesure s'étale dans une plage de -55°C à $+125^{\circ}\text{C}$ avec une précision de $\pm 0,5$ degrés Celsius. Ce dernier est un dispositif qui émet un signal numérique grâce à un circuit intégré. Cet équipement dispose aussi, d'une mémoire pour stocker les informations de calibration son utilisation est facile à utiliser et à intégrer dans les projets grâce à son interface de bus 1-Wire, et se distingue par le nombre réduit de fils de raccordement il se distingue au par une aptitude à être reliaer avec plusieurs capteurs sur une même ligne. Il peut être utilisé avec différents microcontrôleurs et de plates- formes de développement, comme Arduino, ESP32, Raspberry Pi, etc [13].



Figure II.2 : Capteur de température DS18B20 [13]

II.3.1.1 Le principe de fonctionnement du DS18B20 :

- **Communication 1-Wire :** La communication se fait via un seul bus 1-Wire, c'est-à-dire une seule ligne de données. Cette même ligne unique sert à la fois à transmettre les informations et dans certains cas à alimenter le capteur évitant une alimentation externe supplémentaire.
- **Adresse unique :** Chaque capteur DS18B20 possède un numéro de série unique de 64 bits gravé en usine. Cette adresse permet de connecter plusieurs capteurs sur le même bus 1-Wire et de les identifier individuellement, ce qui facilite la gestion de réseaux de capteurs sur une seule ligne.
- **Mesure et conversion de la température :** À l'intérieur du DS18B20, un circuit intégré à semi-conducteurs mesure la température ambiante et la convertit en un signal numérique codé sur 12 bits. Cela permet d'obtenir une précision allant jusqu'à $\pm 0,5$ °C sur une plage de mesure étendue, de -55 °C à +125 °C.
- **Séquence de communication :** Pour communiquer avec le DS18B20, le microcontrôleur envoie un signal de réinitialisation, les capteurs répondent par une impulsion de présence ; ensuite le microcontrôleur envoie des commandes spécifiques pour lire l'identifiant du capteur, lancer la conversion de la température, et lire les données converties.
- **Alimentation parasite :** Le DS18B20 peut fonctionner sans avoir besoin d'une alimentation électrique séparée grâce à un système appelé (alimentation parasite), il récupère l'énergie nécessaire à son fonctionnement directement sur la ligne de

donner lors des phases où celle-ci est à l'état haut (c'est à dire quand le signal est actif). Cette énergie est stockée par un condensateur interne pour assurer le bon fonctionnement continu pendant la conversion de température. [14,15]

II.3.1.2. Le branchement du DS18B20 avec Arduino :

La connexion des broches du capteur DS18B20 est effectuée comme suit :

- GND doit être connectée à la masse de l'Arduino ;
- VCC à la broche 3.3V de l'Arduino ;
- DATA à une broche digitale de l'Arduino (par exemple, la broche 2) ;
- l'utilisation d'une, appelé résistance de pull-up de $4,7k\Omega$ entre VCC et la broche DATA du DS18B20. permet de maintenir un état haut sur la ligne de données 1-Wire lorsque le capteur n'est pas en cours d'utilisation [16].

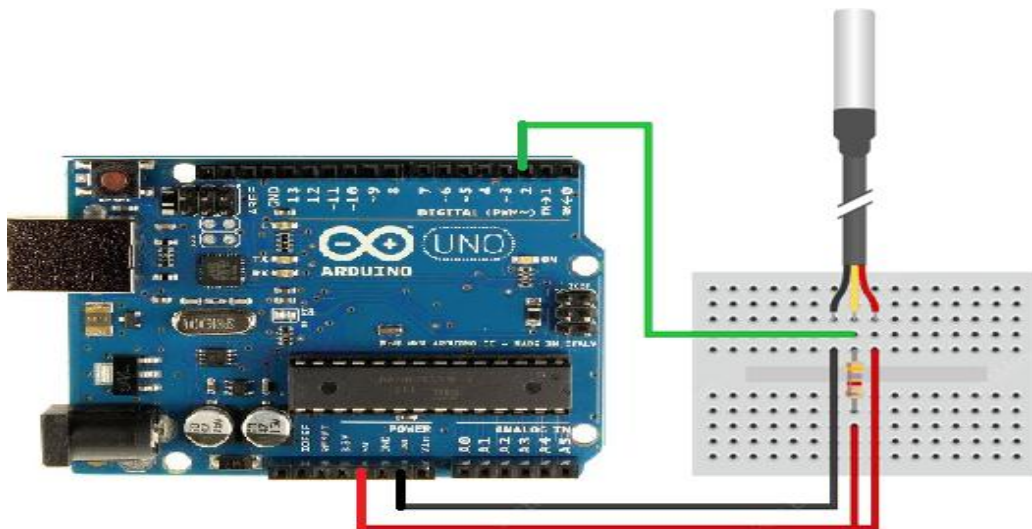


Figure II. 3 Branchement du DS18B20 avec Arduino [16]

II.3.2 Afficheur LCD 20×4 avec interface I2C :

L'afficheur LCD 20×4 est un écran rigide capable d'afficher 20 caractères par ligne sur 4 lignes. Il s'agit uniquement des caractères alphabétique et numérique. Il est à noter aussi que cet équipement ne permet pas d'afficher des graphiques.

A l'arrière on trouve un petit module noir est soudée, intégrant un circuit I2C qui facilite la connexion au microcontrôleur ; ce module comprend un potentiomètre pour régler le contraste de l'écran. Il nécessite seulement 4 connexions : VCC, GND, SDA, SCL, ce qui libère

CHAPITRE II Description matérielle et simulation

des broches d'entrées/sorties sur la carte Arduino. L'éclairage du fond est généralement bleu avec des caractères blancs, et la tension d'alimentation est de 5V.



Figure II. 4 Afficheur LCD 20×4 I2C

II.3.2.1 Principe de fonctionnement :

Comme son nom l'indique, un écran LCD possède des cristaux liquides. Si on regarde de très près sur écran LCD on peut voir une grille de carré. Ces carrés sont appelés des pixels (Élément d'image). Lorsqu'aucun courant ne le traverse, ses molécules sont orientées dans un sens (0°). En revanche lorsqu' un courant le traverse, ses molécules vont se tourner dans la même direction (90°) [17].

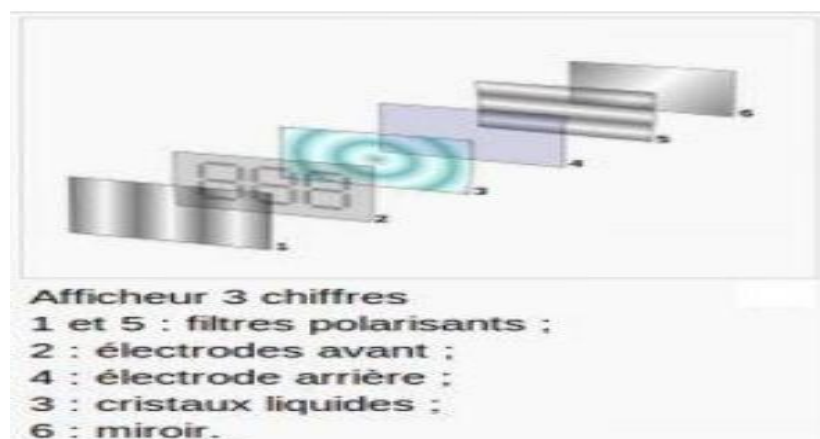


Figure II. 5 Composition d'un écran LCD [17]

CHAPITRE II Description matérielle et simulation

En effet, entre les cristaux liquides et la source lumineuse se trouve un filtre polariseur de lumière. Ce filtre va orienter la lumière dans une direction précise. Entre les cristaux se trouve un autre écran polariseur, qui est perpendiculaire au premier [17].

II.3.2.2 Caractéristiques principales de l'afficheur LCD 20×4 I2C :

Nombre de caractères	20 caractères par lignes
Nombre de lignes	4 lignes
Interface	I2C
Nombre de broches utilisées	4 (VCC, GND, SDA, SCL)
Alimentation	5V
Vitesse de communication	Jusqu'à 400 KHz (mode faste I2C)
Dimensions	98 x 60 x 20 mm
Ajustement de contraste	Potentiomètre intégré

Tableau II. 1 : Caractéristiques principales de l'afficheur LCD

II.3.3 Relais électrique 5VSRD-05VDC-SL-C :

Le module relais 5V est comme un interrupteur contrôlé par un signal électrique. Il peut être utilisé pour allumer ou éteindre un circuit .il est activé par un signal de 5v, ce qui le rend adaptable avec de nombreux microcontrôleurs, il est composé d'un relais et d'un module de Commande. Le relais est composé d'une bobine qui crée un champ magnétique, d'une armature qui bouge pour ouvrir ou fermer le circuit.

Le module de contrôle du relais est une interface. Il possède d'une part des bornes pour être se connecter au microcontrôleur et d'autres part des bornes pour être connecter la charge.

Le relais SRD-05VDC-SL-C peut commuter jusqu'à 10A sous 250V AC ou 30V DC. Sa bobine fonctionne en 5V avec une consommation d'environ 70 mA. Ce relais est robuste, compact et assure une isolation entre la commande et la charge, protégeant ainsi le microcontrôleur [18].



Figure II. 6 Relais électrique 5V [19].

II.3.4 Le capteur de température et d'humidité (DHT11) :

DHT11 est un capteur numérique de température et d'humidité, il contient une sortie de signal numérique calibré de la température et de l'humidité, il comprend un sens résistif de composants humides et un dispositif de mesure de température, et connecté avec un microcontrôleur 8 bits haute performance. Cet équipement présente l'inconvénient de délivrer des données que sur un pas de deux second, alors lorsqu'on déclare la bibliothèque de ce capteur dans le programme, la lecture de ce capteur peut durer jusqu'à 2 secondes, C'est ce qui est déclaré dans la bibliothèque de ce capteur [20].

L'étendu de mesure de cet équipement s'étale entre 0 à +50°C avec une précision de +/- 2°C alors que le taux d'humidités relative soit de 20 à 80 % avec une précision de +/- 5%.

L'alimentation du DHT11 peut varier entre 3.3 volts et 5 volts (le fabricant recommande cependant de toujours alimenter le capteur en 5 volts pour avoir des mesures précises) [21].

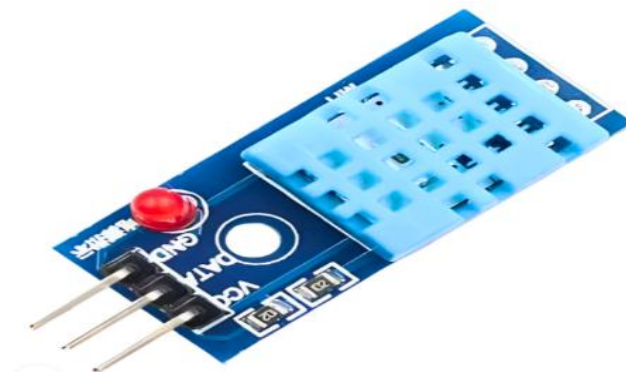


Figure II. 7 Le Capteur DHT11 [20]

II.3.5. La Carte Arduino :

La carte à microcontrôleur Arduino (ou son tout récent synonyme Genuino) est une carte électronique qui intègre un microcontrôleur d'architecture Atmel AVR (comme ATmega328 ou ATmega2560 pour les versions récentes ATmega168 ou ATmega8 pour les plus anciennes), et des composants complémentaires qui permettent la programmation et l'interfaçage avec d'autres circuits. Cette carte est une plate-forme à entrée/sortie simple qui est destinée à la programmation interactive qui peut être utilisée pour communiquer avec des logiciels tels que le Proteus lorsqu'elle est connectée à un ordinateur [22].

II.3.5.1. Types de cartes Arduino :

Il existe plusieurs types de cartes Arduino. Parmi les plus utilisées dans le monde des systèmes embarqués on a : la UNO, la Nano, la LEONARDO, la DUE, la MEGA et sa petite amélioration la MEGA 2560, etc....

