

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البلدية
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique



Mémoire de Master

Mention Electronique
Spécialité Instrumentation

Présenté par

Guerach Abdelkader

&

Yahiaoui Abderraouf

Affichage graphique des capteurs électrique

Proposé par : Amirouche .Nesrine

Année universitaire 2020/2021

Remerciements

Nous tenons à remercier à Dieu de nous avoir donné la patience de terminer ce travail. Au terme de cette étude, nous souhaitons remercier chaleureusement notre encadreur de mémoire **M. Amirouche Epse. KAIS Nesrine** pour son suivi et ses encouragements, ses orientations et ses précieux conseils. Nous remercions également tous les professeurs et les enseignants du département génie électrique pour leurs conseils constructifs et encouragements qui ont été à la fois un complément et supplément indispensable dans notre projet de fin d'étude. Enfin, nous exprimons nos reconnaissances à tous les membres de jury d'avoir acceptés de lire ce manuscrit et d'apporter les critiques nécessaires à la mise en forme finale de ce travail.

Et sans oublier nous chère famille et nous amis et Tu le monde qui a été avec nous dans notre carrière et je remercie

ملخص:

في هذا العمل قمنا بدراسة اللواقط الكهربائية وعرض نتائج قياساتها في واجهة عرض قمنا ببرمجتها بالإستعانة بلغة البايثون و الأردوينو لتتمكن من عرض ما تستقبل هذه اللواقط بصورة أنية بدلالة الزمن .

كلمات المفاتيح: لواقط الكهربائية ، واجهة المستخدم الرسومية ، بايثون ، رسم بياني

Résumé : Dans le travail qui suit, nous avons étudié et afficher dans une interface graphique les déferlants capteurs électriques. nous avons utilisé la relation port série et la fonction pyfirmata entre le langage Python et l'Arduino pour arriver à réaliser un affichage de résultats en temps réel.

Mots clés : Arduino ; IDE ; GUI ; python ; interface graphique

Abstract: In this work we have studied the electrical sensors and project their result in a graphical interface using the relation between the Arduino and the python (the pyfirmata protocol). The work plan consists of two parts organized as follows: plotting exists data, and viewing data from sensors in real time. Finally, we presented a general conclusion.

Keywords: graphical interface; console; IDE; sensors

Table des matières

Introduction générale	1
Chapitre 1 : Généralités sur les capteurs	2
1.1 Définition d'un capteur	2
1.2 Caractéristiques d'un capteur	3
1.3 les capteurs numériques	3
1.4 Tout ou rien (TOR) :	3
1.5 Train d'impulsions :	3
1.6 Échantillonnage :	4
1.7 les capteurs analogiques	5
1.8 Classification des capteurs :	6
1.8.1 Capteurs actifs :	6
1.8.2 Capteurs passifs	8
1.9 Capteurs de couleurs TCS3200	9
1.9.1 Déception :	12
1.9.2 fonctionnement	12
1.10 Capteurs de proximité :	13
1.11 Capteurs de proximité (ultrasons) :	13
1.11.1 le principe des ultrason	13
1.11.2 Applications des capteurs à ultrasons	15
1.11.3 Principes de détection	15
1.11.4 Détecteurs à ultrasons	15
1.12 Capteurs de thermocouple	16
1.12.1 Qu'est-ce qu'un thermocouple ?	16
1.12.2 Applications des capteurs à ultrasons	16
1.12.3 Zoom sur la sécurité par thermocouple	17
1.13 Thermocouple max6675	17
1.13.1 Caractéristiques	19
1.13.2 Utilisation	19
1.14 conclusion	20
Chapitre 2 Choix du langage de développement de l'interface	21
2.1 Introduction	21

2.2	Arduino.....	22
2.2.1	Qu'est-ce que le logiciel Arduino ?	23
2.2.2	Types de cartes Arduino.....	24
2.3	L'interface graphique	26
2.3.1	Le GUI « Graphical User Interface »	27
2.4	Python	28
2.4.1	Les bibliothèques dans python.....	30
2.4.2	Installer pip avec le cmd	31
2.5	Tkinter	30
2.6	NUmpy	33
2.7	matplotlib	34
2.8	pyserial	36
2.8.1	Install pyserial.....	37
2.8.2	Principe	39
2.9	Les objets.....	39
2.10	Zone de texte¶	40
2.11	Conclusion	42
Chapitre 3 : chaines de conditionnement des capteurs		43
3.1	Introduction.....	43
3.2	Disjoncteurs.....	43
3.3	Relais de protection	45
3.3.1	Relais électromécaniques	46
3.3.2	Relais statiques	46
3.3.3	Relais numériques	47
3.3.4	Principes de fonctionnement des relais de protection	48
3.4	Fusible	48
3.5	Arduino et protection circuit.....	49
3.5.1	Précautions d'utilisation	49
3.5.2	Savoir reconnaître les composants électroniques	49
3.5.3	Représentations graphiques	49
3.5.4	protection du port USB	50
3.5.5	Protection de la carte Arduino	50
3.5.6	Protection des entrées numériques	50
3.5.7	Protection des composants.....	50
3.5.8	Circuits de commande et de puissance	51

3.5.9	Circuit de commande.....	51
3.5.10	Circuit de puissance	51
3.5.11	Court-circuit.....	51
3.6	Conclusion	52
Chapitre 4 Les resultats.....		54
4.1	Introduction.....	54
4.2	Logo d'interface :.....	54
4.3	La Botton « email me » :.....	55
4.4	les Botton principales.....	55
4.4.1	la 1 er Botton	55
4.4.2	la 2emme Botton	57
4.4.3	la 3emme Botton	58
4.5	Real time data Reading window:.....	60
4.6	Conclusion	61
Conclusion générale.....		61
Bibliographie.....		62

Liste des figures

Figure 1-1 Role general d'un capteur	4
Figure 1-2 chaine d'accuisition.....	6
Figure 1-3 signal analogique sous forme escalier	7
Figure 1-4 numérisation d'un signal analogique	8
Figure 1-5 train d'impulsion	12
Figure 1-6 transformation d'un capteur analogique.....	12
Figure 1-7 Exemple de capteur a effet hall.....	14
Figure 1-8 Exemple de capteur tcs3200	15
Figure 1-9 exemple de capteur de couleur liée avec arduino	16
Figure 1-10 fonctionnement de capteur ultrason.....	17
Figure 1-11 exemple d'application d'ultrason	18
Figure 1-12 exemple de capteur max6675	19
Figure 1-13 Exemple de thermocouple de type-K	20
Figure 1-14 breakout MAX6675 pour thermocouple type-K.....	23
Figure 2-1 une carte de commande Arduino UNO avec ses connecteurs	24
Figure 2.2 l'interface de l'application arduino	25
Figure 2-3 le site de telechargement de python 3	26
Figure 2-4 l'interface de IDE pycharm	30
Figure 2-5 la fenaitre Executer de windows	31
Figure 2-6 installation de bibliotheque pip avec la console windows	35
Figure 2-7 l'importation de bibliotheque tkinter dans le code.....	38
Figure 2-8 Exemple de code d'in fenétre avec tkinter	38
Figure 2-9 installation de bibliothéque numpy avec la console de windows.....	40
Figure 2-10 Exemple d'un graphe réalisé avec la bibliotheque numpy.	41
Figure 2-11 Recherche de bibliotheque matplotlib dans l'IDE.	42
Figure 2-12 installation de bibliotheque matplotlib	45
Figure 2-13 recherche de bibliotheque matplotlib dans l'IDE.	47
Figure 2-14 Exemple de zone de text dans la fenetre réalisé avec tkinter.	48
Figure 2-15 les défferantes types de graphe réalisé par matplotlib.	51
Figure 3-1 exemple de disjoncteur	56
Figure 3-2 types des relais	56
Figure 3-3 Exemple de relais numérique (SEPAM de schneider électrique	55
Figure 4-1 le chemin d'importation de le logo sur le code	55
Figure 4-3 Exemple de résultat de logo	56
Figure 4-4 exemple de botton sur l'interface	56
Figure 4-5 Exemple de 1^{er} botton principale.	59
Figure 4-6 résultat de trace de 1^{er} botton principale	60
Figure 4-7 Exemple de 2^{em}me botton principale	62
Figure 4-8 resultat de trace de tableau	62
Figure 4-9 Exemple de 3^{em}me botton principale	64

Figure 4-10 resultat de trace des fonction	65
Figure 4-11 Exemple de fenetres en cascade	70
Figure 4-12 fenetre finale de l'interface	71
Figure 4-13 fenetre de l'interface a résultats en temps réel	73

Liste des tableaux

Tableau 1.1 capteurs passifs, principes physiques de base	76
Tableau 4.1 résultats prélevés de mesures d'un TP de modélisation d'un capteur de température	76

Listes des acronymes et abréviations

GUI	graphical user interface
IDE	Integrated Développement Environnement

Introduction générale

Les capteurs sont largement utilisés dans plusieurs domaines (santé, automobile, électromagnétisme, ...), et sont employés dans la vie de tous les jours pour transformer des événements mécaniques, chimiques ou thermiques en signal électrique [1-5]. Ils sont utilisés comme systèmes de détections de grandeur physiques qui relèvent la technologie microélectronique, et plus particulièrement celle du silicium, et offrent des avantages techniques et économiques [6, 7]. Les détections piézorésistive et capacitive sont parmi les principaux types de détection les plus utilisées.

Nous avons développé notre base en utilisant le gui de python et Arduino, et nous avons essayé de faire un affichage en temps réel pour les capteurs électrique suivants : Capteur de température, capteur de proximité et capteur de couleur .

Durant le parcours de formation des Master Instrumentation, la compréhension du fonctionnement et l'utilisation des capteurs d'instrumentation est indispensable. Ainsi nous avons élaborer une plateforme pour la réalisation des TP de capteur d'instrumentation, ce qui permettra aux étudiant une manipulation plus facile, et accessible.

Chapitre 1 : Généralités sur les capteurs

1.1 Définition d'un capteur

Un capteur est un composant technique qui détecte un événement physique se rapportant au fonctionnement du système (présence d'une pièce, température, etc.) et traduit cet événement en un signal exploitable par la PC de ce système. Ce signal est généralement électrique sous forme d'un signal basse tension. La figure 1 illustre le rôle d'un capteur :



Figure 1. 1 : Rôle général d'un capteur

L'information détectée par un capteur peut être d'une grande variété, ce qui implique une grande variété de besoins en capteurs. On cite parmi les plus connus et fréquents, les capteurs de position, de présence, de vitesse, de température et de niveau.

Localisation du capteur dans la chaîne

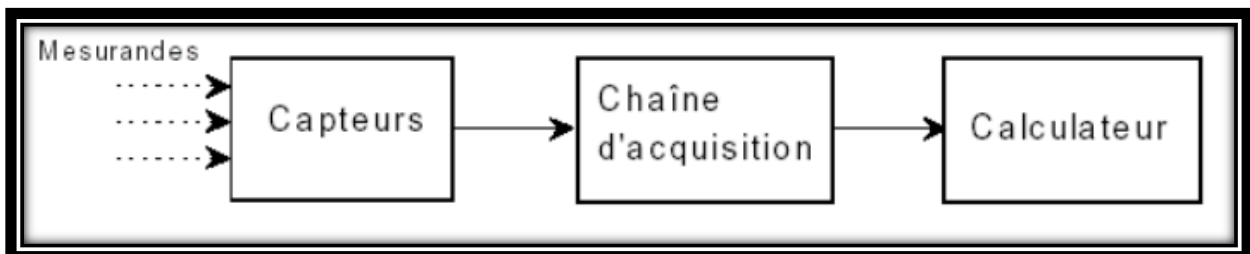


Figure 1.2 : chaîne d'acquisition

- Les capteurs sont les premiers éléments de la chaîne.
- Ce sont les interfaces entre le « monde physique » et le « monde électrique ».

1.2. CARATERISTIQUES D'UN CAPTEUR :

Certains paramètres sont communs à tous les capteurs. Ils caractérisent les contraintes de mise en œuvre et permettent le choix d'un capteur :

- L'étendue de la mesure : c'est la différence entre le plus petit signal détecté et le plus grand perceptible sans risque de destruction pour le capteur.
- La sensibilité : ce paramètre caractérise la capacité du capteur à détecter la plus petite variation de la grandeur à mesurer. C'est le rapport entre la variation ΔV du signal électrique de sortie pour une variation donnée $\Delta \Psi$ de la grandeur physique d'entrée : $S = \Delta V / \Delta \Psi$
- La fidélité : Un capteur est dit fidèle si le signal qu'il délivre en sortie ne varie pas dans le temps pour une série de mesures concernant la même valeur de la grandeur physique Ψ d'entrée. Il caractérise l'influence du vieillissement.
- Le temps de réponse : c'est le temps de réaction d'un capteur entre la variation de la grandeur physique qu'il mesure et l'instant où l'information est prise en compte par la partie commande.