Sur le tableau qui suit sont présentées les caractéristiques des cartes ARDUINO :

Arduino	Microcontrôleur	Flash ko	EEPROM ko	ko SRAM ko	Broches d'E/S numériques	Broches d'entrée analogique	Vitesse du Processeur
UNO	ATmega328P	32	1	2	14	6	16 MHZ
NANO	ATmega328P	32	1	2	14	8	16 MHZ
LEONARDO	ATmega32U4	32	1	2.5	20	12	16 MHZ
DUE	Atmel SAM3X8E	512	0	96	54	12	84 MHZ
MEGA	ATmega1280	128	4	8	54	16	16 MHZ
MEGA 2560	ATmega2560	256	4	8	54	16	16 MHZ

Tableau II. 2 : Quelques cartes Arduino et leurs caractéristiques [26].

Notre choix s'est porté sur la carte UNO, car ses caractéristiques sont parfaitement adaptées au fonctionnement de notre système. Avec ses entrées/sorties suffisantes et sa capacité de traitement adéquate, elle permet de gérer efficacement les capteurs et actionneurs du

CHAPITRE II Description matérielle et simulation

système. De plus son prix reste Intéressant, (environ 2500 dinar) pour la dernière version d'Arduino UNO.

II.3.5.2. Arduino UNO (R3) :

La carte UNO est une énorme option pour l'Arduino initial. Elle se compose de 14 broches d'entrées/sorties numériques, où 6 broches peuvent être utilisées en tant que sorties de modulation de largeur d'impulsion (PWM), 6 entrées analogiques, un bouton de réinitialisation, une prise d'alimentation et une connexion USB.

Il comprend aussi tout ce qui est nécessaire pour tenir le microcontrôleur en place. Elle utilise un microcontrôleur AVR d'architecture ATmega328p cadencé à 16Mhz. Elle possède 32ko de mémoire flash destinée à recevoir le programme, 2ko de SRAM (mémoire vive) et 1 ko d'EEPROM (mémoire morte destinée aux données) [23].

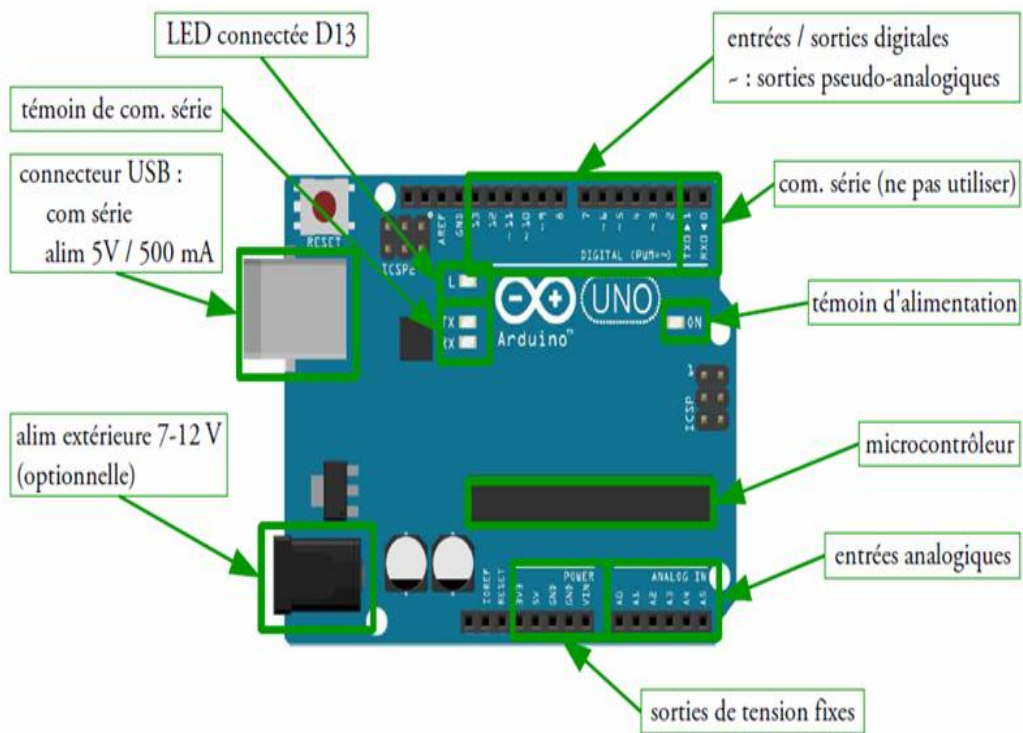


Figure II. 8 Arduino UNO (R3) [23]

II.3.6 Bouton Poussoir :

Un bouton poussoir est un dispositif qui permet de générer une action lorsqu'on appuie dessus. Il revient directement à sa position initiale dès qu'on le relâche.

II.4 Capteur solaire thermique :

Un capteur solaire thermique est un dispositif conçu pour recueillir l'énergie solaire transmise par rayonnement et la transférer à un fluide caloporteur (gaz ou liquide) sous forme de chaleur. Cette énergie thermique peut ensuite être récupérée dans un réservoir de stockage pour être utilisée comme eau-chaude sanitaire, pour le chauffage de bâtiments, ou pour être utilisée encore dans divers procédés industriels. Cette technologie est différente de celle des panneaux photovoltaïques, qui transforment la lumière (les photons) en électricité. Ainsi ces deux technologies peuvent toutefois être combinées dans un même panneau appelé PVT pour produire au même temps les deux énergies photovoltaïques et thermiques [25].



Figure II. 9 Capteur solaire thermique à tubes sous vide (caloduc) [24].

II.4.1 Fonctionnement d'un capteur solaire thermique :

Bien que l'on distingue plusieurs technologies de panneaux, suivant les capteurs installés, le fonctionnement global reste le même :

1. Collecte de la chaleur du soleil par des capteurs solaires grâce à un élément principale appelé absorbeur ;
2. Transfert de la chaleur de l'absorbeur vers le fluide caloporteur à travers le circuit primaire vers le réservoir de stockage ;
3. Stockage de l'eau chauffée dans un ballon de stockage
4. à partir du fond bas de la cuve de stockage, l'eau, la plus froide rejoint la partie basse du Capteur et remonte à travers le radiateur pour absorber l'énergie convertie par l'absorbeur et rejoint la partie haute du réservoir de stockage. Ainsi, lorsqu'il s'agit

d'un système de chauffe-eau individuel la circulation du fluide caloporteur entre le capteur et le réservoir de stockage se fait naturellement par le principe thermosiphon. Alors que quand le capteur et la cuve de stockage sont séparés, l'utilisation de pompe s'impose.

5. Restitution de la chaleur du réservoir de stockage via un circuit secondaire dédié à l'utilisateur. IL est à noter que l'utilisation du système de régulation s'impose que si le capteur est séparé de la cuve de stockage.

1. Les capteurs tubulaires sous vide :

Nettement plus performants, mais aussi beaucoup plus chers et plus fragiles, ces capteurs sont composés de tubes en verre sous vide d'air, ce qui diminue les pertes par convection et ceux par rayonnement de grande longueur d'onde conséquent de l'échauffement de l'absorbeur. Ces capteurs qui permettent d'atteindre des températures élevées pouvant atteindre les 120°C, ils sont considérés comme l'avenir du solaire thermique, compte tenu de leur rendement important.



Figure II.10 Capteur tubulaire sous vide [24].

2. Capteur solaire plan vitré :

Les panneaux les plus courants sont constitués d'un capteur « plan vitré ». Vus de l'extérieur, ils peuvent parfois assembler à un module photovoltaïque Leur architecture est pourtant très différente :

- Le panneau est composé d'un coffret en acier ou aluminium sur lequel est posée une couverture transparente fabriquée généralement en verre trempé d'une épaisseur de 3 mm à 4mm. Le rôle de la couverture est de protéger le capteur des intempéries.

- Un fluide caloporteur circule dans un serpentin installé dans l'espace, sous vide d'air ou non, situé entre la vitre et la feuille. Il absorbe la chaleur du soleil. Le serpentin est parfois bordé d'ailettes noires, ce qui permet de maximiser l'absorption de chaleur. Un matériau isolant thermique est placé au dos du serpentin pour réduire les pertes de chaleur. Par sa robustesse, sa facilité d'intégration, son rapport coût/performance et sa capacité à produire une eau entre 50 et 90°C, les panneaux à capteurs plans vitrés sont les plus populaires en utilisation domestique.



Figure II.11 Capteur solaire plan vitré [24]

Notre choix s'est porté sur le capteur solaire plan vitré, car il offre un excellent compromis entre performance, coût et durabilité, tout en assurant une production efficace d'eau chaude sanitaire grâce à sa capacité à capter et conserver la chaleur de manière optimale.

II.5 Les différents types d'installation :

La plupart des installations sont conçues avec une boucle fermée sous pression dont la circulation est forcée, mais il existe d'autres types d'installation :

- **Installation à boucle solaire Fermée (indirecte), boucle à circulation forcée :** la boucle appelée boucle primaire. Dans cette boucle, le capteur est relié à un échangeur de chaleur plongé dans le réservoir de stockage. Ce dernier est rempli par le second circuit d'eau dont l'entrée est reliée au réseau publique. Une fois chauffé par l'échange thermique via l'échangeur de chaleur, l'eau chaude est acheminée vers son utilisation, appelés points de puisages (douches...) [24].

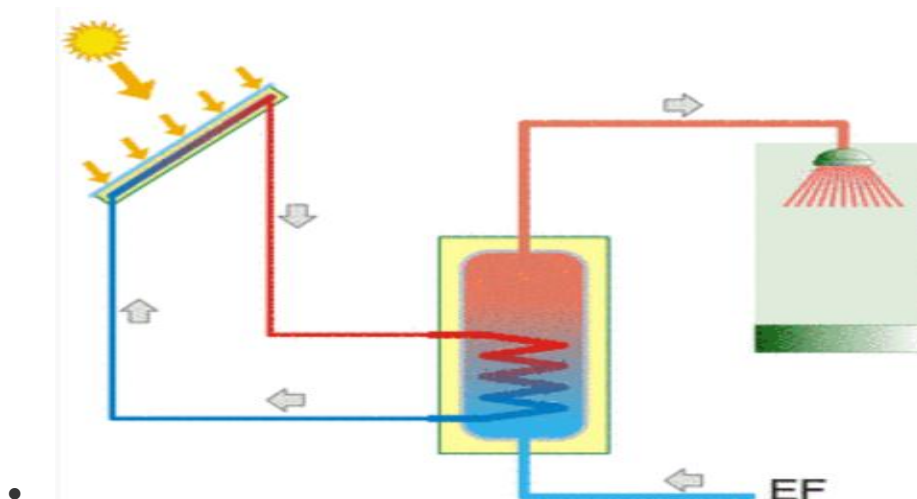


Figure II. 12 Boucle solaire fermée Circulation forcée [11].

- **Boucle solaire ouverte (directe) :** Dans le cas où la boucle est dite ouverte, l'eau qui circule dans les capteurs est la même que celle qui est consommée aux points de puisage. On le trouve donc plus souvent dans les pays chauds, où les capteurs constituent le seul moyen de chauffage.

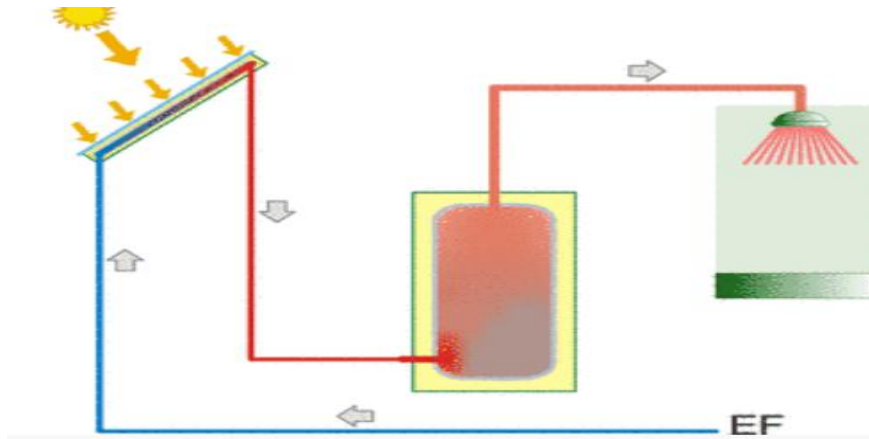


Figure II. 13 Boucle ouverte [11]

- **Installation à circulation naturelle (principe de circulation à thermosiphon) :** Dans les installations à thermosiphon, le fluide de la boucle solaire circule effet naturel où, de plus en plus que fluide s'échauffe, sa densité diminue et les particules les plus chaude tendent à rejoindre les parties les plus hautes du radiateur jusqu'à rejoindre le réservoir de stockage

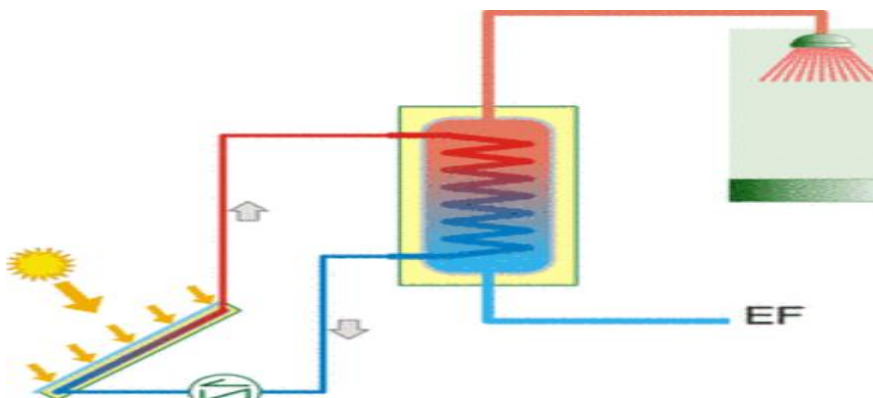


Figure II. 14 Circulation thermosiphon

Notre choix s'est porté sur un capteur solaire plan vitré à une installation en boucle fermée avec une circulation forcée pour plusieurs raisons, car c'est ce type d'installation qui est présent au Centre de Développement des Énergies Renouvelables (CDER) de Bouzaréah, où

nous avons effectué notre stage. Ce choix nous a permis de travailler sur un système réel et de mettre en pratique nos connaissances directement sur l'équipement disponible dans le centre.

Ainsi notre travail consiste à installer de sondes de température dont l'une permet de mesurer la température de l'eau dans le réservoir de stockage et l'autre à la sortie du capteur. Ainsi ceci nous permettra commander la mise en marche ou l'arrêt de la pompe de circulation selon la différence de température entre les valeurs mesurées par les deux sondes. Par ailleurs, généralement durant les périodes creuses, ou l'eau contenue dans le réservoir n'est pas suffisamment chauffée, un système d'appoint en matière de chauffage s'impose, il s'agit un thermoplongeur.

De même une régulation de type TOR est utilisée pour la mise en marche ou l'arrêt de la résistance de chauffage plongée dans le réservoir de stockage.

II.6. Partie logicielle :

II.6.1. Présentation du logiciel utilisé :

Aujourd'hui, grâce à l'utilisation combinée de logiciels et de matériels, il est possible de créer des interfaces complexes et performantes de gestion de système ou d'un environnement spécifique. Ainsi, Ces outils, permettent de concevoir, réguler, commander et réaliser des systèmes adaptés aux besoins variés de l'industrie manufacturière.

Bien que cette approche soit relativement récente, elle connaît une évaluation rapide, donnant naissance à de nouvelles méthodes de conception et pilotage. L'un de ses principaux atouts réside dans l'interaction fluide entre les différents modules qui composent un système global.

Parmi les solutions accessibles et efficaces, l'utilisation conjointe d'**Arduino** et de **Proteus** se démarque. Cette combinaison représente une alternative innovante et peu coûteuse, parfaitement adaptée au prototypage et l'expérimentation.

II.6.2 L'environnement Arduino :

Les créateurs d'Arduino ont développé un logiciel pour que la programmation des cartes Arduino soit visuelle, simple et complète à la fois. C'est ce que l'on appelle une IDE, qui signifie Integrated Development Environment (EDI).

L'IDE affiche une fenêtre graphique qui contient un éditeur de texte et tous les outils nécessaires à l'activité de programmation. Ce qui permet de saisir, enregistré, compiler, vérifier, transférer le programme sur une carte Arduino.

➤ Présentation de l'interface :

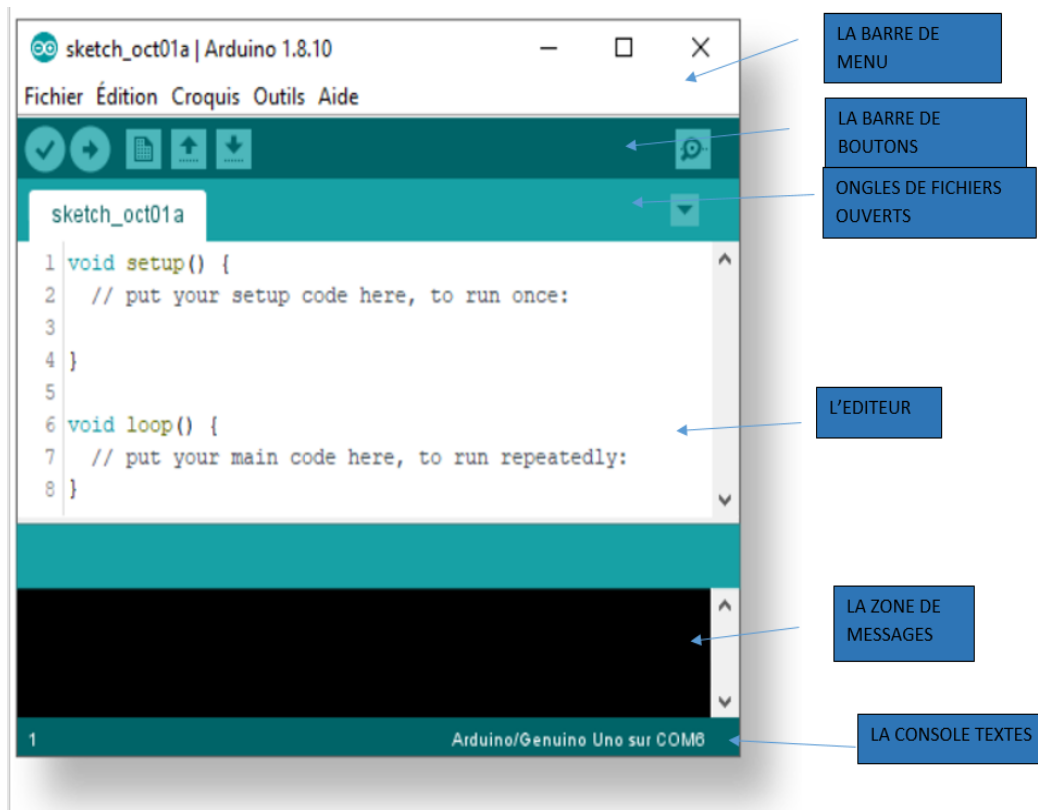


Figure II. 15 Interface de logiciel Arduino IDE

Le logiciel Arduino comporte :

- **La BARRE DE MENUS** : C'est une interface graphique présente dans tous les logiciels.
- **La BARRE DE BOUTONS** : Elle simplifie un accès direct aux fonctions essentielles du logiciel
- **L'ÉDITEUR** : Permet d'écrire un code de programme, avec ongles de navigation.
- **La ZONE DE MESSAGES** : Elle affiche et indique l'état des actions en cours.
- **La CONSOLE TEXTE** : Elle affiche le résultat de compilation.

Barre de menu du logiciel Arduino :



Figure II. 16 Les boutons de l'IDE

	Tache à exécuter
Bouton 1	Permet de compiler le programme, tout en cherchant les erreurs.
Bouton 2	Permet le transfert du programme dans la carte Arduino.
Bouton 3	Crée un nouveau fichier.
Bouton 4	Ouvrir un fichier Arduino existant
Bouton 5	Enregistrement des fichiers.
Bouton 6	Pour envoyer et recevoir des données de l'Arduino au Pc a des fins de débogage.

II.6.3 Structure d'un programme Arduino :

Le langage de programmation utilisé avec Arduino est assez simple en matière de structure. Il repose principalement sur deux parties essentielles, organisées autour de fonctions encadrées par des blocs d'instructions.

On retrouve tout d'abord la fonction **setup**, qui joue un rôle de préparation. Cette partie du code s'exécute une seule fois, dès que la carte Arduino est mise sous tension. C'est ici que l'on initialise les paramètres, comme les broches d'entrée /sortie ou la communication série, si celle-ci est utilisée.

Ensuite, la fonction **Loop** qui engendre les instructions qui fonctionnent en continu tant que la carte reste alimentée.

Il est important de noter que toute variable utilisée doit être déclarée en début de programme. Quant aux broches disponibles sur la carte (le nombre varie selon le modèle), elles peuvent être configurées soit comme entrées, soit comme sorties, généralement dans la section **setup**.

II.6.4 Proteus (ISIS et ARES) :

Proteus est une suite logicielle destinée à des applications l'électronique, développé par la société Labcenter Electronics, les logiciels incluent dans Proteus permettent la CAO dans le domaine électronique.

➤ **Présentation générale :**

Cette suite logicielle est très connue dans le domaine de l'électronique. De nombreuses entreprises et organismes de formation (incluant lycée et université) utilisent cette suite logicielle. Outre la popularité de l'outil, Proteus possède d'autres avantages

- Pack contenant des logiciels facile et rapide à comprendre et utiliser ;
- Son support technique est performant ;
- L'outil de création de prototype virtuel permet de réduire les coûts matériel et logiciel lors de la conception d'un projet ;

➤ **ISIS :**

L'interface ISIS de Proteus est principalement connu pour éditer des schémas électriques. Par ailleurs, le logiciel permet également de compiler ces schémas électriques ce qui permet de déceler les erreurs engendrées dans le schéma considéré dès l'étape de conception.

➤ **ARES :**

Le logiciel ARES est un outil d'édition et de routage qui complète parfaitement ISIS. Un schéma électrique réalisé sur ISIS peut alors être importé facilement sur ARES pour réaliser le PCB de la carte électronique. Ainsi ce logiciel permet d'optimiser le routage et configure l'architecture du PCB en proposant l'emplacement des différents composants et de réaliser le routage automatiquement.

➤ **Présentation de l'interface :**

L'interface du logiciel Proteus est présentée comme suit :

Elle est constituée par :

- Barre de menus ;
- Barre d'outils de commande ;

CHAPITRE II Description matérielle et simulation

- Barre d'outils de sélection de mode ;
- Barre d'outils d'orientation ;

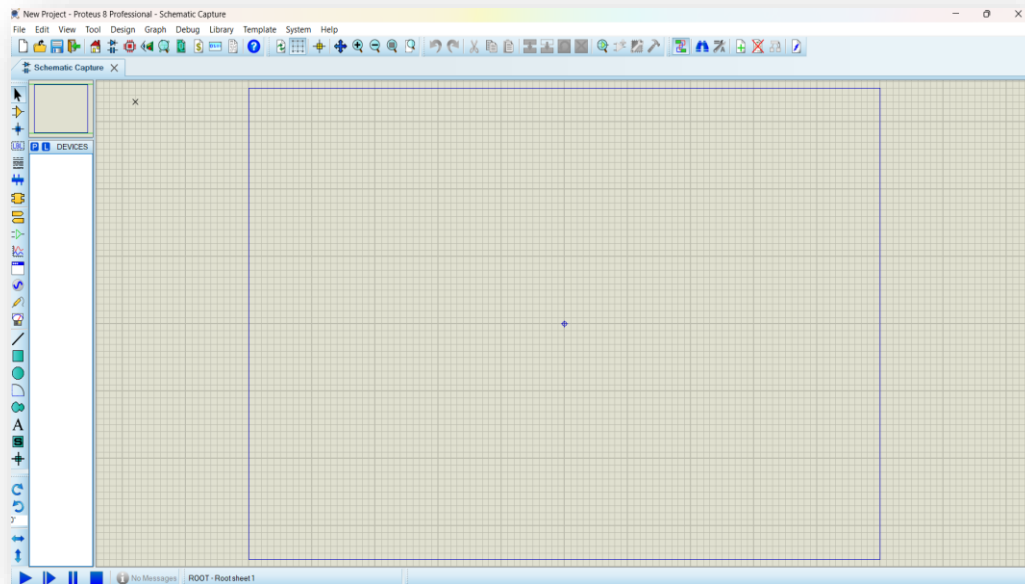


Figure II. 17 Interface Proteus

➤ **Barre de menus :**

File Edit View Tool Design Graph Debug Library Template System Help

Cette barre permet de contrôler les actions, telle que l'ouverture, l'enregistrement, l'impression et le mode d'affichage.