1.3 Les capteurs numériques :

Les capteurs numériques donnent en sortie une valeur finie. Par exemple, si une grandeur physique croît de manière linéaire, la sortie du capteur qui va la mesurer donnera soit une information du type " TOR " (Tout Ou Rien), un train d'impulsion ou un échantillonnage.

1.4 Tout ou rien (TOR) :

Il informe sur l'état d'un système ne pouvant avoir que deux états stables comme par exemple une vanne. Si cette vanne est ouverte, on affectera à la sortie du capteur un " 0 " ou un " 1 " logique selon la volonté du concepteur du capteur et la sortie prendra l'autre valeur si elle est fermée. Ce type de montage est dit " bivalent " : c'est le fait que la sortie du montage ne peut prendre que deux valeurs (ouvert/fermé, 1/0).

1.5 Train d'impulsions :

Chaque impulsion est l'image d'un changement d'état. Par exemple, un codeur incrémental donne un nombre fini et connu d'impulsions par tour.

1.6 Échantillonnage :

C'est l'image numérique d'un signal analogique sous la forme d'un "escalier" (on retrouve cette caractéristique en sortie des convertisseurs analogiques / numériques). Pour un intervalle de tension d'entrée, correspond une seule valeur de la sortie.

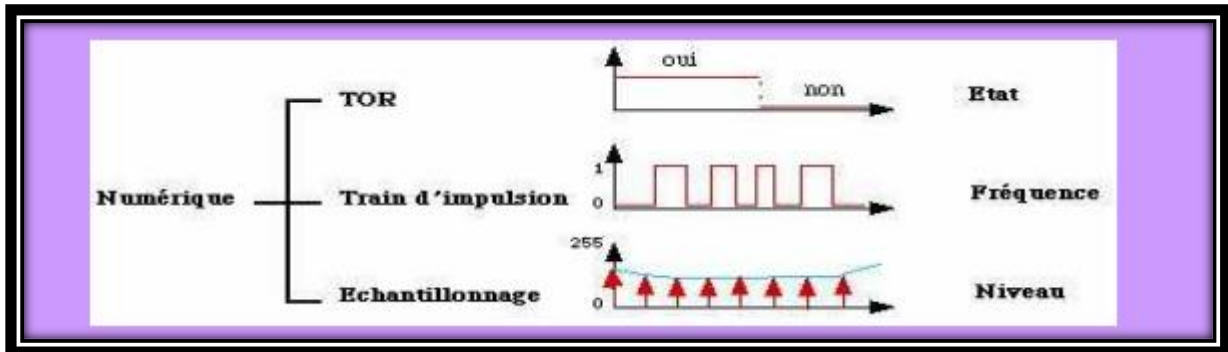


Figure 1.3: signal analogique sous forme escalier

Echantillonnage :

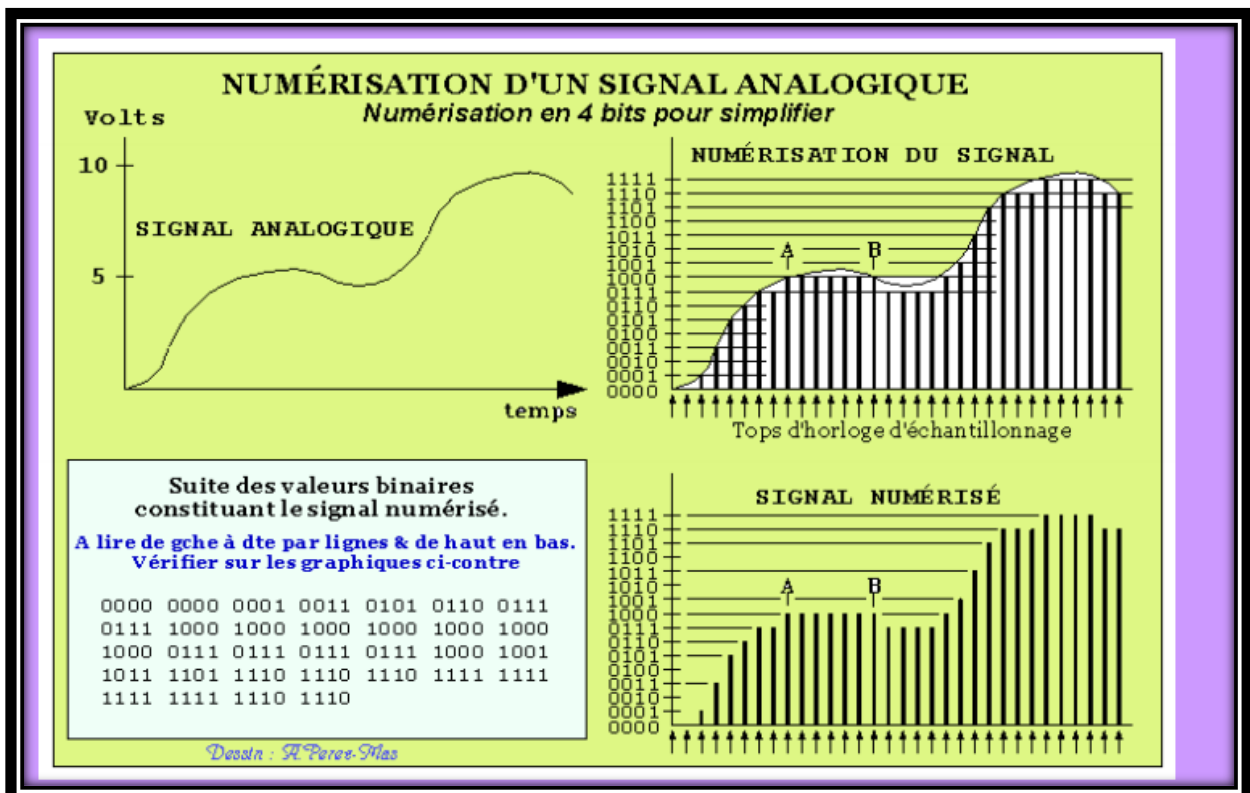


Figure 1.4: numérisation d'un signal analogique

Train d'impulsions :

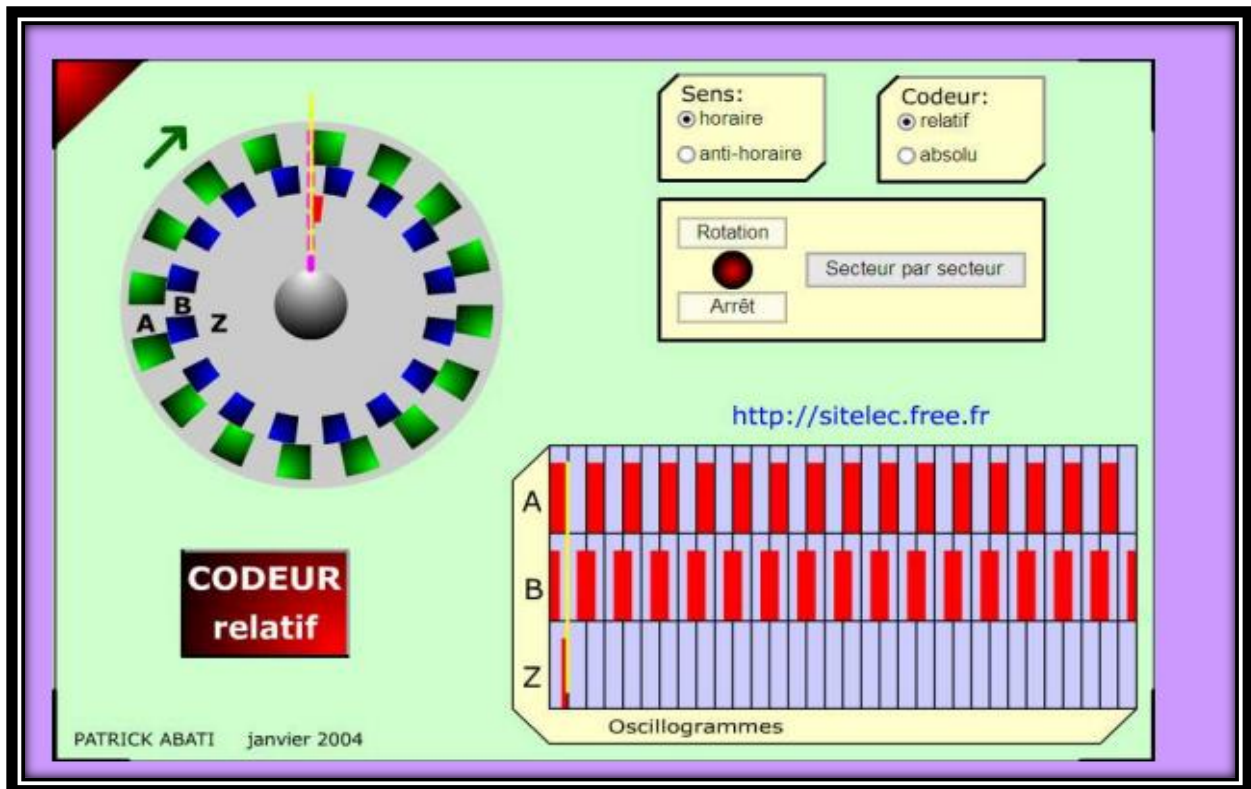


Figure 1.5: train d'impulsion

1.7 Les capteurs analogiques :

Les capteurs analogiques servent à transformer une grandeur physique en un autre type de variation d'impédance, de capacité, d'inductance ou de tension. Un signal est dit analogique si l'amplitude de la grandeur physique qu'il représente peut prendre une infinité de valeurs dans un intervalle donné. Ainsi, on peut dire que la tension de secteur sinusoïdale (230VAC) est un signal de type analogique.

- signal continu :

C'est un signal qui varie " lentement " dans le temps et qu'on retrouve en sortie d'une sonde de température, de pression ou encore d'une photo résistance.

- temporel :

C'est la forme de ce signal au cours du temps. C'est aussi la trace du signal sur l'écran d'un oscilloscope.

- fréquentiel :

C'est le spectre fréquentiel qui transporte l'information désirée :

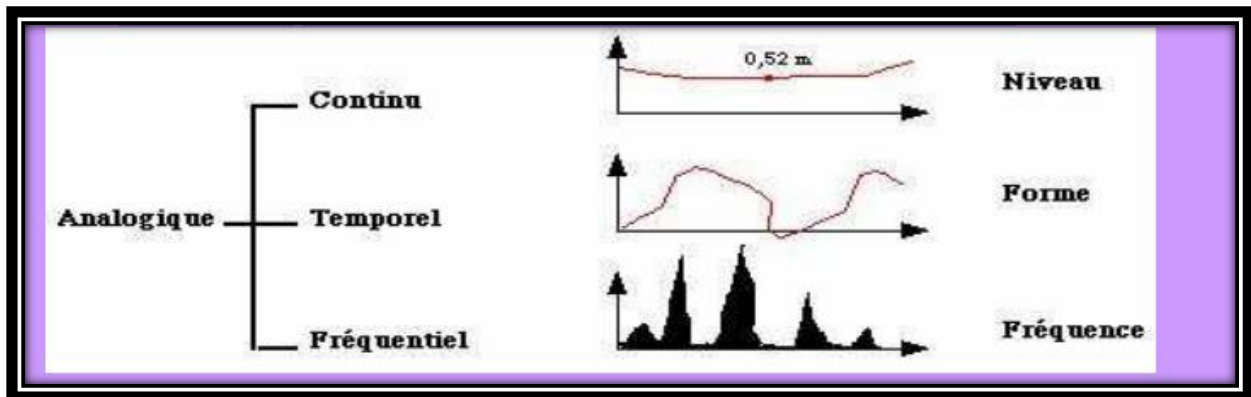


Figure 1.6: transformation d'un capteur analogique

1.8 Classification des capteurs :

Les capteurs présentent des caractéristiques différentes, leurs différences peuvent se faire aussi grâce à:

Leur principe de fonctionnement :

- capteurs actifs
- capteurs passifs

Leur principe de traduction du mesurande :

- capteurs résistif
- capteurs à effet Hall
- etc...

1.8.1 Capteurs actifs :

Les capteurs qui fonctionnent en générateur sont dits actifs, c'est à dire qu'ils assurent la conversion de la grandeur à mesurer qui est un effet physique en signal électrique, on peut citer parmi ces effets :

L'effet thermoélectrique : la thermoélectricité régit la conversion à l'état solide d'énergie électrique en énergie thermique et vice-versa, c'est à dire que si on soumet un circuit formé

de deux matériaux de natures différentes à deux gradients de température T_1 pour le premier matériau et T_2 pour le deuxième on peut récupérer aux extrémités des deux matériaux une tension, et si on garde une des deux températures à une valeur fixe on peut déduire l'autre température à partir de la tension délivrée par le système.

L'effet piézoélectrique : la piézoélectricité est l'apparition d'une polarisation électrique sous l'effet de contraintes mécaniques sur certains cristaux, céramiques ferroélectriques ou de matériaux polymères.

L'effet électrodynamique : l'effet électrodynamique se base sur la conversion de l'énergie en utilisant le couplage électromécanique par l'effet du champ magnétique ou du champ électrique, ce sont essentiellement des capteurs de vitesse.