➤ **Barre d'outils de commande :**

CHAPITRE II Description matérielle et simulation

Ces barres offrent un accès similaire aux commandes des menus et peuvent être cachées via l'option « Barre d'outils » dans le menu « Affichage ».



Commandes Fiches / Projet.



Commande d'affichage.



Commandes édition / Bibliothèque.



Commandes Outils.

➤ Barre d'outils de sélection de mode :

Elle permet de sélectionner un outil parmi les 3 modes d'édition disponibles.



Mode principal.



Mode gadgets.



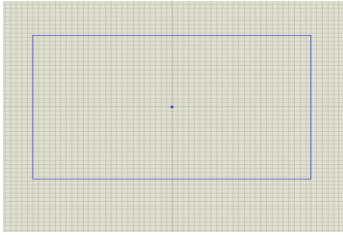
Mode graphique.

➤ Barre d'outils d'orientation :



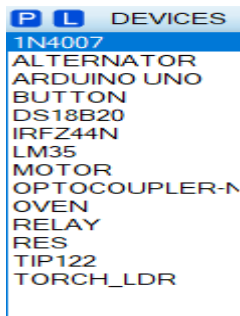
Elle permet de contrôler et d'afficher la rotation et la réflexion d'un objet placé ou à placer.

➤ Zone de travail ou d'édition des schémas :



Cette zone rectangulaire, c'est là où le travail prend vie tous les dessins et schémas s'afficheront ici, et on peut choisir la taille grâce aux options zoom.

➤ Sélecteur d'objets :



Le sélecteur d'objets liste les différents éléments. On y trouve des composants, des connexions, des symboles, des repères et même des graphiques.

II.7 Conclusion :

Nous avons divisé ce chapitre en deux parties. La première partie est consacrée à la description matérielle que nous avons utilisée dans la réalisation de notre maquette. Nous nous sommes particulièrement intéressés à décrire dans le détail le capteur de température que nous avons utilisée en l'occurrence le DS18B20.

Tandis que la deuxième partie est consacrée à la présentation des différents logiciels que nous avons utilisé pour notre simulation.

Chapitre III

Tests de validation et résultats expérimentaux

III.1. Introduction :

Dans le cadre de notre projet, nous avons conçu et testé un système de régulation de la température de l'eau dans un réservoir, basé sur des capteurs, des actionneurs et une carte Arduino. L'objectif est d'assurer une gestion automatisée de la chaleur produite par un chauffe-eau solaire et une résistance électrique, selon les mesures relevées par les capteurs de température.

Ce chapitre présente les principales étapes de la mise en œuvre du projet : la simulation du système sous Proteus, la réalisation pratique sur maquette d'essai, ainsi que les tests fonctionnels et l'analyse du comportement du régulateur.

III.2. Simulation et premiers tests :

Cette partie regroupe les étapes clés de la simulation logicielle et les difficultés rencontrées lors de la simulation du régulateur de température.

Avant de passer à la simulation de notre système nous avons d'abord testé le capteur principal de ce régulateur.

III.1.1. Simulation sur Proteus avec DS18B20 :

Les bibliothèques de la carte Arduino R3 et du capteur DS18B20 doivent être téléchargées et ajoutées dans la bibliothèque du logiciel Proteus Isis ; voici le branchement du capteur sur Arduino sur Proteus :

- La broche VCC du capteur va sur celle de l'alimentation 5V de l'Arduino ;
- La broche GND du capteur va sur celle représentative de la masse GND de l'Arduino ;
- La broche DQ du capteur est reliée à la broche (PIN A1) de l'Arduino ;
- La résistance de 4,7 k Ω est placée entre la broche DQ du DS18B20 et la broche d'alimentation 5V de l'Arduino.

La figure suivante montre le montage que nous avons réalisé sur Proteus pour l'essai de mise en marche du capteur DS18B20 connecté à l'Arduino :

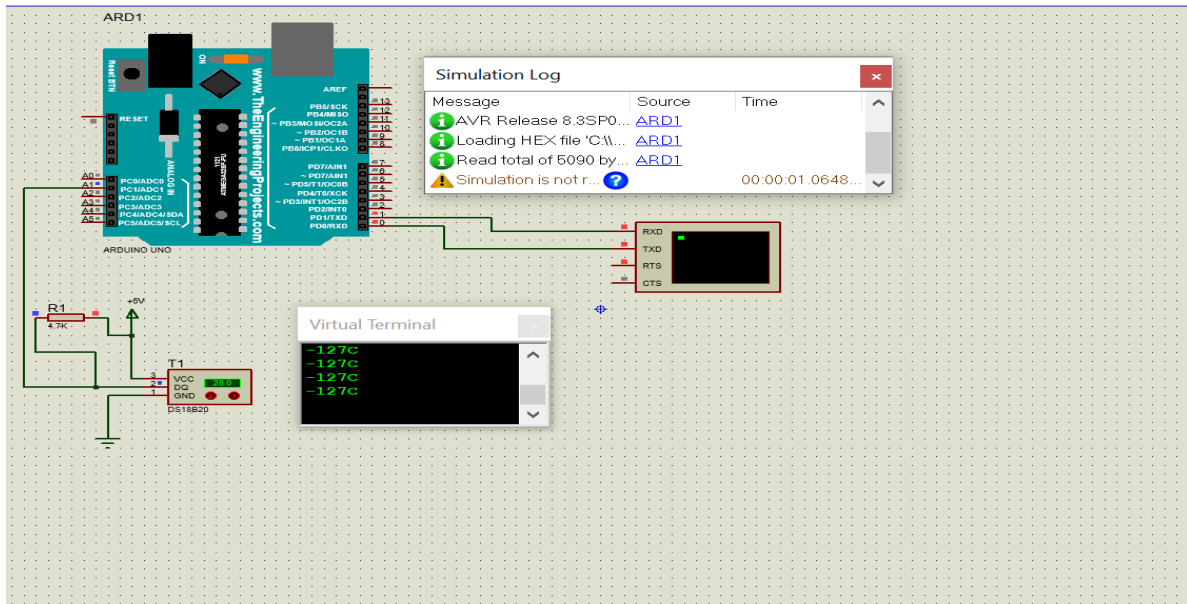


Figure III. 1 : Visualisation de l’erreur du capteur DS28B20 simulé sous Proteus

Lors de la compilation du programme Proteus pour vérifier le bon fonctionnement du capteur DS18B20, nous avons constaté que le programme affichait systématiquement une valeur de -127°C au lieu de 22°C et afin de s’assurer du résultat, l’essai est répété plusieurs fois. Ce qui signifie qu’il s’agit d’un dysfonctionnement ou une incompatibilité du DS18B20 avec la simulation en matière de timing dans l’environnements de simulation.

Ce problème nous a amenés à envisager d’utiliser un autre capteur pour poursuivre notre test virtuel, il s’agit du capteur de température LM35. Ceci est due à notre sens à l’compatibilité ou de timing dans les environnements de simulation

III.1.2. Simulations sur Proteus avec LM35 :

Nous avons ainsi remplacé le DS18B20 par le capteur LM35, pour continuer nos essais en remplaçant ce dernier par un capteur LM35.

Ainsi, voici le branchement du LM35 s’effectue de la manière suivante :

- La broche VCC du capteur est reliée à celle d’alimentation VCC de 5V ;
- La broche GND est reliée à la masse GND
- La broche Vout est connectée vers une des broches d’entrée de l’Arduino (PIN A1)

Sur la figure La figure ci-dessous est présenté le montage que nous avons réalisé sur Proteus:

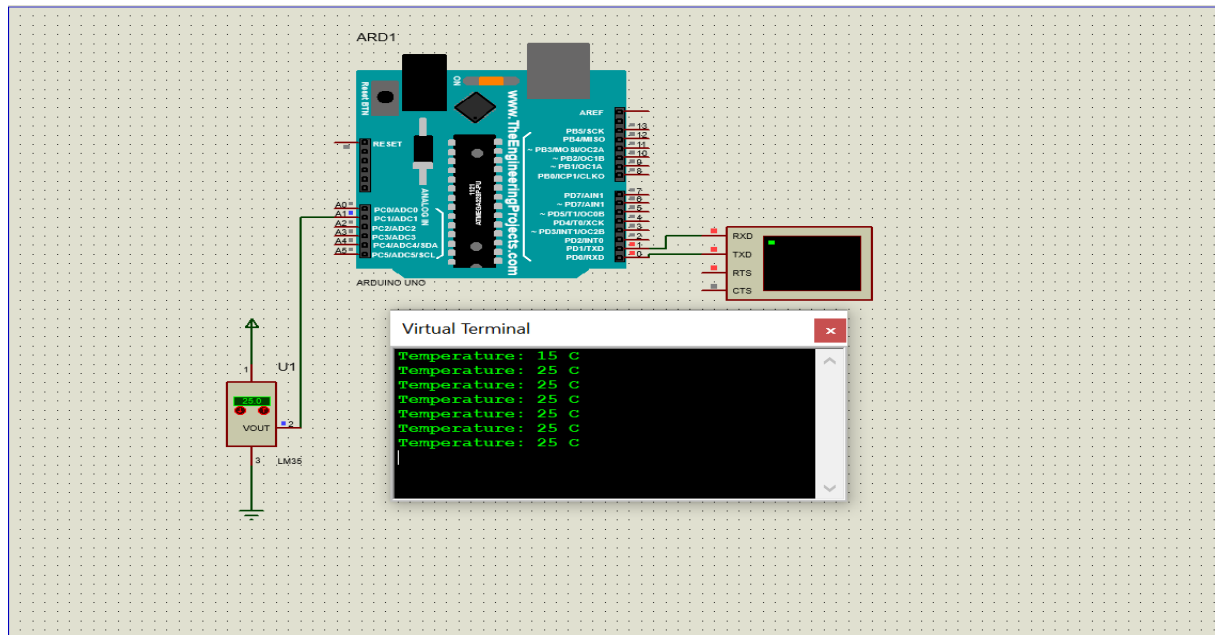


Figure III. 2 Visualisation de l'efficacité du capteur LM35 simulé sur Proteus

Comme résultats, la valeur de la température affichée correspond à sa valeur réaliste, ce qui nous a permis d'utiliser le LM35 pour le reste de nos tests de simulation.

III.1.3. Mise en œuvre du régulateur complet en simulation :

Pour mettre en œuvre le fonctionnement du régulateur sous le logiciel Proteus, nous avons commencé par assembler l'ensemble des composants nécessaires dans l'environnement Proteus qui sont :

- Carte Arduino UNO R3
- Capteur de température LM35
- Afficheur LCD 20×04 I2C
- Pompe
- Plaque chauffante
- Boutons poussoirs
- Capteur de température DHT11

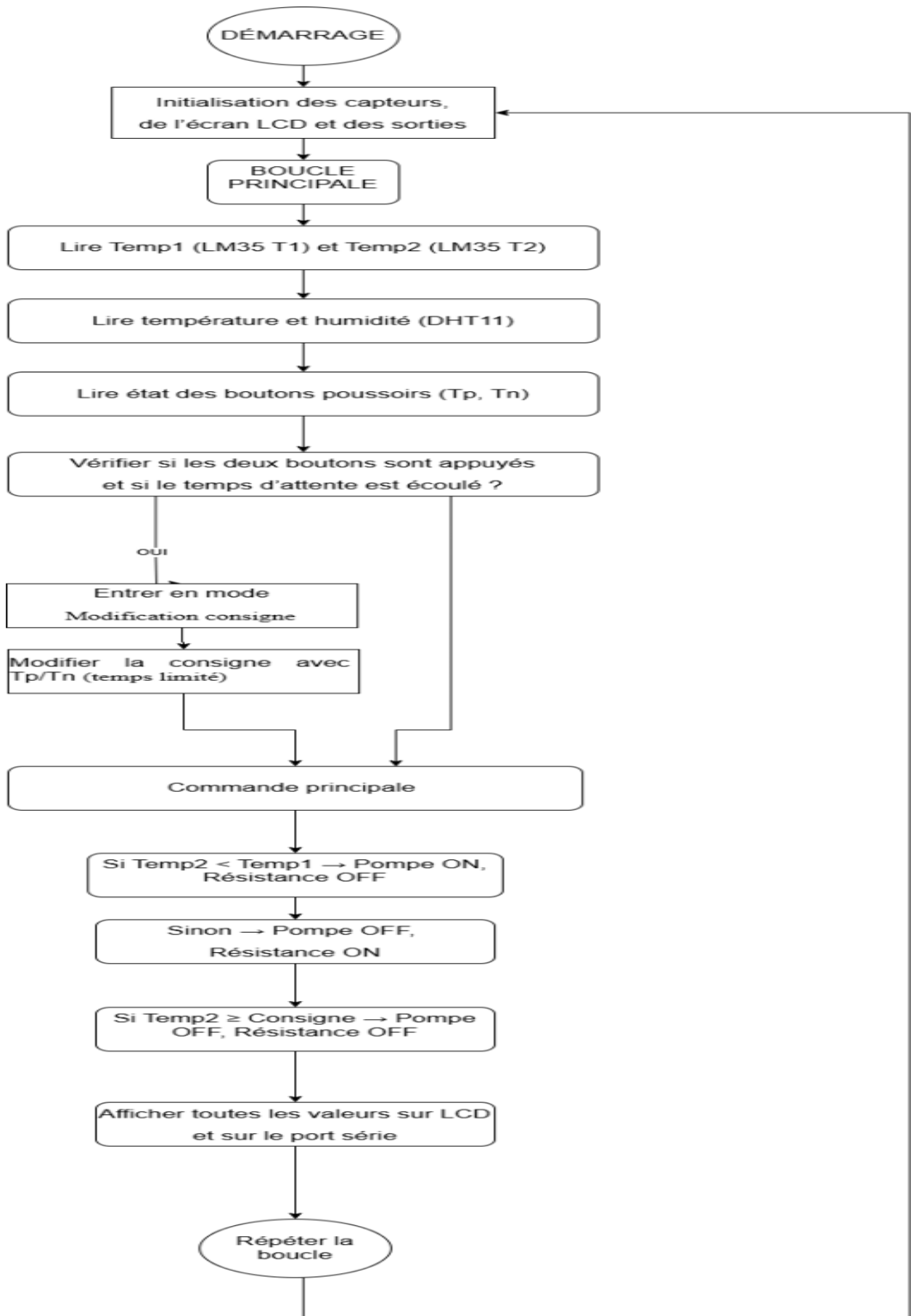
Chapitre III Tests et validation et résultats expérimentaux

En effet, Le fonctionnement du régulateur est assuré par l'ensemble des instructions définies dans le programme.

Les tests ont été réalisés en faisant varier la température des deux capteurs LM35 moyennant une plaque chauffante ce qui nous a permis d'observer la réactivité du système : la régulation s'effectuait correctement, la pompe s'active ou s'éteint selon la différence de température en les deux sondes. Et d'autre part, la résistance chauffante aussi s'active et se désactive selon la différence entre la valeur de la consigne et la valeur mesurée. Cette simulation a nous permis de validé le fonctionnement du régulateur avant qu'on passe à la réalisation matérielle.

III.1.4. Schéma bloc du fonctionnement du régulateur de température :

Sur la figure ci-dessous est présenté le schéma bloc du fonctionnement du système :



III.2. Réalisation de la maquette d'essai :

III.2.1. Présentation de la LAB d'essai :

C'est une installation où sont réalisés des tests et des analyses sur des matériaux, des composants ou des produits afin d'évaluer leurs propriétés qualité et leur conformité aux normes techniques. Ces essais permettent de garantir la fiabilité, la sécurité et la performance des matériaux ou produits à tester.

L'objectif d'un Laboratoire d'essai c'est de vérifier la qualité des travaux et des matériaux utilisés et de voir le comportement des matériaux face à diverses sollicitations (thermiques, mécaniques).

III.2.2. Liste des matériaux utilisés sur notre montage :

Pour la réalisation du montage sur le laboratoire d'essai, on a utilisé les mêmes composants que dans la simulation, à l'exception du capteur de température, en simulation on a choisi le LM35 car il est bien supporté par le logiciel ; pour la partie pratique, on le remplace par le DS18B20 qui offre une meilleure précision et une communication numérique plus fiable, ce qui le rend plus adapté à une utilisation réelle. On a aussi utilisé deux modules relais 5V pour actionner la pompe et la résistance chauffante ; ainsi l'ensemble des matériaux utilisés sur la LAB d'essai :

- 1x Carte Arduino.
- 2x Capteur de température DS18B20.
- 1x Capteur de température et d'humidité DHT11.
- 1x Ecran LCD 20x04 I2C.
- 2x Bouton poussoirs.
- 2x Résistance 4.7 Ω .
- 2x Modules Relais 5V.
- 1x Breadboard
- 20x Fils dupont

Sur la figure ci-dessous est présenté le branchement final réalisé sur le laboratoire d'essai :

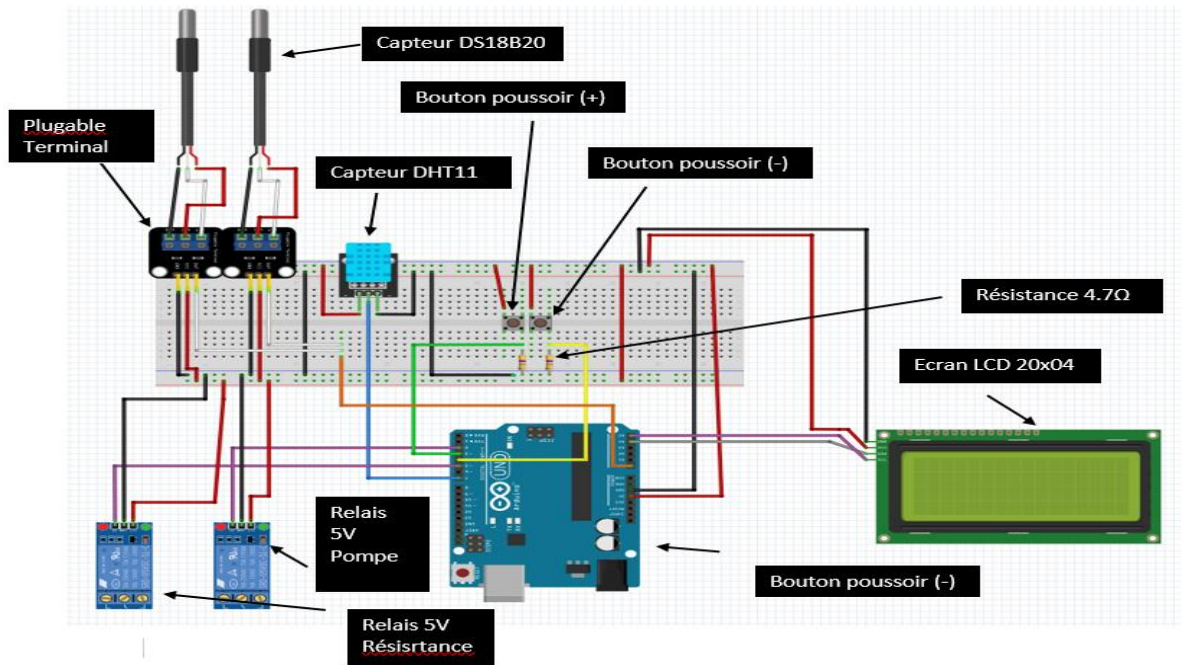


Figure III. 4 Branchement des composants

- Premièrement on va brancher la carte Arduino a la LAB d'essai , il faut relier la broche 5V de la carte a la ligne d'alimentation positive de la breadboard , et relier broche GND de l'Arduino a la ligne négative de la breadboard ; ce qui permet de distribuer la tension 5V et la masse a l'ensemble des composants du montage via la LAB d'essai.
- Puis on va entamer le branchement des deux capteurs DS18B20, ce dernier est déjà équipée d'une résistance pull-up 4,7 kΩ ce qui simplifie le branchement du capteur ; sur le schéma on voit qu'on a relié la broche VCC de chaque capteur a la ligne d'alimentation 5V avec un câble rouge , et la broche GND a la ligne négative (GND) avec un câble noir ; puis les broches de données des deux capteurs on les a relier avec deux câble blanc sur les deux PIN 20 (C) et 20(B) de la breadboard pour les brancher

ensemble sur la même entrée de l'Arduino qui est la PIN A0 avec un câble orange ; ce branchement est possible grâce au protocole 1-Wire de DS18B20 qui permet de connecter plusieurs capteurs sur une seule ligne de données .

- Pour le branchement du capteur DHT11 il se fait en trois connexions : la broche VCC du capteur est reliée à la ligne d'alimentation 5V de la breadboard avec un câble rouge, la broche GND du capteur est reliée à la ligne négative de la breadboard avec un câble noire, et la broche DATA du capteur est reliée à la PIN numéro 7 de l'Arduino.
 - Par rapport aux deux modules relais 5V, pour chaque module on relie sa broche VCC a la ligne positive 5V de la breadboard avec deux câble rouge et sa broche GND a la ligne négative de la breadboard (la masse) avec deux câbles noirs , et la broche de commande IN se relie a l'Arduino (la PIN 5 pour la résistance chauffante et la PIN 2 pour la pompe) avec deux câbles violet .
 - Ensuite, pour ce qui de l'afficheur LCD 20×04 I2C on relie la broche VCC a la ligne d'alimentation 5V de la breadboard avec un câbles rouge, et la broche GND a la ligne négative avec un câble noir , la broche SDA de l'afficheur a la PIN A4 de l'Arduino avec un câble gris et la broche SCL de l'afficheur a la PIN A5 de l'Arduino avec un câble violet .
 - Les deux boutons poussoirs :
 - Bouton (+) :
 - Une pate se relie à la ligne de l'alimentation 5V
 - L'autre pate a la PIN 3 de l'Arduino avec un câble vert.
 - Bouton (-) :
 - Une pate se relie à la ligne d'alimentation 5V
 - L'autre pâte à la PIN 4 de l'Arduino avec un câble jaune.
- Et entre chaque bouton et la masse on introduit une résistance de 4,7 K Ω

III.2.3. Fonctionnement général du système :

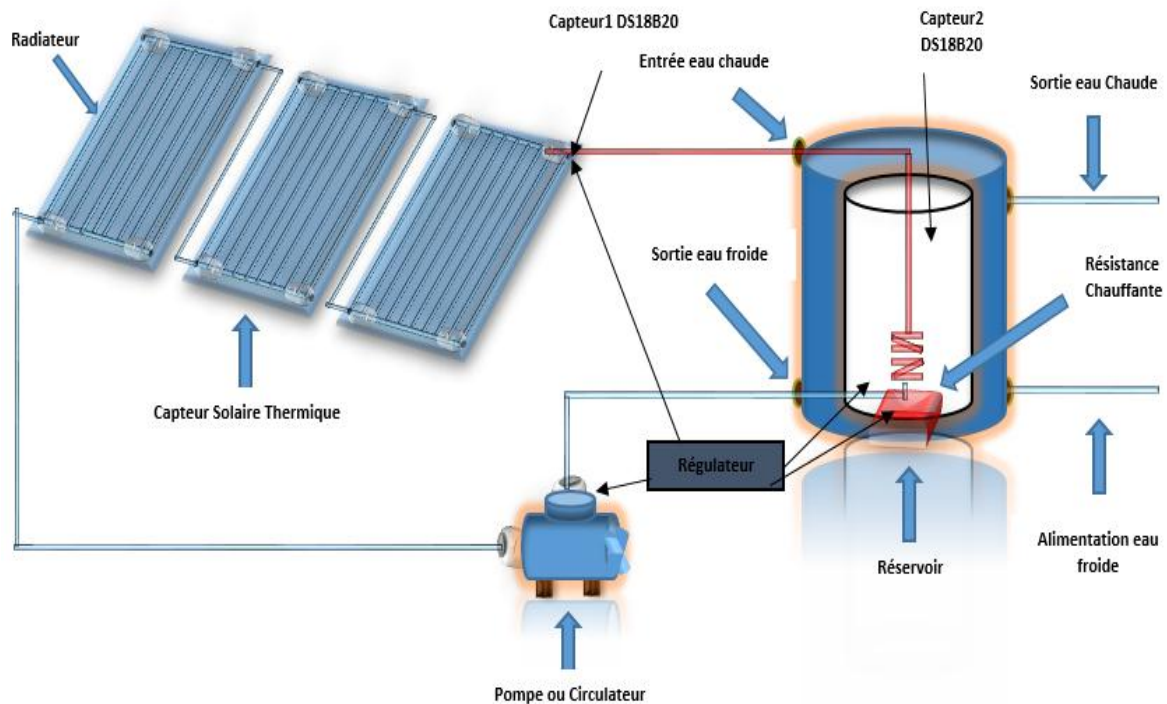


Figure III. 5 Installation d'un chauffe-eau solaire

Notre système consiste à réguler automatiquement la température de l'eau dans un réservoir en combinant l'énergie solaire, grâce à un chauffe-eau solaire et un thermoplongeur.