Effet photoélectrique : l'effet photoélectrique est la transformation d'une grandeur lumineuse (plus précisément l'éclairement de la cellule) en courant, leur principe est la libération de paires électrons-trous sous l'influence d'une énergie fournie par les photons. Ils doivent être distingués des capteurs photoélectriques classiques dont le fonctionnement est tributaire d'une alimentation électrique.

Effet Hall : lorsqu'une plaque d'un matériau semi-conducteur, placée dans un champ magnétique est parcourue par un courant électrique, une tension V_H (tension de Hall) perpendiculaire à l'induction magnétique apparaît entre les faces de cette plaque. Les capteurs à base d'effet Hall sont utilisés pour déterminer des vitesses de rotation (capteurs de vitesse de rotation, capteurs de vitesse d'un véhicule par exemple). La tension V_H mesurée sur le générateur Hall est généralement de faibles valeurs (inférieure ou égale à quelques millivolts) ; elle doit de ce fait être amplifiée à l'aide d'un circuit intégré Hall pour une meilleure exploitation.

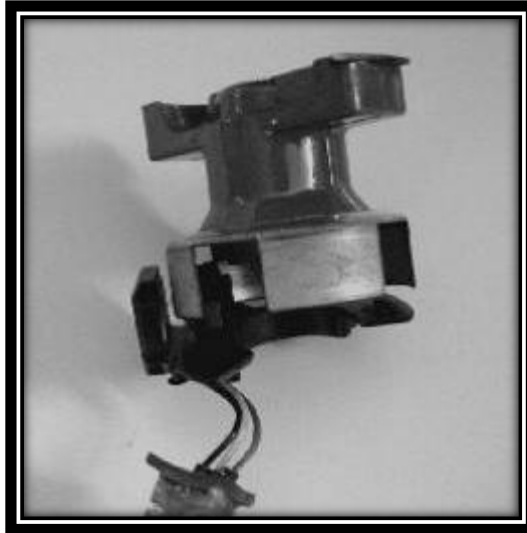


Figure1.7: exemple de capteur a effet Hall¹

1.8.2 Capteurs passifs

Les capteurs dont le signal électrique délivré est une variation d'impédance dont l'un des paramètres (résistivité, perméabilité magnétique, constante diélectrique) est sensible au mesurande sont dits passifs car ils nécessitent une source d'énergie électrique pour que l'on puisse lire le signal. Le circuit dans lequel ils sont incorporés est appelé conditionneur. Tous les transducteurs à impédance variable font partie de cette catégorie. On peut citer à titre d'exemple : Les potentiomètres rotatifs ou linéaires qui permettent de transformer la variation de résistance en un déplacement angulaire ou linéaire. Les jauges de contrainte qui permettent de transformer une microdéformation (allongement allant de 10^{-6} à 10^{-3}) en une variation de résistance. Deux paramètres interviennent dans ce cas : les variations dimensionnelles de la jauge ainsi que l'influence directe des déformations sur la résistivité du matériau (effet piézorésistif). Les thermistances métalliques permettant de prendre des mesures très précises de la température et les thermistances à semi-conducteurs qui présentent une sensibilité plus élevée. Le tableau suivant [02] représente les principes physiques de base des capteurs passifs²

¹ http://philippe.berger2.free.fr/automatique/cours/cpt/les_capteurs.htm

² Christophe Pijolat, étude des capteurs microélectroniques : actifs et passifs, Ecole Nationale Supérieure des Mines de Saint-Etienne (2004)

Mesurande	Caractéristique électrique sensible	Types de matériaux utilisés
Température	Résistivité	Métaux : platine, nickel, cuivre. Semi-conducteurs.
Très basse température	Constante diélectrique	Verres
Déformation	Résistivité Perméabilité magnétique	Alliages de nickel, silicium dopé. Alliage ferromagnétique
Position (aimant)	Résistivité	Matériaux magnéto résistants : bismuth, antimoniure d'indium
Humidité	Résistivité Constante diélectrique	Chlorure de lithium Alumine : polymères

Tableau 1.1 : capteurs passifs, principes physiques de base

Exemples de quelques capteurs utilisés en industrie :

1.9 Capteurs de couleurs TCS3200

Définition:

Le TCS3200 est un capteur conçu pour déterminer la couleur d'un objet, Il est constitué d'une matrice de minuscules photodiodes : 16 d'entre elles sont munies d'un filtre qui ne laisse passer que la lumière rouge, 16 autres ne captent que la lumière verte, 16 photodiodes ne captent que la lumière bleue, et les 16 dernières n'ont pas de filtre et captent donc la totalité de la lumière reçue³.

³ <http://electroniqueamateur.blogspot.com/2020/09/capteur-de-couleurs-tcs3200-gy-31-et.html>

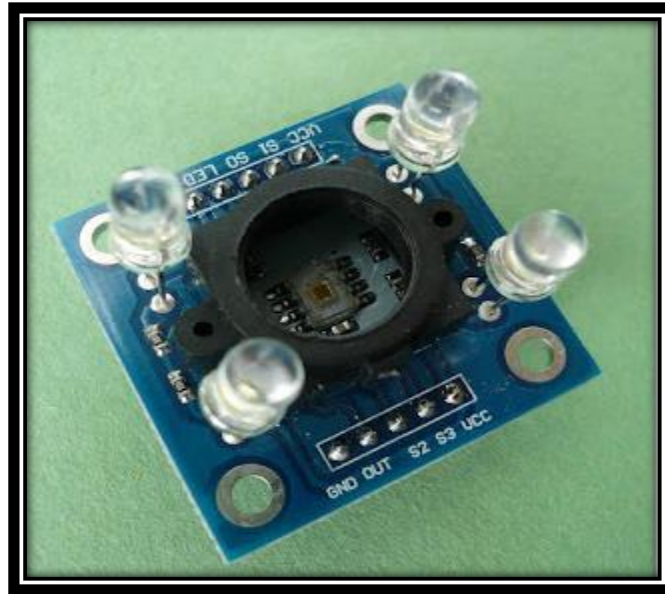


Figure1.8 : exemple de capteur tcs3200

En plus du capteur TCS3200, le module GY-31 comporte 4 LED qui permettent de bien éclairer la cible.

Le module comporte 10 connecteurs (mais 2 d'entre eux sont redondants) :

- 2 connecteurs Vcc : Alimentation située entre 2,7 V et 5,5 V
- 2 connecteurs GND : la masse, évidemment
- S0 et S1 : sélection de l'échelle de fréquence
- S2 et S3 : sélection des photodiodes actives
- OUT : signal de sortie
- LED : contrôle des 4 LED d'éclairage

C'est l'état logique des entrées S2 et S3 qui détermine quel groupe de photodiode est actif :

- S2 = 0 et S3 = 0 : Rouge
- S2 = 0 et S3 = 1 : Bleu
- S2 = 1 et S3 = 1 : Vert
- S2 = 1 et S3 = 0 : Sans filtre

Le signal de sortie est une onde carrée dont la fréquence est proportionnelle à l'intensité lumineuse désirée. Pour faciliter la mesure de cette fréquence, on peut la diviser par 5 ou par 50 grâce aux entrées S0 et S1:

- S0 =1 et S1 = 1: 100% de la fréquence (environ 600 kHz)
- S0 = 1 et S1 = 0: 20% de la fréquence (environ 120 kHz)
- S0 = 0 et S1 = 1: 2% de la fréquence (environ 12 kHz)
- S0 = 0 et S1 = 0: capteur inactif.
-

Puisque j'ai choisi de mesurer la demi-période du signal au moyen de la fonction `pulseIn()`, il m'a semblé préférable de travailler à l'échelle 2%, ce qui m'a donné des mesures de quelques centaines de microsecondes.

Les LED seront allumées par défaut si on ne branche rien à l'entrée LED du module. Si on désire les éteindre, il s'agit de régler cette entrée au niveau logique 0.

Pour un même objet coloré, les résultats obtenus dépendent beaucoup de la lumière ambiante et de la distance entre le capteurs et l'objet. Dans le but d'uniformiser mes résultats, j'ai placé mon capteur au fond d'un réceptacle noir qui bloquait la lumière ambiante, et qui me permettait de placer l'échantillon coloré à exactement 3 cm de distance par rapport aux capteurs.

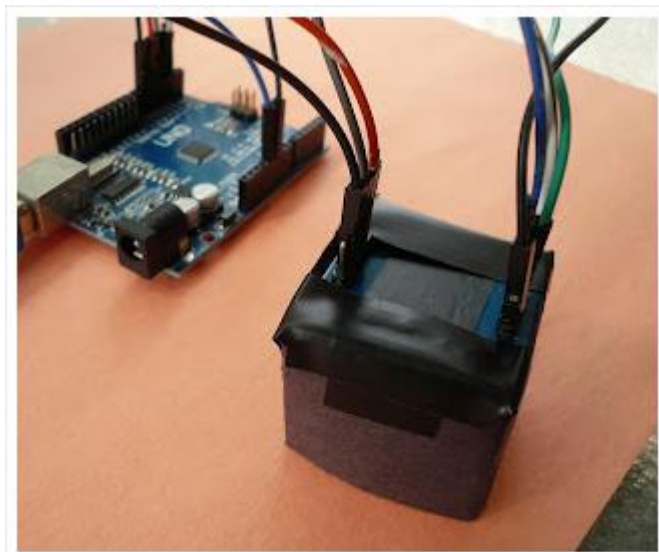


Figure1.9 : exemple de capteur de couleur liée avec Arduino

1.9.1 Description :

Capteurs de couleur programmable Joy-It basé sur le circuit RGB TCS3200 et 4LED blanches. Les capteurs délivre une fréquence en fonction de la couleur et est constitué de 4 filtres (un filtre pour chaque couleur : rouge, vert, bleu et sans filtre).

Le module se raccorde sur les E/S digitales d'une carte compatible Arduino pour la sélection du filtre et la mesure de la fréquence de sortie.

Manuel d'utilisation et exemple de programme en français (voir onglet Fiche technique).

Alimentation: 2,7 à 5,5 Vcc

Sorties: digitales

Erreur de non-linéarité: 0,2% à 50 kHz

Coefficient de T°: 200 ppm

Dimensions: Ø 37 x 18 mm

1.9.2 Fonctionnement :

Ça fonctionne vraiment bien pour différencier deux objets dont les couleurs sont vives et très différentes l'une de l'autre.

Même en prenant des précautions pour assurer une certaine uniformité des conditions de mesure (même carton coloré, écran bloquant la lumière ambiante, même distance entre les capteurs et le carton), les mesures pour une même couleur peuvent varier de plusieurs dizaines de microsecondes d'un essai à l'autre (surtout si la couleur est foncée). Il est peut-être possible d'obtenir de meilleurs résultats en améliorant la procédure pour mesurer la fréquence, ou en utilisant un microcontrôleur plus rapide.

1.10 Capteurs de proximité :

Définition :

Les capteurs de proximité ou « détecteurs de présence » sont des dispositifs autrefois mécaniques, mais aujourd'hui de plus en plus caractérisés par l'absence de liaison mécanique entre le dispositif de mesure et l'objet cible (personne, animal, objet animé tel qu'un véhicule). L'interaction entre le capteurs et sa « cible » est alors réalisée par l'intermédiaire d'une caméra associée à un système d'analyse de l'image, ou plus souvent d'un champ (magnétique, électrique, électromagnétique) ou d'un capteurs.

Selon les capteurs, objets et situation, l'objet détecté doit être plus ou moins proche des capteurs ou illuminé par une source rayonnante (éventuellement non visible, par exemple dans l'infrarouge).

Les capteurs de proximité sont utilisés soit en mode analogique, soit en mode binaire. Dans le premier cas, l'amplitude du signal est une fonction de la position relative de l'objet cible ; dans le second cas, le signal ne peut avoir que deux niveaux (haut et bas), selon que l'objet est présent à proximité ou non du capteur inductif.

1.11 Capteurs de proximité (ultrasons) :

Définition :

Le principe des ultrasons

Les capteurs à ultrasons ont établi de nouvelles références en automatisation⁴

1.11.1 Le principe des ultrasons :

Un capteur à ultrasons émet à intervalles réguliers de courtes impulsions sonores à haute fréquence. Ces impulsions se propagent dans l'air à la vitesse du son. Lorsqu'elles rencontrent un objet, elles se réfléchissent et reviennent sous forme d'écho au capteur. Celui-ci calcule alors la distance le séparant de la cible sur la base du temps écoulé entre l'émission du signal et la réception de l'écho.