L'installation contient un ensemble de capteur solaire thermique vitré qui convertissent l'énergie solaire en énergie thermique et l'ensemble est relié à un réservoir d'eau chaude sanitaire. Une pompe ou un circulateur est utilisé pour assurer la circulation du fluide caloporteur entre le réservoir de stockage et les capteurs solaires.

Ainsi, pour la régulation du système nous avons utilisé de capteurs de température DS18B20, d'une part pour surveiller la température à la sortie du capteur solaire et d'autre part celle régnant dans le réservoir de stockage.

Dans ce travail, deux processus de régulation sont utilisés :

Une régulation de température différentiel qui consiste à actionner la pompe de circulation du fluide caloporteur, et l'autre travaillant en tout ou rien pour actionner le thermoplongeur pour assurer un appoint de chauffage en cas de déficit de chauffage par énergie solaire.

Voici les différentes étapes du processus de ce régulateur :

- 1.** Le capteur solaire thermique reçoit de l'énergie solaire et la convertit en énergie thermique. Cette énergie permet de chauffer le fluide caloporteur circulant à l'intérieur du circuit primaire. Une fois chauffée, cette eau est acheminée par la pompe vers le réservoir de stockage et moyennant l'échangeur de chaleur l'énergie produite est transférée à l'eau du circuit d'utilisation du réservoir.
- 2. Mesurage des Températures :** Deux capteurs DS18B20 sont utilisés dans le système
 - **le premier** est placé à la sortie du chauffe-eau solaire pour mesurer la température de l'eau chauffée à la sortie.
 - **le deuxième** est placé dans le réservoir pour surveiller la température de l'eau stockée.
- 3. Contrôle par Arduino :** L'Arduino prend en charge le contrôle automatique du système en analysant les températures en temps réel et en suivant ses instructions du programme qui sont téléverser dans l'Arduino :
 - Lorsque la température mesurée par le capteur situé à la sortie du chauffe-eau solaire thermique (Temp1) est supérieure à celle relevée dans le réservoir (Temp2) , un relais est activé pour mettre la pompe en marche et faire transférer l'eau chaude vers le réservoir .
 - Et si la (Temp2 est supérieur à la Temp1), le relais désactive la pompe pour stopper la circulation, et un autre relais active la plaque chauffante pour atteindre la consigne souhaitée.
 - Une fois la Temp2 (température du réservoir) atteint ou dépasse la consigne, les deux relais coupent simultanément la plaque chauffante et la pompe.
- 4. La consigne :** Dans notre schéma du montage on trouve deux boutons poussoirs : bleu pour augmenter la consigne (+) et rouge pour diminuer la consigne (-). Pour passer au mode « modification de consigne » il faut maintenir les deux boutons au même temps pendant 5 secondes puis voir le message « modification de la consigne » s'affiche sur l'écran LCD, après ça on peut changer la consigne.

III.2.4. Tableau récapitulatif des rôles des composants :

Composants	Rôle dans le système
Arduino UNO R3	Unité centrale, exécute le programme de régulation
DS18B20	Mesure la température du chauffe-eau et du réservoir
DHT11	Mesure la température et l'humidité ambiante
Pompe	Circule l'eau du chauffe-eau solaire vers le réservoir
Plaque chauffante	Chauffe le réservoir si besoin
Relais	Commande la pompe et la plaque chauffante
LCD 20×4 I2C	Affiche les températures, la consigne, les messages
Boutons poussoirs	Permettant de modifier la consigne

Tableau III. 1 rôle des composants

III.3. Réalisations final sur circuit imprimé :

Après avoir vérifié le bon fonctionnement du système sur breadboard lors de la phase de prototypage, on a procédé à la réalisation d'un circuit imprimé (PCB). Cette étape nous a permis d'organiser de manière ordonnée les composants électroniques sur une carte, garantissant ainsi des connexions robustes et un montage plus fiable, solide.

III.3.1. Etapes de conception de circuit imprimé :

III.3.1.1. Réalisation du schéma électrique :

Pour commencer, nous avons repris le câblage réalisé sur la maquette d'essai afin de le transformer en un schéma électrique précis à l'aide d'un logiciel Eagle. Nous avons intégré l'ensemble des composants du projet (Arduino, capteurs DS18B20, relais, afficheur LCD, boutons poussoirs, etc.), en les représentant avec toutes leurs connexions respectives.

III.3.1.2. Routage de piste :

Après avoir finalisé le schéma électrique, on a réalisé le routage des pistes. Cette étape consistera réaliser à l'aide du logiciel les connexions électriques entre les broches des composants directement sur la carte, remplaçant ainsi les fils utilisés lors de la phase d'essai sur la breadboard. La figure suivante montre le résultat du routage :

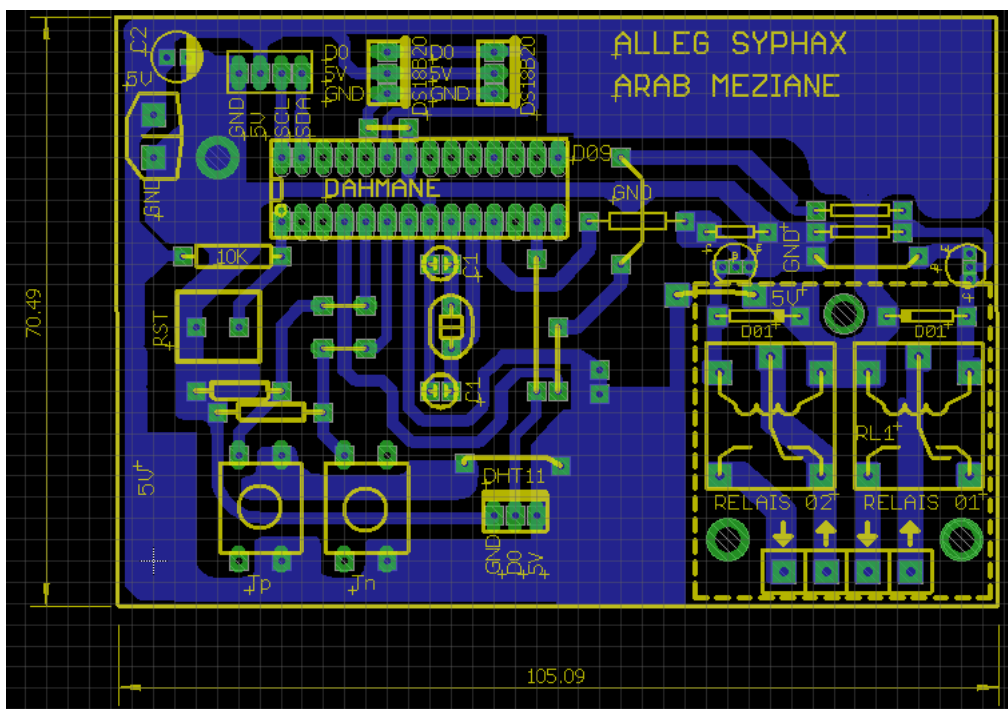


Figure III. 6 Schéma électrique.

Après avoir finalisé la conception du circuit imprimé, on a remis l'impression et la fabrication de la carte à une entreprise spécialisée, pour garantir la qualité et la fiabilité de la carte PCB. Après l'obtention de la carte, on a entamé l'assemblage, avec le placement et la soudure des différents composants :

- **Placements des composants sur la carte :**
 - Les capteurs DS18B20 ont été placés sur la partie supérieure de la carte, à côté de l'afficheur LCD ; juste en dessous on trouve le microcontrôleur ATmega328.
 - Sur la partie inférieure de la carte on a placé les deux boutons poussoirs et le capteur DHT11.

- Les deux relais 5VSRD-05VDC-SL-C ont été placés à la partie inférieure droite de la carte.
- **Éléments de sécurité et fiabilité :**
 - On a ajouté des transistors (BC547) utilisés pour la commutation des relais, ils permettent d'isoler le microcontrôleur des charges plus importantes et de protéger ses sorties.
 - Diodes (1N4007), placées en parallèle sur les bobines des relais, elles servent de diodes de roue libre pour éviter les surtensions lors de la coupure des relais, protégeant aussi les autres composants du circuit.
 - Condensateurs, ajoutés pour le filtrage de l'alimentation et la stabilisation du fonctionnement du microcontrôleur, ils réduisent les parasites et améliorent la robustesse du montage.
 - On a ajouté un bouton de réinitialisation (RST), pour permettre le redémarrage du système en cas de dysfonctionnements.

La figure suivante présente l'emplacement de l'ensemble des composants sur la carte PCB :

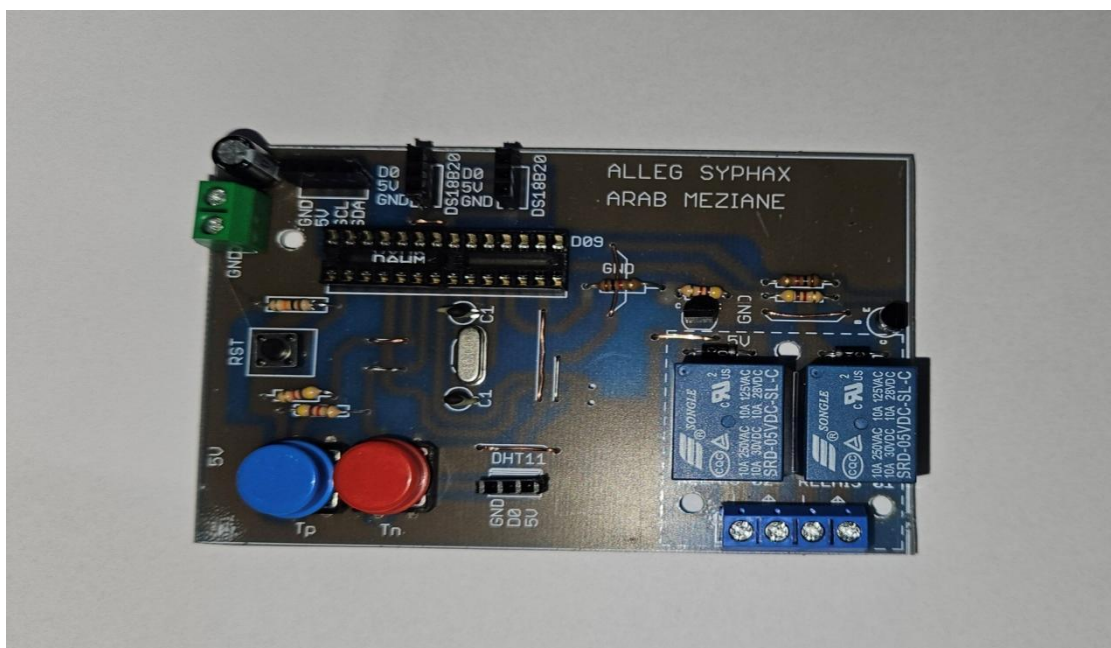


Figure III. 7 Emplacements des différents composants sur le PCB

III.3.2. Equivalence des broches entre Arduino et ATmega328 pour le câblage des composants :

Dans le cadre de la réalisation de notre carte électronique, on a choisi de souder directement le microcontrôleur ATmega328 sur le circuit imprimé, plutôt que d'utiliser une carte Arduino complète. Cette méthode impose de bien comprendre la correspondance entre les broches utilisées dans le code Arduino et les pattes physiques du microcontrôleur.

Sur Arduino, les broches sont identifiées par des numéros logiques (comme PIN2, PIN5, A0...), ce qui simplifie la programmation. Cependant, lorsqu'on utilise le microcontrôleur nu, chaque broche Arduino correspond à une patte physique précise de l'ATmega328. Il est donc indispensable de réaliser une correspondance précise lors de la conception du PCB afin de garantir que chaque composant soit relié à la bonne patte du microcontrôleur.

Voici comment on a assuré cette correspondance pour notre projet :

- Le relais de la pompe, il était branché sur la broche numérique 2 de l'Arduino, puis on l'a connecté à la patte 4(PD2) de l'ATmega328.
- Le relais de la plaque chauffante, prévu sur la broche 5 de l'Arduino, est relié à la patte 11(PD5) de l'ATmega328.
- Les deux capteurs de température DS18B20, utilisaient la broche A0 sur Arduino, sont branchés sur la patte 23 (PC0).
- Le capteur DHT11, connecté à la broche 7 de l'Arduino, est relié à la patte 13(PD7).
- Les boutons poussoirs : le bouton « + » est branché à la broche 3 sur Arduino, soit la patte 5(PD3). Le bouton « - » sur la broche 4 d'Arduino, soit la patte 6(PD4).
- L'afficheur LCD I2C utilise la broche SCL « A5 sur Arduino, patte 28(PC5) sur l'ATmega328 » et la broche SDA « A4 sur Arduino, patte 27(PC4) »

Ce travail de correspondance nous a permis d'assurer que le fonctionnement du montage sur le PCB reste parfaitement fidèle à celui obtenu lors des essais sur la carte Arduino. Ainsi, lors de la conception du circuit imprimé, il a été essentiel de connecter chaque composant à la patte physique du microcontrôleur correspondant à la broche utilisée dans le code Arduino, en s'appuyant sur le tableau sur le tableau d'équivalence officiel entre les broches Arduino et celles de l'ATmega328, présenté ci-dessous :

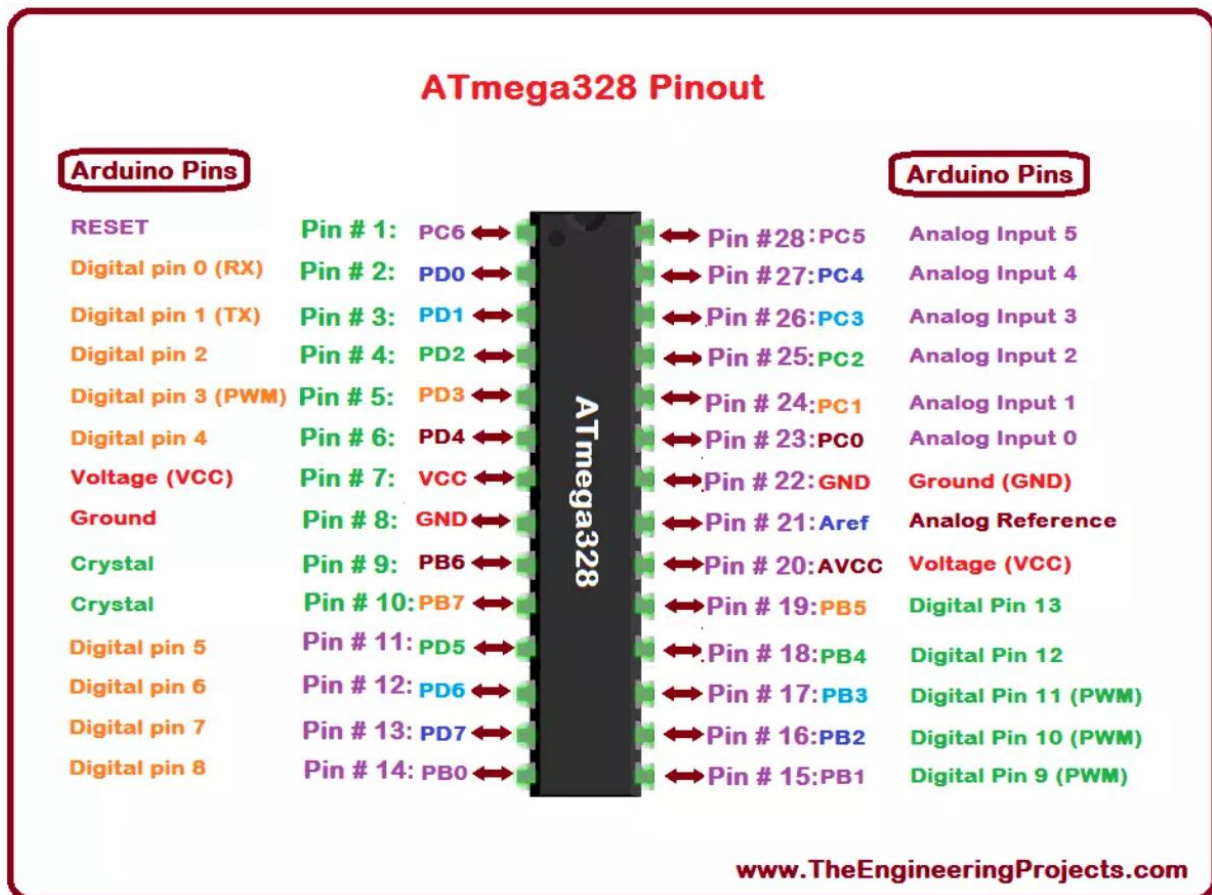


Figure III. 8 Equivalence entre les broches Arduino et les pattes de l'ATmega328

[29]

III.3.3. Vue d'ensemble du montage final réalisé :

Dans le but de renforcer la protection des relais et d'améliorer la sécurité de la partie de commande, un contacteur a été intégré entre la sortie des relais et la partie de puissance du système. Ce dispositif, installé hors du circuit imprimé, permet une séparation efficace de la commande des charges de puissance et empêche toute possibilité de surtension ou de retour de courant vers l'Arduino.

De plus, pour simplifier l'étape de test et de représenter de manière claire le fonctionnement du système, deux lampes ont été mises en pratique à la place de la pompe et de la résistance chauffante : l'allumage de chaque lampe indique respectivement l'activation de la pompe ou de la résistance.

Cette méthode offre la possibilité d'évaluer le fonctionnement du régulateur sans faire appel directement aux actionneurs physiques, tout en assurant une supervision simple et sécurisée du cycle de fonctionnement.

La figure ci-après représente la réalisation finale , comportant le contacteur de sécurité :

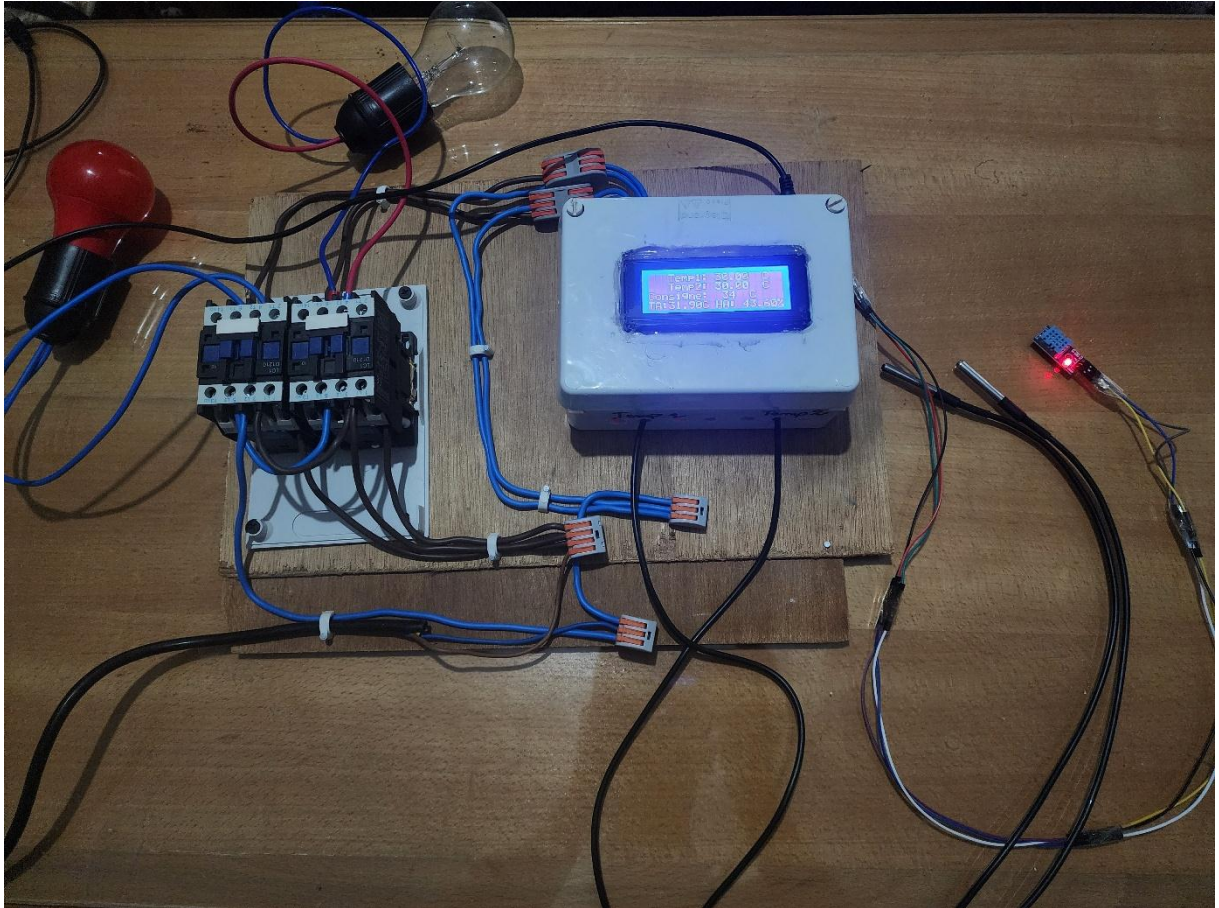


Figure III.9. Vue d'ensemble du montage final réalisé sur circuit imprimé avec contacteur et lampes.

La figure ci-dessus représente l'assemblage final réalisé sur circuit imprimé, qui comprend la carte de commande, les relais ainsi que les contacteurs de sécurité. Les deux lampes utilisées fonctionnent comme des témoins pour représenter respectivement la résistance chauffante (rouge) et la pompe (blanche) pendant les tests. Cette mise en place permet de vérifier le fonctionnement du système tout en maintenant une séparation claire entre la partie commande et la partie puissance.

III.3.4. Tests et validation du montage :

Ce tableau représente les différents scénarios de test effectués, indiquant pour chaque essai la consigne, les températures mesurées, l'action attendue et le résultat observé :

Scenario testé	Consigne	Temp 1	Temp 2	Action attendue	Résultat observé
Temp 1 >Temp 2	45°C	40°C	30°C	Pompe : ON Résistance : OFF	Activation de la pompe
Temp 2 >= Temp 1	45°C	35°C	40°C	Pompe : OFF Résistance : ON	Activation de la résistance
Temp 2 >= Consigne	45°C	35°C	45°C	Pompe : OFF Résistance : OFF	-Système a l'arrêt
Modification Consigne	50°C	35°C	45°C	Pompe : OFF Résistance : ON	- Affichage modification consigne -Réaction rapide
Affichage LCD	50°C	35°C	45°C	-Affichage de : Temp 1 Temp 2 consigne TA , HA	- Affichage correct

Tableau III.2. Différents tests effectués

III.4 Conclusion :

Ce travail nous a permis de concevoir, simuler et réaliser un système complet de régulation thermique associant des capteurs numérique (DS18B20 en pratique, LM35 en simulation pour des raisons de compatibilité) ; une interface de commande et des actionneurs pilotés par le microcontrôleur.

Les simulations sur Proteus menées avec le LM35 ont permis de valider la logique de fonctionnement du régulateur, tandis que la réalisation pratique avec le DS18B20 et le DHT11 a confirmé l'efficacité et la fiabilité du système en condition réelles. Grace à l'affichage LCD,

Chapitre III Tests et validation et résultats expérimentaux

la température est surveillée en temps réel et la régulation s'effectue automatiquement via l'activation ou la désactivation de la pompe et la résistance selon les besoins.

Ce projet ouvre la voie à de nombreuses améliorations, telles que l'intégration de capteurs de niveau d'eau (CAP-028) pour éviter le fonctionnement à vide des actionneurs, ou encore de capteurs de qualité de l'eau (pH, conductivité) pour une gestion plus fine et sécurisée. Ces évolutions permettront d'augmenter la sécurité et la performance du système, tout en facilitant son adaptation à d'autres applications de gestion thermique automatisée.