⁴ <https://www.microsonic.de/fr/support/capteurs-%C3%A0-ultrasons/principe.htm>

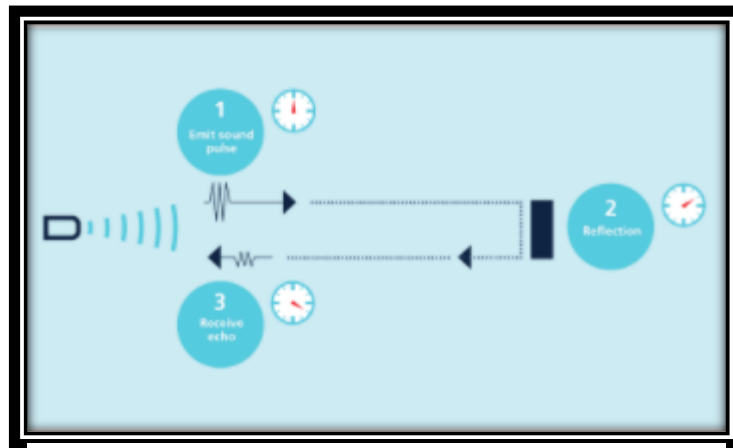


Figure1.10 : fonctionnement de capteur ultrasons

La distance étant déterminée par le temps de propagation des ultrasons et non par leur intensité, les capteurs à ultrasons conviennent parfaitement à une suppression d'arrière-plan.

Pratiquement tous les matériaux qui reflètent le son peuvent être détectés et ce, quelle que soit leur couleur. Même les matériaux transparents ou les feuilles minces ne représentent aucun problème pour un capteur à ultrasons.

Les capteurs à ultrasons micro Sonic sont disponibles pour des portées de 20 mm à 10 m et, du fait même de leur principe, donnent la valeur mesurée au millimètre près. Certains capteurs peuvent même atteindre une précision de 0,025 MM.

Les capteurs à ultrasons peuvent voir à travers l'air chargé en poussières et les brouillards d'encre. Même les dépôts minces sur la membrane du capteur ne nuisent pas à son fonctionnement.

Les capteurs avec une zone morte de seulement 20 mm et un faisceau extrêmement étroit trouvent aujourd'hui des applications inédites : mesure du niveau de remplissage des orifices de plaques d'analyses ou de tubes à essai, ainsi que la détection des petites

bouteilles dans l'industrie de l'emballage. Ils peuvent être mis en œuvre avec facilité. Même les fils minces sont détectés de manière fiable.

1.11.2 Applications des capteurs à ultrasons

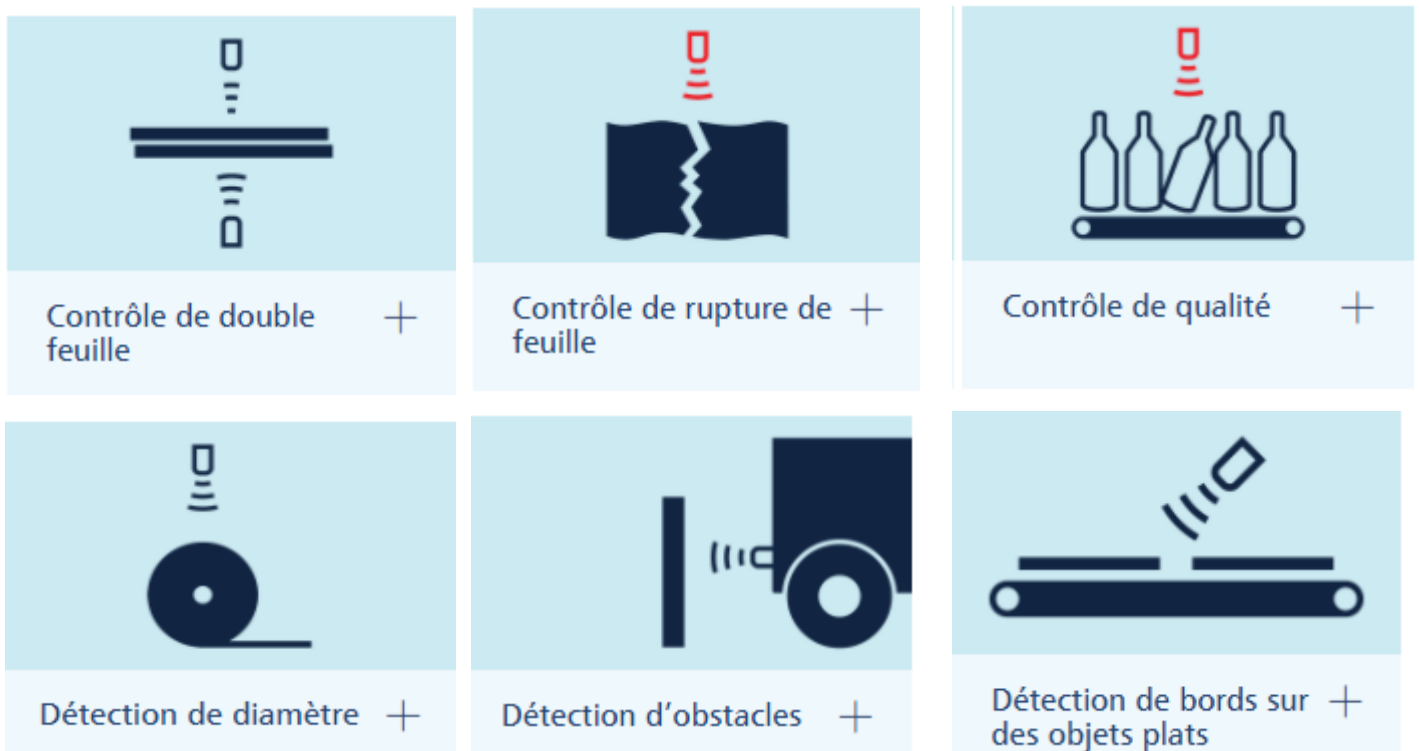


Figure1.11 : exemple d'application d'ultrasons

1.11.3 Principes de détection

La plupart des détecteurs à ultrasons reposent sur le principe de la mesure du parcours du son entre l'émission et la réception (détecteurs de proximité). Dans le cas du principe des barrières sera déterminée la distance du détecteur au réflecteur (barrière réflex) ou à un objet (barrière simple) dans la plage de mesure.

1.11.4 Détecteurs à ultrasons

Les détecteurs de proximité à ultrasons représentent la manière la plus simple de détecter des objets par ultrasons. L'émetteur et le récepteur sont intégrés dans un boîtier. Les ultrasons sont directement réfléchis vers le récepteur par l'objet détecté. Les détecteurs à

ultrasons avec la fonction « Teach-in » se différencient des types conventionnels par leur capacité d'utilisation plus aisée et plus diversifiée par simple pression sur une touche.

1.12 Capteurs de thermocouple

Définition :

Un thermocouple est un capteur qui permet de mesurer la température. Il est composé de deux métaux différents, soudés entre eux à une extrémité. La définition du thermocouple est souvent associée à son inventeur : le physicien allemand, Johann Seebeck⁵.

1.12.1 Qu'est-ce qu'un thermocouple ?

Un thermocouple est donc un instrument de mesure thermique. Il permet de connaître la température d'une pièce ou d'une surface, par exemple. Pour calculer la température, il se base sur une propriété physique : l'effet Seebeck.

Comment fonctionne-t-il ? Ses fils de métaux sont soudés en un point de la sonde thermocouple. Ce point, appelé "soudure chaude", est mis en contact avec l'élément dont on veut mesurer la température. La "soudure froide", l'autre extrémité, est maintenue à une température de référence, 0°C, par exemple. Se crée alors une tension entre la soudure froide et la soudure chaude permettant la mesure de la température.

1.12.2 Les différents types de thermocouple

Il existe différents types de thermocouples. En fonction des métaux utilisés, la sonde thermocouple mesurera des plages de température plus ou moins grandes. Les plus courants dans le commerce sont :

- Le thermocouple de Type K : composé de chromel et d'allume, il est indiqué pour des plages de température comprises entre -200 °C à 1250°C
- Le thermocouple de Type T : composé en cuivre et constantan qui mesure des températures entre -200 à 350°C
- Le thermocouple de Type J : en fer et constantan pour des températures entre 0 et 750°C.

⁵ <https://gaz-tarif-reglemente.fr/lexique-gaz/definition-thermocouple.html>

Le thermocouple peut s'utiliser dans l'industrie mais aussi pour un usage domestique.

1.12.3 Zoom sur la sécurité par thermocouple

Dans une maison, une sonde thermocouple permet d'assurer la sécurité du réseau de gaz. Sur les équipements au gaz, la soudure chaude du thermocouple est chauffée par la flamme. Si cette dernière venait à s'éteindre par accident, la sonde enregistre une chaleur différente. A ce moment-là, le thermocouple coupe le gaz du chauffe-eau, du four ou de la gazinière. Ce dispositif permet de prévenir le risque d'explosion. De nos jours, la plupart des gazinières modernes en sont équipées.

1.13 Thermocouple max6675

Définition : Le MAX6675 un amplificateur pour thermocouple Type-K permettant de mesurer une température entre -50°C et $+700^{\circ}\text{C}$ sans aucun problème⁶.

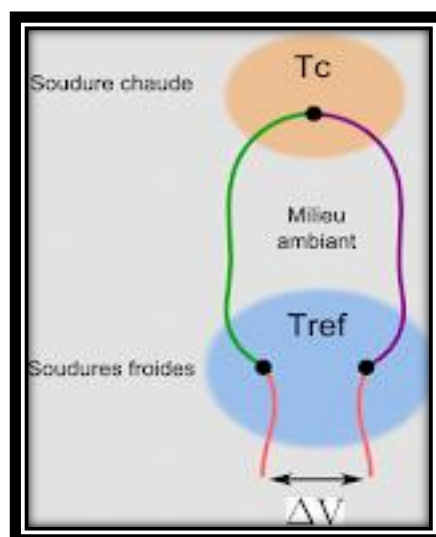


Figure1.12: exemple de capteur max6675

⁶ <https://arduino103.blogspot.com/2020/02/mod-tc-utiliser-un-thermocouple-type-k.html>

Thermocouple

Le thermocouple (aussi appelé CTE) est composé d'une soudure de deux matériaux différents (dit "point chaud" ou TC) exploitant l'effet Seebeck.

L'effet Seebeck produit une tension électrique (une différence de potentiel) en relation avec la température effective du point chaud.

En mesurant cette différence de potentiel avec des instruments de précision il est possible d'en déduire la température du "point chaud".

Le thermocouple se présente donc sous la forme d'une gaine en fibre de verre (isolant résistant à la température) terminé par une "perle de métal"

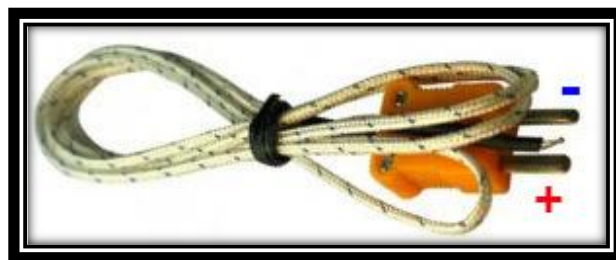


Figure1.13 : Exemple de thermocouple de Type-K

Le MAX6675

Le MAX6675 est un composant de Maxim Integrated. C'est un amplificateur pour thermocouple de Type-K.

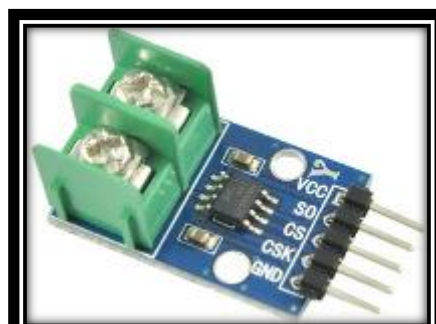


Figure1.14 : Breakout MAX6675 pour thermocouple type-K

Celui-ci effectue les mesures de potentiel sur le point chaud, applique un filtrage puis échantillonne l'information qui pourra ensuite être transmise de façon numérique vers un microcontrôleur (exploite une interface de type SPI).

1.13.1 Caractéristiques⁷:

- Plage de température d'usage jusqu'à 1024 °C.
- Forte résolution atteignant 0.25 °C.
- Compensation de jonction de température froide sur le module.
- Entrée haute impédance.
- Détection de rupture du thermocouple.
- Large plage de tensions de fonctionnement, de 3.0V à 5.5V
- Courant de travail 50mA.
- Température d'utilisation du module (la partie électronique, pas le thermocouple) -20°C à +85 °C.
- Interface numérique du module : GND VCC SCK CS SO.
- Tension d'alimentation de 3 - 5 V DC
- Taille du module 15 x 25 mm
- Diamètre du capteur 4.5mm, filetage 14.5 mm de long, écrou 9.8mm
- Fil résistant à la chaleur 0.95 m

1.13.2 Utilisation

- 2 Fils rouge / bleu à brancher sur le bornier à vis.
- Livré avec 5 fils de liaison Dupont (pour utilisation d'une carte Arduino, ou autre microcontrôleur).
- Fixation: écrou.

⁷ <http://tiptopboards.com/339-thermocouple-de-type-k-avec-module-d-amplificateur-num%C3%A9rique-max6675.html>

Le kit comprend :

- 1 thermocouple de type K + fil
- 1 amplificateur MAX6675 avec sortie numérique SPI
- 5 fils de câblage F/F Dupont

1.14 Conclusion

Nous avons présenté dans ce chapitre généralités sur les capteurs, les différents capteurs, ses caractéristiques et principes de fonctionnements qui correspond à notre travail. Dans le chapitre suivant nous présentons le choix du langage de développement qui nous facilite de créer notre interface.

Chapitre 2 Choix du langage de développement de l'interface

2.1 Introduction

On montre comment commander un Arduino depuis un script Python tournant sur un ordinateur. La communication entre l'Arduino et l'ordinateur se fait par la liaison série (câble USB). Pour la partie Python, le module pyserial est nécessaire (ce module est inclus dans pythonxy).

On considère dans un premier temps un exemple simple de communication, où des commandes standard de l'Arduino sont envoyées par le script Python.

On verra ensuite comment échanger des tableaux de données entre l'Arduino et Python.

Un programme muni d'une interface graphique fonctionne différemment d'un programme classique. Un programme classique est une succession séquentielle d'instructions. Il y a un début ou point d'entrée du programme et aucun événement extérieur ne vient troubler son déroulement. Avec une interface graphique, le point d'entrée du programme est masqué : il est pris en compte automatiquement. Du point de vue du programmeur, le programme a plusieurs points d'entrée : une simple fenêtre avec deux boutons propose deux façons de commencer et il faut prévoir une action associée à chaque bouton.

La conception d'une interface graphique se déroule généralement selon deux étapes. La première consiste à dessiner l'interface, c'est-à-dire choisir une position pour les objets de la fenêtre (boutons, zone de saisie, liste déroulante, ...). La seconde étape définit le fonctionnement de la fenêtre, c'est-à-dire associé à chaque objet des fonctions qui seront exécutées si un tel événement se réalise (pression d'un bouton, pression d'une touche, ...).

Pour le moment, nous allons supposer que ces deux étapes sont scindées même si elles sont parfois entremêlées lorsqu'un événement implique la modification de l'apparence de la fenêtre. La section qui suit décrit des objets que propose le module tkinter. La section suivante présente la manière de les disposer dans une fenêtre. La section d'après décrit les événements et le moyen de les relier à des fonctions du programme. Ce chapitre se termine par quelques constructions courantes à propos des interfaces graphiques.

2.2 Arduino

Définition

Arduino® est un ensemble matériel et logiciel qui permet d'apprendre l'électronique (en s'amusant) tout en se familiarisant avec la programmation informatique.

Arduino est en source libre ; vous pouvez donc télécharger le schéma d'origine et l'utiliser pour élaborer votre propre carte et la vendre sans payer des droits d'auteur⁸.

Qu'est-ce que le matériel Arduino ?

Ce sont des cartes électroniques programmables (donc dotées d'un processeur et de mémoire) sur lesquelles nous pouvons brancher des capteurs de température, d'humidité, de vibration ou de lumière, une caméra, des boutons, des potentiomètres de réglage, des contacts électriques...

Il y a aussi des connecteurs pour brancher des LED, des moteurs, des relais, des afficheurs, un écran...

Une carte Arduino est un cerveau qui permet de rendre intelligent des systèmes électroniques et d'animer des dispositifs mécaniques.

L'image ci-dessous montre une carte Arduino Uno qui est très utilisée pour débiter.

⁸ <https://www.positron-libre.com/electronique/arduino/arduino.php>

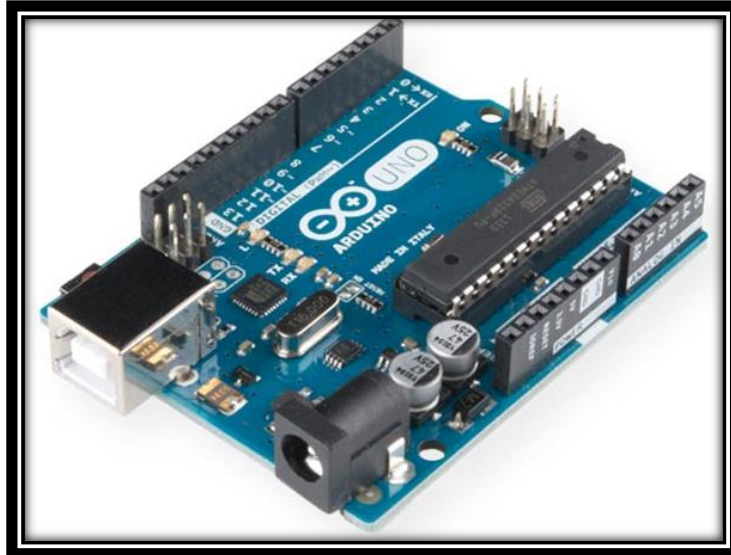


Figure2.1: Une carte de commande Arduino Uno avec ses connecteurs.

Dans les écrits en rapport avec Arduino vous allez souvent voir les mots « microprocesseur », « microcontrôleur », « MCU », « AVR », « ATmega168 », « ARM Cortex-M3 » ...

D'une manière très simplifiée : tous ces termes désignent un processeur.

Le processeur est l'unité de calcul (CPU) contenu à l'intérieur du circuit intégré désigné par l'un des termes précédemment cité (exemple : MCU, ATmega168...)

2.2.1 Qu'est-ce que le logiciel Arduino ?

Les créateurs de 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 Développement Environnement ou Environnement de Développement « Intégré » en français (donc EDI).

L'IDE Arduino est le logiciel qui permet de programmer les cartes Arduino.

L'IDE affiche une fenêtre graphique qui contient un éditeur de texte et tous les outils nécessaires à l'activité de programmation.

Vous pouvez donc saisir votre programme, l'enregistrer, le compiler, le vérifier, le transférer sur une carte Arduino...

A la date de rédaction de cette page, la version la plus récente de l'IDE Arduino est la 1.8.10.

L'aspect est à peu près identique sur chaque plate-forme (Windows, Mac et Linux). L'image suivante montre l'écran initial qui apparaît au lancement de l'IDE.

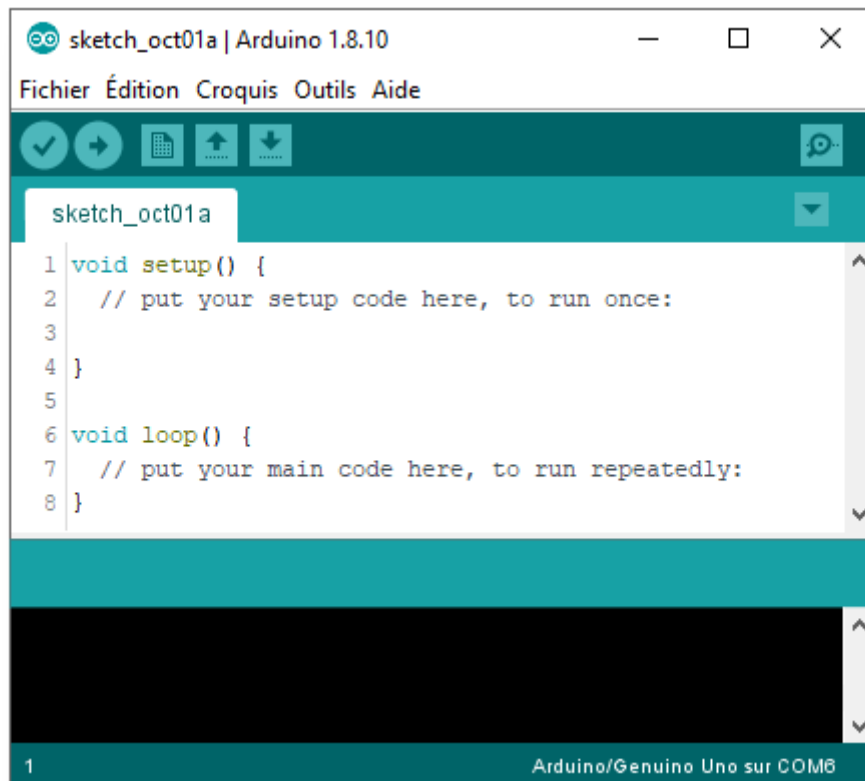


Figure2.2:l'interface de l'application Arduino

2.2.2 Types de cartes Arduino

Au fil des ans, les concepteurs d'Arduino.cc ont mis au point un certain nombre de modèles de cartes. La première carte Arduino, la Diecimila, est sortie en 2007. Et depuis, la famille Arduino a évolué pour tirer parti des différents types de micro-processeurs Atmel.

La Due, sortie en 2012, est la première Arduino à utiliser un processeur ARM Cortex-M3 32 bits. Elle se démarque du reste de la famille en termes de puissance de traitement et de configuration du brochage de la carte.

D'autres cartes, comme la LilyPad et la Nano, n'ont pas le même brochage également et sont destinées à une gamme différente d'applications plutôt « mobiles ».

Dans le cas de la LilyPad c'est pour une intégration facile dans les vêtements et les tissus. L'Esplora intègre des capteurs et des actionneurs et la taille compacte des Mini, Micro et Nano les prédestinent à des applications miniatures, légères et discrètes.

Lorsque plusieurs types de microcontrôleurs sont indiqués, cela signifie qu'une première version a été produite avec le premier type et plus tard avec l'autre (généralement plus puissant).

Par exemple, une version plus ancienne du Duemilanove aura un ATmega168, alors que les modèles plus récents auront l'ATmega328. Fonctionnellement, l'ATmega168 et l'ATmega328 sont identiques, mais l'ATmega328 a davantage de mémoire interne.

Les derniers ajouts à la famille Arduino, Leonardo, Esplora, Micro et Yún, utilisent tous le microcontrôleur ATmega32U4. Si ce dernier est similaire à un ATmega328, il intègre également un composant d'interface série USB, ce qui élimine un circuit intégré (routage plus facile) comme celui présent sur les cartes Uno et Duemilanove (ATmega16U2, FT232RL).

Que peut-on faire avec un Arduino ?

En plus de la facilité de programmation rendue possible par l'IDE Arduino, l'autre grande caractéristique d'une Arduino est la capacité du microcontrôleur sur lequel elle est basée. Avec quelques shields supplémentaires facilement disponibles, un large choix de modules capteurs économiques et d'actionneurs, il n'y a vraiment pas beaucoup de choses que vous ne puissiez faire avec une Arduino.

La condition est de garder à l'esprit quelques contraintes de base : mémoire, fréquence d'horloge, courants de sortie des périphériques et niveaux tensions.

Voici quelques applications possibles pour une Arduino :

- Mesure et détection
 - Station météorologique automatisée,
 - détecteur de foudre,
 - suivi du soleil pour orientation des panneaux solaires,
 - moniteur de radiation,
 - détecteur automatique de la faune,

- système de sécurité domestique ou professionnel.
- Contrôle
 - Petits robots,
 - maquette de fusée ou d'avion,
 - drones multi-rotor,
 - CNC simple pour petites machines-outils.
- Automatisation
 - Serre automatisée,
 - aquarium automatisé,
 - robot navette d'échantillon de laboratoire,
 - enceinte thermique de précision (couveuse, yaourtière, étuve, séchoir...),
 - système de test électronique automatisé.
- Art
 - contrôle d'éclairage et sonore dynamique,
 - structures cinématiques,
 - œuvre d'art.

2.3 L'interface graphique

Définition

L'interface graphique désigne la manière dont est présenté un logiciel à l'écran pour l'utilisateur. C'est le positionnement des éléments : menus, boutons, fonctionnalités dans la fenêtre. Une interface graphique bien conçue est ergonomique et intuitive afin que l'utilisateur la comprenne tout de suite⁹.

L'interface graphique est le langage d'échange entre l'humain (vous) et la machine (votre ordinateur). Votre ordinateur affiche à l'écran des éléments que vous comprenez et que vous interprétez. Chaque système dispose de sa propre interface graphique :

- Le tableau de bord de votre voiture

⁹ <https://cours-informatique-gratuit.fr/dictionnaire/interface-graphique/>

- Le distributeur de billet à la banque
- Votre téléviseur
- Votre ordinateur
- Votre tablette
- Votre smartphone

La personne en charge de réaliser cette interface est le designer, parfois également appelé l'expert en expérience utilisateur. C'est lui qui définit l'apparence d'une interface à partir des besoins du système. Une interface se doit d'être intuitive et ergonomique.

2.3.1 Le GUI « Graphical User Interface »

Définition

Signifie "Interface utilisateur graphique" et se prononce "gluant". C'est un Interface utilisateur qui comprend des éléments graphiques, tels que fenêtres, icônes et des boutons. Le terme a été créé dans les 1970 pour distinguer les interfaces graphiques des interfaces à base de texte, telles que interfaces de ligne de commande. Cependant, aujourd'hui presque tous numérique les interfaces sont des interfaces graphiques.

La première interface graphique disponible dans le commerce, appelée "PARC", a été développée par Xerox. Il était utilisé par le système d'information Xerox 8010, publié sous 1981. Après que Steve Jobs ait vu le interface Au cours d'une tournée chez Xerox, il a demandé à son équipe d'Apple de développer un système d'exploitation similaire. Le système d'exploitation graphique d'Apple était inclus dans la Macintosh, qui a été publié dans 1984. Microsoft a publié son premier système d'exploitation basé sur une interface graphique, Windows 1.0, dans 1985¹⁰.