CONCLUSION GENERALE

Conclusion générale

La régulation de température est cruciale pour des processus industriels variés, elle joue un rôle clé dans la qualité des produits, l'efficacité énergétique, la sécurité ainsi que les coûts associés.

La production de l'eau chaude solaire, constitue un secteur primordial dans les applications de l'énergie solaire. Par ailleurs, généralement durant les périodes creuses, ou l'eau contenue dans le réservoir n'est pas suffisamment chauffée, un système d'appoint en matière de chauffage s'impose *afin de combler le déficit énergétique résultant et d'assurer le bon fonctionnement du système*, il s'agit d'un thermoplongeur. De même une régulation de type TOR est utilisée pour la mise en marche ou l'arrêt de la résistance de chauffage plongée dans le réservoir de stockage.

L'objectif de notre travail était de réaliser un système de régulation de température, ce dernier consiste à installer des sondes de température dont l'une permet de mesurer la température de l'eau dans le réservoir de stockage et l'autre à la sortie du capteur. Ainsi ceci nous permettra de commander la mise en marche ou l'arrêt de la pompe de circulation selon la différence de température entre les valeurs mesurées par les deux sondes.

Ce type de régulateur permet surtout d'optimiser le fonctionnement du chauffe-eau en assurant un maximum d'utilisation de l'énergie solaire et la résistance permet l'asservissement électrique pour les cas les plus défavorables.

Notre système est composé d'un ensemble de capteurs ayant pour rôle l'acquisition des données qui sont traitées par une carte à microcontrôleur Arduino, d'actionneurs servant à réguler les grandeurs choisis.

Pour la réalisation de notre projet, nous avons opté en premier lieu pour une simulation en utilisant le logiciel Proteus afin d'assurer le comportement du système avant sa réalisation pratique, les résultats obtenus étaient largement satisfaisants.

La réalisation pratique a confirmé l'efficacité et la fiabilité du système en condition réelles. Grâce à l'affichage LCD, la température est surveillée en temps réel et la régulation s'effectue automatiquement via l'activation ou la désactivation de la pompe et la résistance selon les besoins.

Conclusion générale

Ce travail nous a permis d'acquérir une expérience enrichissante dans le domaine de la régulation et de la programmation et nous a permis également de mettre en pratique les connaissances acquises durant notre cursus.

Perspective :

Ce projet ouvre la voie à de nombreuses améliorations :

- L'intégration de capteurs de niveau d'eau pour éviter le fonctionnement à vide des actionneurs,
- Ajout de capteurs de qualité de l'eau (pH, conductivité) pour une gestion plus fine et sécurisée.

Annexe

Programme Arduino du régulateur :

```
// Déclarations des capteurs
#include <OneWire.h>
#include <DallasTemperature.h>
// Pin DATA des capteurs DS18B20
#define ONE_WIRE_BUS A0
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
// Adresse des capteurs (à remplir après détection)
DeviceAddress capteur1, capteur2;

// Sorties
int pompe = 2;
int RESISTANCE = 5;

// Boutons de réglage
int Tp = 3;
int Tn = 4;

// Réglage consigne
int consigne = 30;
int etatTp = 0;
int etatTn = 0;
int reglage = 0;

// Timers
unsigned long Tempo = 5000;
unsigned long Tempo2 = 10000;
```

```
unsigned long Temps_Actuel;

unsigned long Temps_Precedent = 0;

unsigned long Temps_Precedent2 = 0;

#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27,20,4);

#include <DHT.h>

#define DHTPIN 7

#define DHTTYPE DHT11

DHT dht (DHTPIN,DHTTYPE);

float temperature;

float humidite;

void setup() {

    sensors.begin();

    // Trouver les adresses des capteurs

    if (!sensors.getAddress(capteur1, 0)) Serial.println("Capteur 1 non
trouvé");

    if (!sensors.getAddress(capteur2, 1)) Serial.println("Capteur 2 non
trouvé");

    // Résolution des capteurs (9 à 12 bits)

    sensors.setResolution(capteur1, 10);

    sensors.setResolution(capteur2, 10);

    pinMode(pompe, OUTPUT);

    pinMode(RESISTANCE, OUTPUT);

    pinMode(Tp, INPUT);

    pinMode(Tn, INPUT);

    Serial.begin(9600);

    lcd.init();
```

```
lcd.backlight();

lcd.setCursor(0,0);

lcd.print("DAHMANE AZUL");

delay(1500);

lcd.clear();

dht.begin();

// Réglage consigne

if ((etatTp == HIGH) && (etatTn == HIGH) && (Temps_Actuel -
Temps_Precedent >= Tempo)) {

    Temps_Precedent = Temps_Actuel;

    Temps_Precedent2 = Temps_Actuel;

    reglage = 1;

    while (reglage == 1) {

        if (digitalRead(Tp) == HIGH) {

            consigne++;

            delay(200);

        }

        if (digitalRead(Tn) == HIGH) {

            consigne--;

            delay(200);

        }

        Serial.print("Consigne : ");

        Serial.println(consigne);

        lcd.setCursor(0,3);

        lcd.print("MODIFICATION CONSIGN");

        lcd.setCursor(11,2);

        lcd.print(consigne);

    }

}
```

```
    Temps_Actuel = millis();

    if (Temps_Actuel - Temps_Precedent2 >= Tempo2) {

        reglage = 0;

        break;

    }

}

}

// COMMANDE PRINCIPALE

if (Temp2 < Temp1) {

    digitalWrite(pompe, HIGH);

    digitalWrite(RESISTANCE, LOW);

} else {

    digitalWrite(pompe, LOW);

    digitalWrite(RESISTANCE, HIGH);

}

if (Temp2 >= consigne ){

    digitalWrite(pompe, LOW);

    digitalWrite(RESISTANCE, LOW);

}

}

// AFFICHAGE

Serial.print("Température 1 (Temp1): ");

Serial.print(Temp1);

Serial.println(" °C");

Serial.print("Température 2 (Temp 2): ");

Serial.print(Temp2);

Serial.println(" °C");
```

```
Serial.print("Consigne: ");  
Serial.println(consigne);
```

```
Serial.print("Temperature: ");  
Serial.print(temperature);  
Serial.println(" C");  
Serial.print("Humidite: ");  
Serial.print(humidite);  
Serial.println(" %");
```

```
lcd.setCursor(3,1);  
lcd.print("Temp2: ");  
lcd.setCursor(10,1);  
lcd.print(Temp2);  
lcd.setCursor(16,1);  
lcd.print(" C");  
lcd.setCursor(9,3);  
lcd.print(" HA: ");  
lcd.setCursor(14,3);  
lcd.print(humidite);  
lcd.setCursor(19,3);  
lcd.print("%");
```

Bibliographie

Bibliographie

- [1] SEDDAR Yahia, GHEZA Katia. « Contribution à l'Implémentation d'un Régulateur Flou sous un API S7-300 ». Mémoire de Fin d'Etude de MASTER PROFESSIONNEL. **2017**.<https://dSPACE.ummto.dz/server/api/core/bitstreams/c284e12c-0e19-4c2d-b7f2-018afa24050f/content>.
- [2] Cours_regulation_electromecanique. « Régulation industrielle ».3eme année licence électromécanique.
- [3] Mallek Abdennour , MERABET Ghiles. « Régulation de la température huile de turbine de la centrale thermique de CAP-DJINET ». Mémoire de Fin d'Etude **2017**
<https://dSPACE.ummto.dz/server/api/core/bitstreams/0b030602-a007-4fae-b3e8-a001c11deb0e/content>.
- [5] DR.M.Rabi, « régulation analogique industriel », ESTT.G. Thermique.
<https://fr.scribd.com/document/464685140/Regulation-analogique-industrielle-ESTT-G-Thermique>.
- [6] Benkrea Bilel Bouguern Amarseifeddine. «Etude de la boucle de régulation de la température du Réacteur de polymérisation du complexe CP2K» Projet professionnel de fin de formation 2020.
[.file:///C:/Users/REPTECH/Desktop/m%C3%A9moire%20%20alleg/Benkrea-Bilel.pdf](file:///C:/Users/REPTECH/Desktop/m%C3%A9moire%20%20alleg/Benkrea-Bilel.pdf) .
- [7] BOUKABOUS Fedwa, AHMED M'BAREK Manel. « Implémentation sur DSP de la commande PID optimisée à l'aide de l'algorithme TLBO : Application à la commande d'un moteurDC ».Mémoire de Fin d'études.2018.
<https://di.univblida.dz/jspui/bitstream/123456789/2314/1/m%C3%A9moire%20hmd%2018%2006.pdf>.
- [8].Dr.BOU CERRE DJLEILA. « REGULATION INDUSTRIELLE » Support de cours pédagogique.2022.
[https://dSPACE.univguelma.dz/jspui/bitstream/123456789/15876/1/POLYCOPIE%20DE%20COURS%20R%C3%A9gulation Industrielle.pdf](https://dSPACE.univguelma.dz/jspui/bitstream/123456789/15876/1/POLYCOPIE%20DE%20COURS%20R%C3%A9gulation%20Industrielle.pdf).
- [9] Omega Engineering, *Temperature Controllers Selection Guide*, 2022:<https://www.omega.com/en-us/resources/temperature-controllers>.
- [10]. Analog Devices, *Principles of PID Control*, 2023: <https://www.analog.com/en/analog-dialogue/articles/pid-control.html>.
- [11]. Barabanov, M., *Embedded Control Systems: From Analog to Digital Regulation*, Springer, 2019.
- [12]. Digi-Key Electronics, *Digital Temperature Controller Design Guide*, 2023. [En ligne] :<https://www.digikey.com/en/articles/designing-a-digital-temperature-controller>.

Bibliographie

[13] <https://www.moussasoft.com/ds18b20arduino/#:~:text=Le%20capteur%20DS18B20%20est%20un,%2B%2F%2D%200%2C5%20degr%C3%A9s%20Celsius.>

[14] https://www.framboise314.fr/wp-content/uploads/2017/11/04_1_1-wire-DS18B20.pdf.

[15] https://www.framboise314.fr/wp-content/uploads/2017/11/04_1_1-wire-DS18B20.pdf.

[16] <https://www.moussasoft.com/ds18b20-arduino/>.

[17]. ACHLAF AYMEN-LAGREB ALI HICHEM : Réalisation d'un thermomètre infrarouge automatisé Mémoire de Maste, Université SAAD DAHLAB de Blida année

[18] <https://www.geya.net/fr/5v-relay-module-how-it-works-and-application/>

[19] <https://www.moussasoft.com/produit/relai-electrique-5v-srd-05vdc-sl-c/>

[20] Optimal Protech, « DHT11 - Capteur température/Humidité / 0 à + 50°C / RH: 20 - 90% », 2025.

<https://www.optimalprotech.com/produit/dht11-capteur-temperature-humidite-55-a-125c/>

[21].MANAMANI YASSINE. « Amélioration des fonctionnalités de l'horloge géante de la FSSA».Mémoire de Fin d'Etude

2019.<http://dspace.univbouira.dz:8080/jspui/bitstream/123456789/9636/1/M%c3%a9moire%20finale.pdf>.

[22] .Aghiles ABEDHassina KACI MOUSSA. « Conception et réalisation d'un système de régulation à base d'un microcontrôleur ». Mémoire de fin d'étude de MASTER ACADEMIQUE.2018.

file:///C:/Users/REPTECH/Desktop/m%C3%A9moire%20%20alleg/AbedAghiles_KaciMoussaHassina.pdf.

[23] <https://www.electronicshub.org/arduino-introduction/#Introduction>.

[24] <https://www.revolution-energetique.com/fiches-pedagogiques/comment-fonctionne-un-panneau-solaire-thermique/>

[26] INGUEL MALHA. « Conception et réalisation d'un système Domotique par GSM ». Mémoire de Fin d'étude 2017.

<https://dspace.ummtto.dz/server/api/core/bitstreams/b2eef448-e78a-489a-aa54-ca020144dc1b/content>.

[27] CHAIB Amina. « Optimisation des paramètres d'un régulateur PID flou par les algorithmes génétiques ». Memoire 2010.

<https://biblio.univannaba.dz/wpcontent/uploads/2014/06/Memoire-Final.pdf>.

Bibliographie

[28].<https://www.guide-panneaux-photovoltaiques.be/technologies/le-panneau-solaire-thermique/>

[29].<https://images.theengineeringprojects.com/image/webp/2017/07/ATmega328Pinout.png.webp?ssl=1> .