Pendant plusieurs décennies, les interfaces graphiques ont été contrôlées exclusivement par un souris ainsi que d'un clavier. Alors que ces types de des dispositifs d'entrée sont suffisants pour ordinateurs de bureau, ils ne fonctionnent pas aussi bien pour les appareils mobiles, tels que smartphones et comprimés. Par conséquent, les systèmes d'exploitation mobiles

¹⁰ <https://techlib.fr/definition/gui.html>

sont conçus pour utiliser une écran tactile interface. De nombreux appareils mobiles peuvent désormais aussi être contrôlés par des commandes vocales.

Étant donné qu'il existe à présent de nombreux types de périphériques numériques, les interfaces graphiques doivent être conçues pour le type de périphérique approprié. Contribution. Par exemple, un système d'exploitation de bureau, tel que OS X, comprend un barre de menus et des fenêtres avec de petites icônes sur lesquelles vous pouvez facilement naviguer à l'aide d'une souris. Un système d'exploitation mobile, comme iOS, inclut des icônes plus grandes et prend en charge les commandes tactiles telles que glisser et pincer pour zoomer ou dézoomer. Les interfaces automobiles sont souvent conçues pour être contrôlées par des boutons et des boutons, tandis que les interfaces de télévision sont conçues pour fonctionner avec une télécommande. Quel que soit le type d'entrée, chacune de ces interfaces est considérée comme une interface graphique, car elle comprend des éléments graphiques.

REMARQUE : Les interfaces graphiques spécialisées qui utilisent la reconnaissance vocale et la détection de mouvement sont appelées interfaces utilisateur naturelles. NUIs.

2.4 Python

Définition :

Il s'agit d'un langage de programmation informatique comme il en existe pleins D'autres ayant vu sa première version voire le jour au début des années 90 Grâce à son créateur GUIDO VAN ROSSUM a fait d'autres projets notamment Comme un navigateur web « Gry » qui est sorti en 1999 mais vous allez me Dire, pourquoi ce langage, et pas un autre ? D'abord le python a la particularité d'être à la fois très simple à apprendre Donc parfait pour commencer, très puissant, il est énormément utilisé dans le Monde scientifique pour des calculs numériques par exemple, et enfin il Possédé une assez grosse base de communauté avec laquelle, vous pourrez

Interagir et échanger en cas de besoin. Les différents prérequis nécessaire pour développer en python Dans de bonnes conditions Il nous faudra deux composants :

1/-un interpréteur ou Environnement :

C'est tout simplement un endroit où ça s'exécute notre code python, car vous ne pouvez pas le faire comme ça sure votre ordinateur N'importe ou.

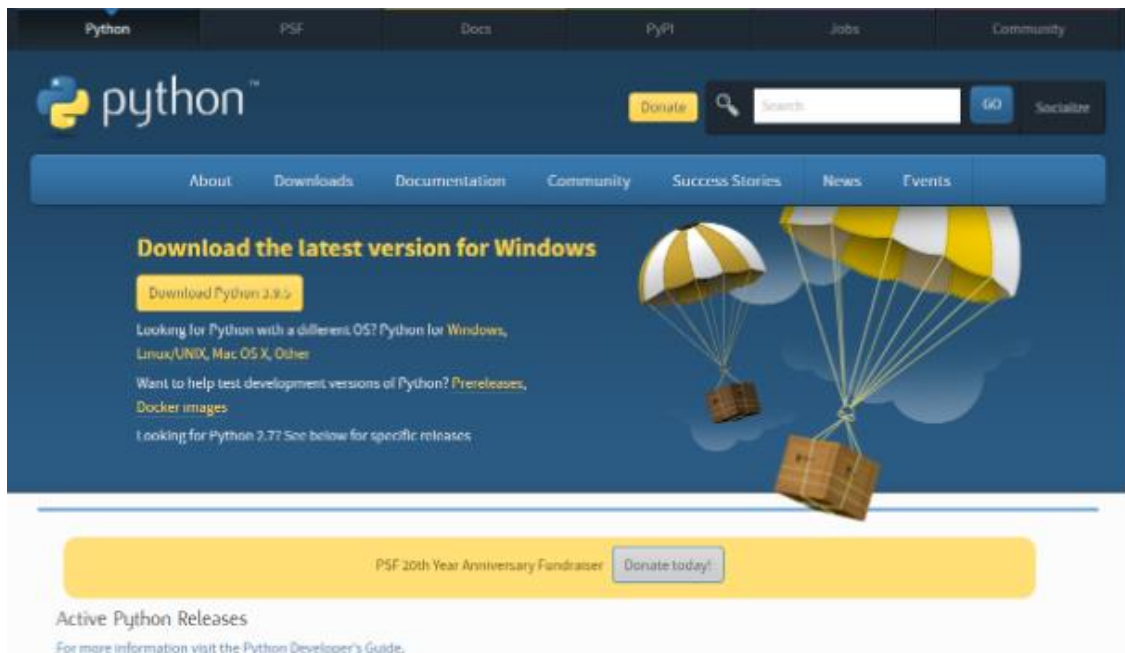


Figure2.3 : le site de téléchargement de python 3

2/-environnement de développement intègre « IDE » :

C'est un logiciel sur lequel vous allez pouvoir écrire votre code on Pourrait très bien utiliser un simple éditeur de texte pour faire du Python, comme (Notepad++ ; sublimText ou même le bloc-notes) Dans notre cas on est choisi de travailler sure de bonne base, on Prend quelque chose d'une qualité supérieure dont ma part on a Utilisé un IDE qui s'appelle PyCharm qui été créé par la société jetBrains qui fait pas mal d'environnement de développement pour Développeur, justement.

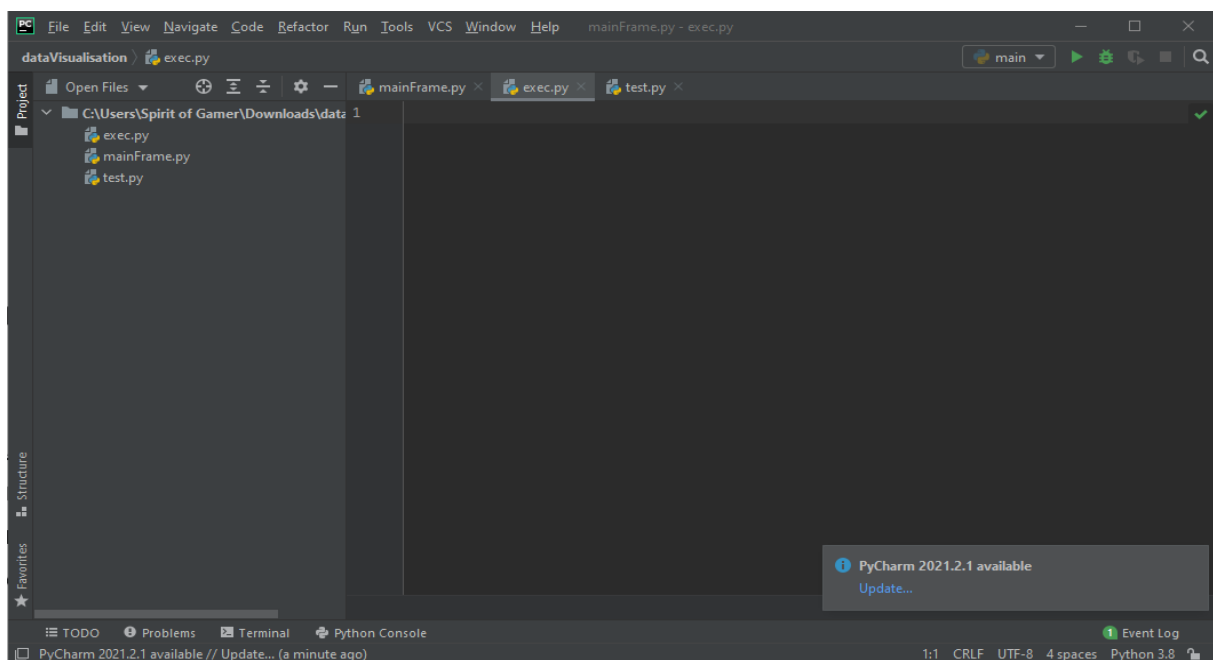


Figure2.4: l'interface d'IDE pycharm

Les étapes pour réaliser une interface graphique avec le python :

Installation des bibliothèques de python

2.4.1 Les bibliothèques dans python

Dans Python, une bibliothèque est un ensemble logiciel de modules (classes (types d'objets), fonctions, constantes, ...) ajoutant des possibilités étendues à Python : calcul numérique, graphisme, programmation internet ou réseau, formatage de texte, génération de documents, ...

Il en existe un très grand nombre, et c'est d'ailleurs une des grandes forces de Python. La plupart est regroupée dans le [itg-tooltip tooltip-content= »<p>stockage centralisé et organisé de données</p> »] dépôt [itg-tooltip] officiel de paquets logiciels (package repository) de Python appelé PyPi (Python Package Index)¹¹.

Installation d'une bibliothèque

Avec PIP

PIP est un utilitaire, fourni avec Python et qui permet d'installer simplement les bibliothèques déposées sur PyPi., ou bien à partir de fichiers Wheel (.whl).

PIP se trouve dans le sous dossier Scripts du dossier d'installation de Python.

Pour l'utiliser, il faut lancer une fenêtre de commande (ou terminal) : sous Windows, faire cmd.exe.

Attention, il faut avoir les droits administrateurs !

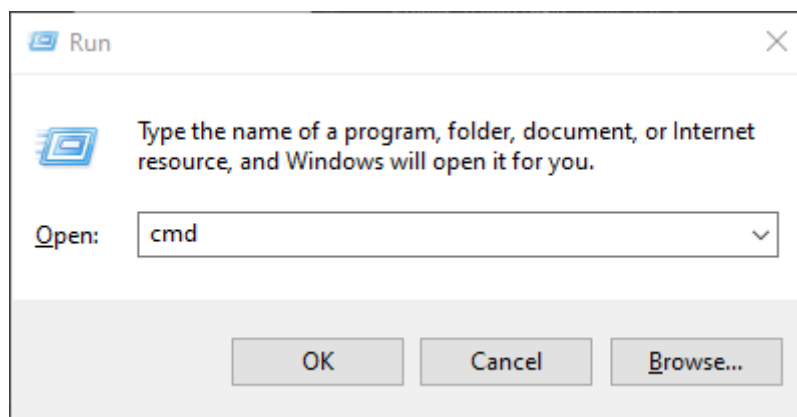
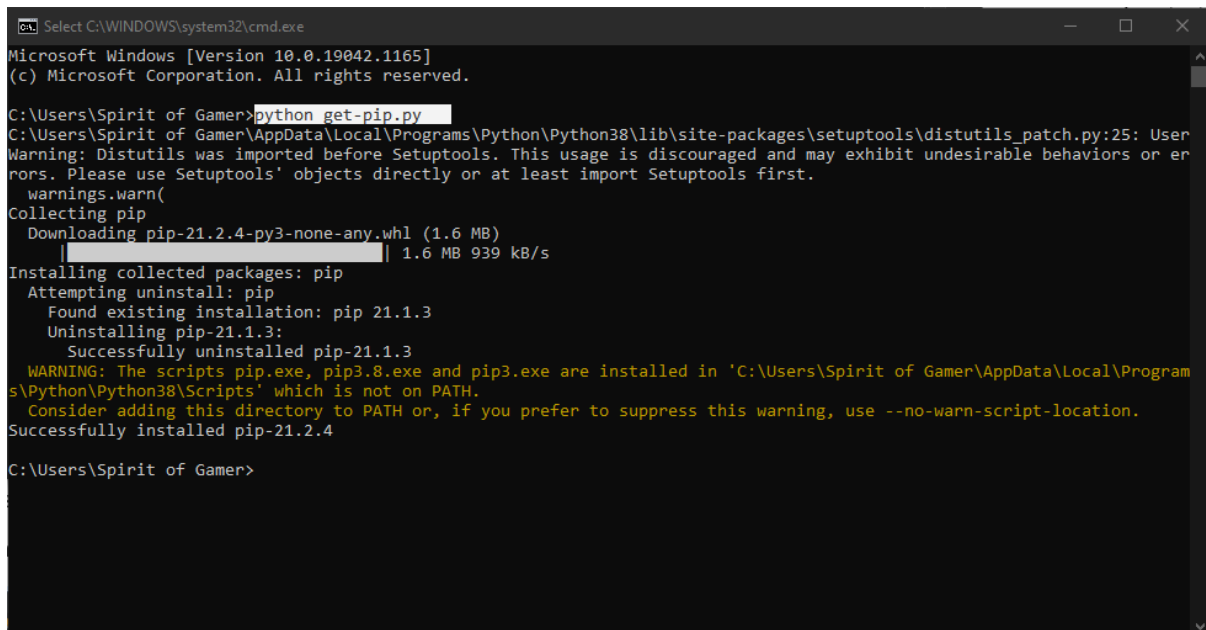


Figure2.5:la fenêtre Exécuter de Windows

¹¹ https://fr.wikibooks.org/wiki/Programmation_Python/Bibliothèques_pour_Python

2.4.2 Installer pip avec le cmd

Utiliser cette commande selon les étapes ci-dessus : `python get-pip.py`



```
Microsoft Windows [Version 10.0.19042.1165]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Spirit of Gamer>python get-pip.py
C:\Users\Spirit of Gamer\AppData\Local\Programs\Python\Python38\lib\site-packages\setuptools\distutils_patch.py:25: User
Warning: Distutils was imported before Setuptools. This usage is discouraged and may exhibit undesirable behaviors or er
rors. Please use Setuptools' objects directly or at least import Setuptools first.
  warnings.warn(
Collecting pip
  Downloading pip-21.2.4-py3-none-any.whl (1.6 MB)
    | 1.6 MB 939 kB/s
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 21.1.3
    Uninstalling pip-21.1.3:
      Successfully uninstalled pip-21.1.3
  WARNING: The scripts pip.exe, pip3.8.exe and pip3.exe are installed in 'C:\Users\Spirit of Gamer\AppData\Local\Program
s\Python\Python38\Scripts' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed pip-21.2.4

C:\Users\Spirit of Gamer>
```

Figure2.6: installation de pip bibliothèque avec la console de Windows

2.5 Tkinter

Tkinter (Tk interface) est un module intégré à la bibliothèque standard de Python, permettant de créer des interfaces graphiques :

- des fenêtres,
- des widgets (boutons, zones de texte, cases à cocher, ...),
- des évènements (clavier, souris, ...).

Tkinter est disponible sur Windows et la plupart des systèmes Unix : les interfaces créées avec Tkinter sont donc portables.

Les noms de Tkinter sont disponibles dans le module `tkinter`¹².

Le module Tkinter `{tkinter-intro}` est basé sur Tk, la librairie développée par J.~Ousterout pour faire des interfaces graphiques en Tcl `{tcltk}`, et disponible sur plusieurs plates-formes (dont X11, Mac OS X, MS-Windows). Tkinter est le standard de facto pour Python offrant une vision objet de Tk. Toutefois, Tcl est un langage où tout est chaînes de caractères. Il en résulte qu'un certain nombre de valeur utilisées dans le cas de Tkinter sont

¹² <http://infohost.nmt.edu/tcc/help/pubs/tkinter/web/index.html>

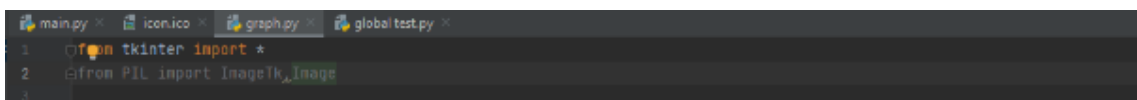
des chaînes de caractères (utilisées directement comme tel ou bien sous couvert de variables offertes par le module)¹³.

Comme tout programme avec une interface graphique, l'exécution est dirigée par les événements. Les interactions entre Python et l'interface graphique peut prendre plusieurs formes :

- Traitement GUI Python puis Tkinter puis Tk puis Librairie graphique.
- Événement graphique puis Tk puis Tkinter puis Traitements Python.

L'importation de cette module sure le code doit être avec le syntaxe suivant :

```
From tkinter import
```



```
main.py icon.ico graph.py global test.py  
1 from tkinter import *  
2 from PIL import ImageTk, Image  
3
```

Figure2.7:limportation de tkinter bibliothèque dans le code

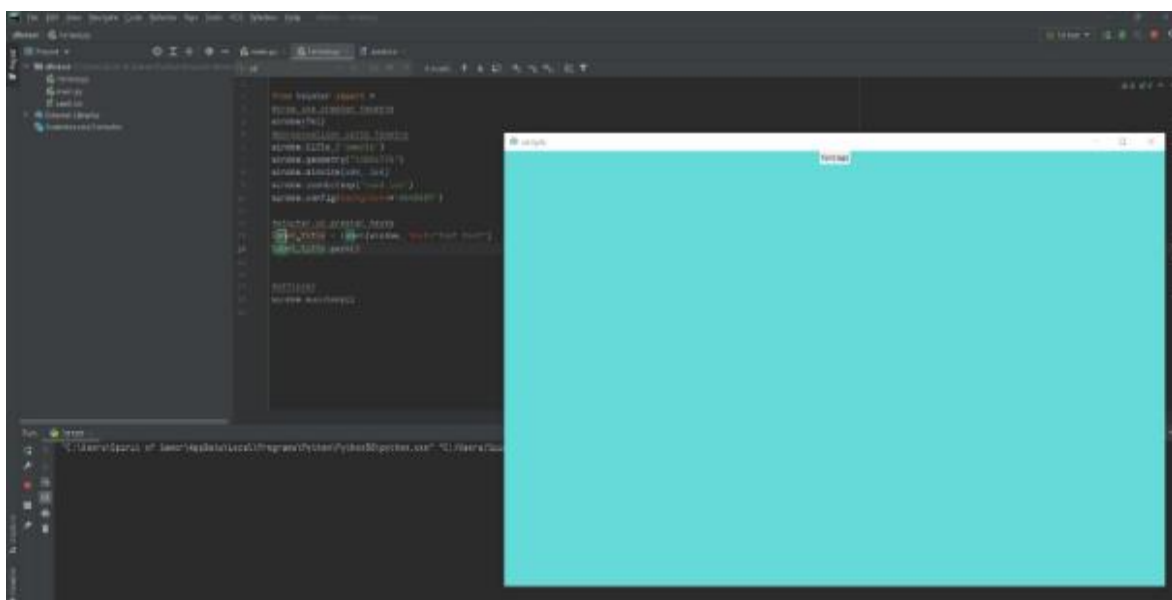


figure2.8:exemple de code d'une fenêtre avec tkinter

¹³ <https://www.fil.univ-lille1.fr/~marvie/python/chapitre6.html>

2.6 Numpy

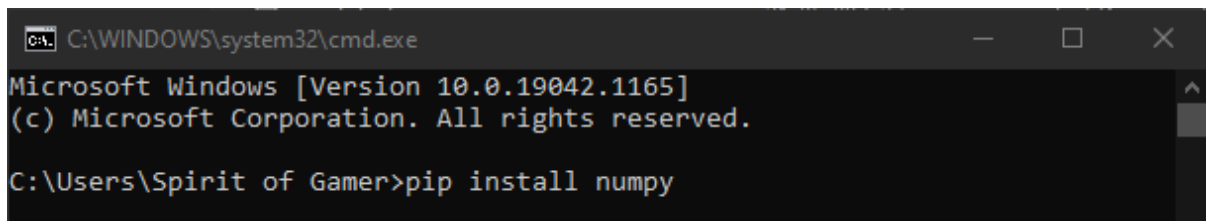
NumPy est une bibliothèque pour langage de programmation Python, destinée à manipuler des matrices ou tableaux multidimensionnels ainsi que des fonctions mathématiques opérant sur ces tableaux.

Plus précisément, cette bibliothèque logicielle libre et open source fournit de multiples fonctions permettant notamment de créer directement un tableau depuis un fichier ou au contraire de sauvegarder un tableau dans un fichier, et manipuler des vecteurs, matrices et polynômes.

NumPy est la base de SciPy, regroupement de bibliothèques Python autour du calcul scientifique ¹⁴

Install numpy

Installer numpy avec pip selon les étapes suivantes :



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.19042.1165]
(c) Microsoft Corporation. All rights reserved.
C:\Users\Spirit of Gamer>pip install numpy
```

Figure8.9: installation de bibliothèque numpy avec la console de Windows

Importer numpy dans le code selon la syntaxe suivant :

```
import numpy as np
```

¹⁴ <https://www.scipy.org/about.html>

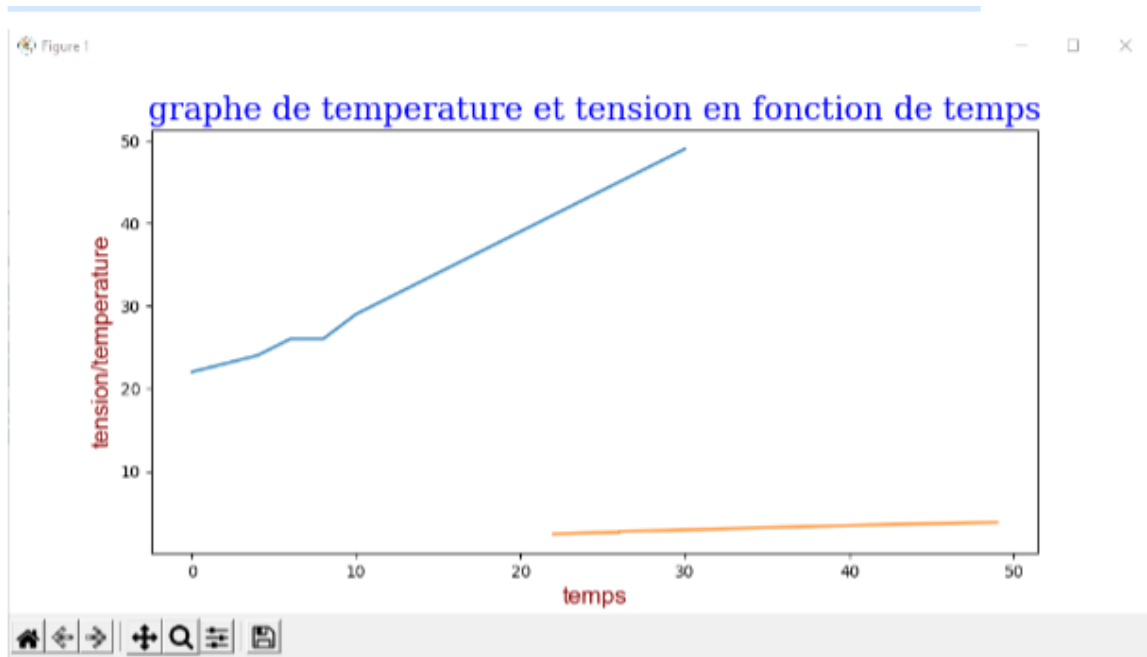


Figure2.10 : Exemple d'un graphe réalisé avec la bibliothèque numpy

2.7 matplotlib

Matplotlib est une bibliothèque du langage de programmation Python destinée à tracer et visualiser des données sous formes de graphiques ¹⁵ Elle peut être combinée avec les bibliothèques python de calcul scientifique NumPy et SciPy. Matplotlib est distribuée librement et gratuitement sous une licence de style BSD¹⁶. Sa version stable actuelle (la 2.0.1 en 2017) est compatible avec la version 3 de Python.

Plusieurs points rendent cette bibliothèque intéressante :

- Export possible en de nombreux formats matriciels (PNG, JPEG...) et vectoriels (PDF, SVG...)
- Documentation en ligne en quantité, nombreux exemples disponibles sur internet
- Forte communauté très active
- Interface pylab : reproduit fidèlement la syntaxe MATLAB
- Bibliothèque haut niveau : idéale pour le calcul interactif

¹⁵https://books.google.fr/books?id=4wMn27KSnTsC&printsec=frontcover&dq=Matplotlib&hl=fr&sa=X&ei=H87fUpvZCpSw7Aa9loDQBw&redir_esc=y#v=onepage&q=Matplotlib&f=false

¹⁶ <https://matplotlib.org/stable/users/license.html>

Install matplotlib

Suivez le chemin suivant :

Pycharm IDE>settings>project interpreter>add>matplotlib>install

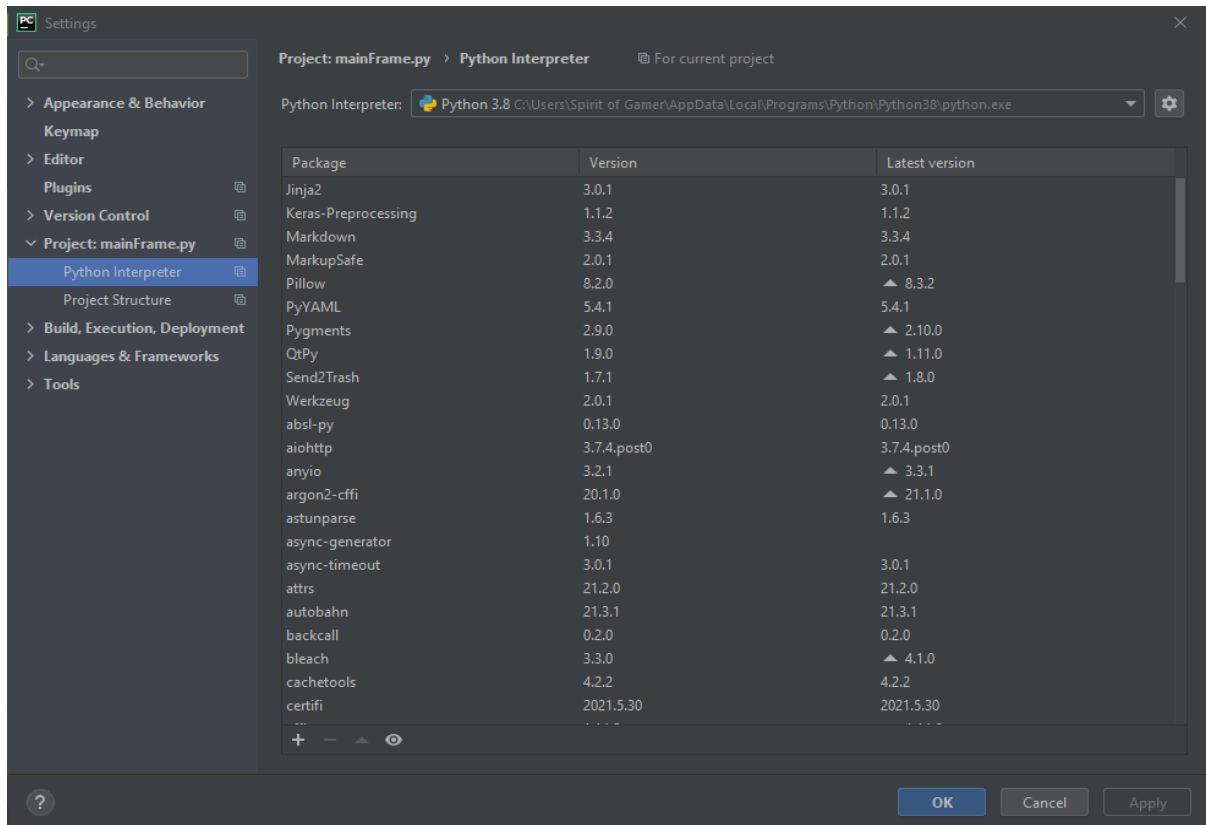


Figure2.11 : recherche de bibliothèque matplotlib dans l'IDE

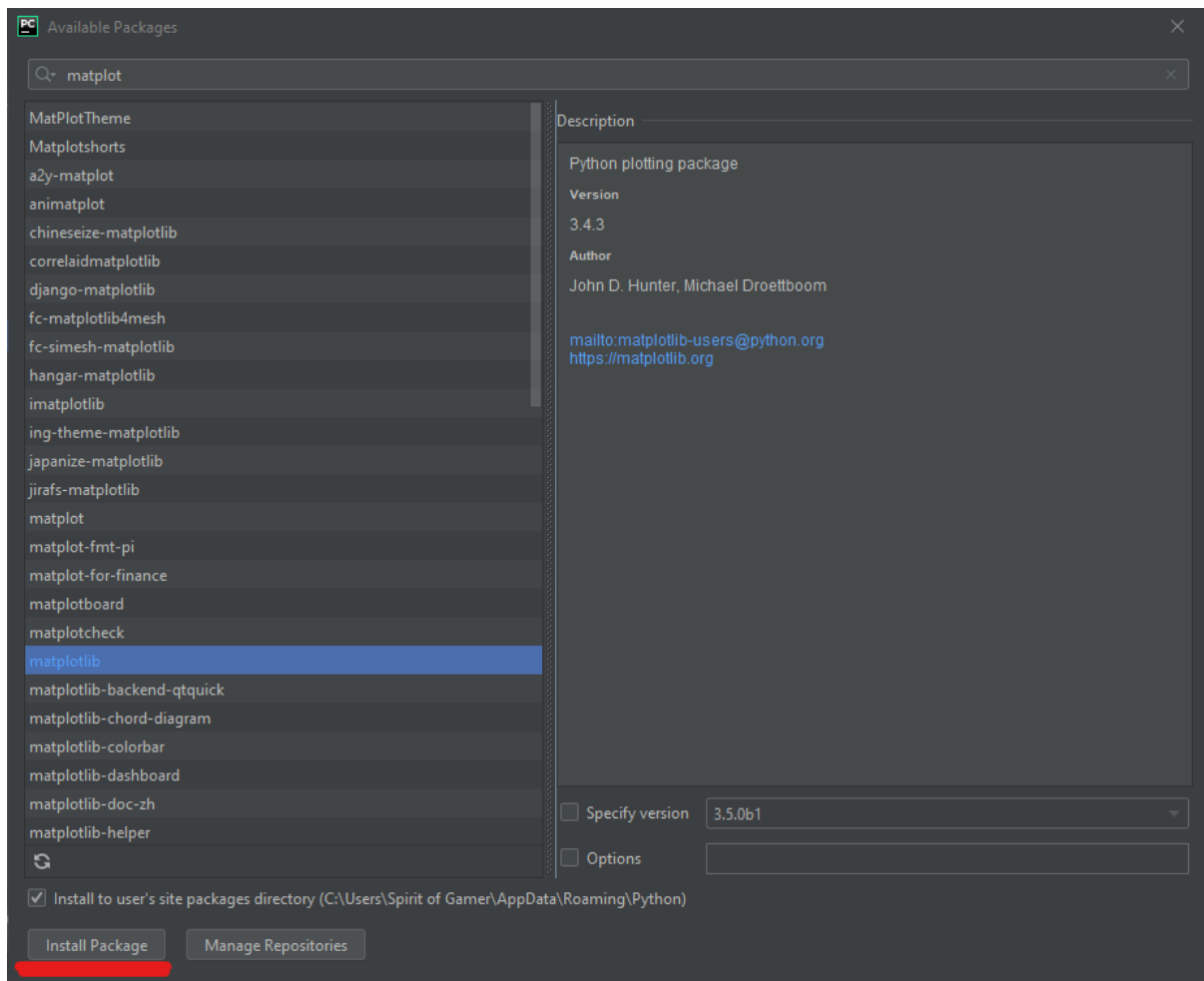


Figure2.12 : installation de bibliothèque matplotlib avec l'IDE

En utilisant matplotlib dans notre code selon la syntaxe suivante :

```
Import matplotlib.pyplot as plt
```

2.8 pyserial

Pyserial est une bibliothèque du langage de programmation Python d'accès au port série.

Ce module encapsule l'accès pour le port série. Il fournit des backends pour Python fonctionnant sous Windows, OSX, Linux, BSD (éventuellement tout système compatible

POSIX) et IronPython. Le module nommé "serial" sélectionne automatiquement le backend approprié¹⁷.

2.8.1 Install pyserial

Suivez le chemin suivant :

Pycharm IDE>settings>Project interpréter>add>pyserial>Install

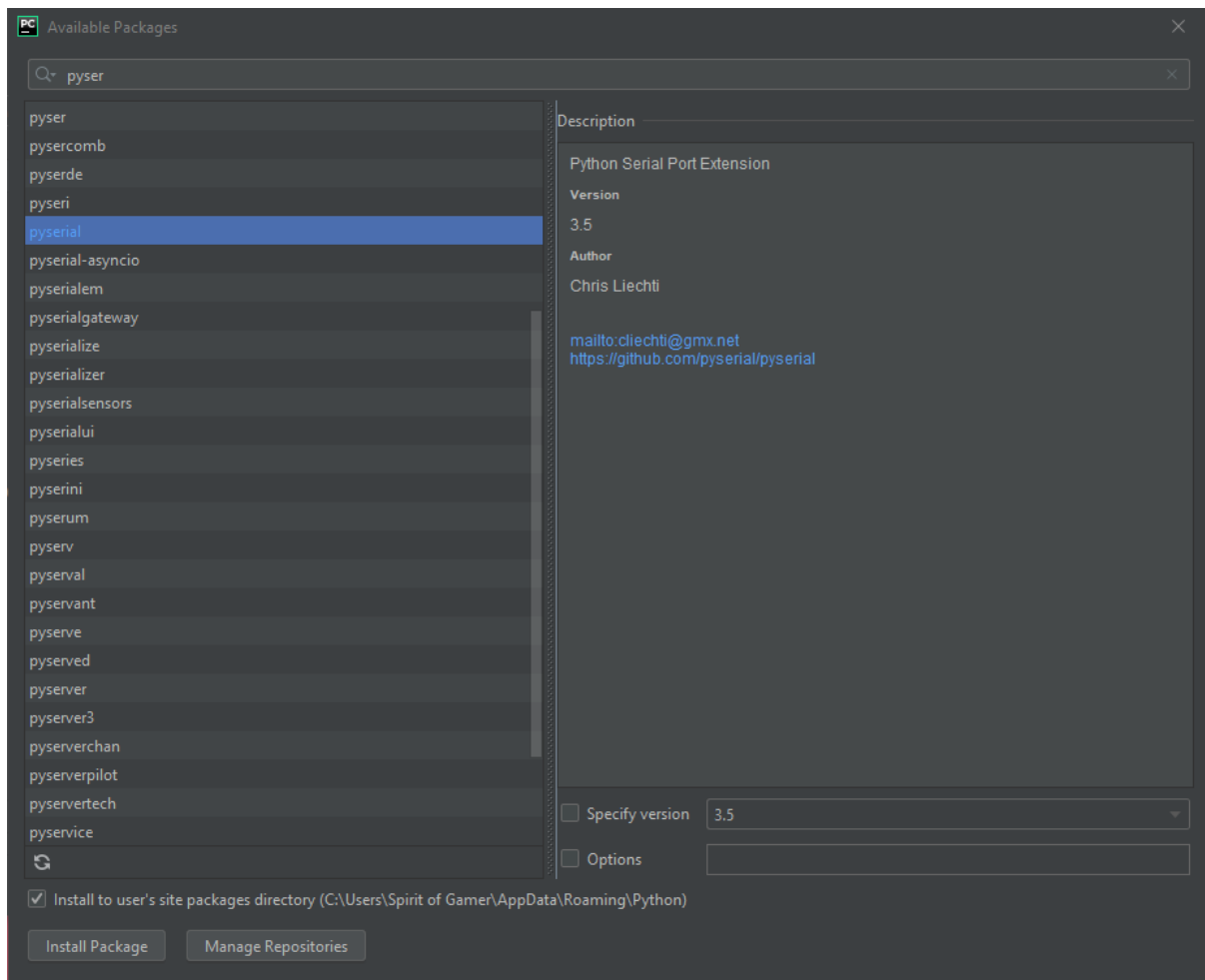


Figure2.13 : recherche de bibliothèque matplotlib dans l'IDE

En utilisant pyserial dans notre code selon la syntaxe suivante :

```
import time
import serial
```

¹⁷ <https://wiki.python.org/moin/PySerial>

Note : la syntaxe « import time » utilise pour afficher les résultats obtenant d'Arduino en temps réel.

La communication entre Arduino et python :

La carte Arduino permet de faire l'acquisition d'un signal analogique par l'intermédiaire d'un capteur et de le convertir en signal numérique grâce à son CAN (10 bits). Il est ensuite possible de transférer ces données par le port série vers l'ordinateur pour réaliser un traitement numérique avec Python. Capteur

conversion Traitement numérique du signal Echantillonnage et

Photorésistance | Carte d'acquisition Arduino | Python

Donc très schématiquement on se sert de l'interface de programmation d'Arduino pour écrire un petit programme qui explique à la carte comment faire l'acquisition (programme qui est ensuite téléverser sur la carte par le port série) puis on récupère les données via le port série pour en faire une analyse avec Python.

Code Python : Pour notre premier exemple, nous allons créer une liaison série pour que Python puisse communiquer avec la carte Arduino :

- Fermer le moniteur série coté Arduino, pour pouvoir établir une liaison avec Python
- Ouvrir un nouveau projet.
- Changer le nom du projet : Arduino-Python
- Recopier l'exemple ci-dessous en n'oubliant pas d'exécuter la cellule de code. Attention de bien indiquer le port sélectionné dans le menu Arduino (Outils -> Port série). Sous Windows : COM suivi d'un numéro (1, 2, 3, ...), sous linux : /dev/ttyACM suivi d'un numéro (0 ou 1 en général) ou /dev/ttyUSB0

La communication dans notre code doit être sous cette forme :

```
Import serial
Serial_port=Serial.Serial("COM5",baudrate=9600,timeout=1)
Serial_port.readline()
```

2.8.2 Principe

Le protocole Firmata permet de commander un Arduino depuis un programme tournant sur un ordinateur (en langage Python, Java, etc.). Le programme tournant sur l'Arduino reçoit des ordres correspondant aux commandes standards (par exemple digitalWrite) de la part du programme de l'Ordinateur.

Ce type d'interface est intéressant pour effectuer des tâches simples ne nécessitant pas de réaction en temps réel de la part de l'Arduino. Il peut aussi servir pour vérifier le fonctionnement d'un circuit électronique relié à l'Arduino, avant de mettre en place une solution en temps réel complète.

On propose ici une mise en œuvre de ce principe, avec un protocole élémentaire, qui permet de comprendre comment se fait la communication série. Le programme Arduino exécute sur demande l'une des commandes suivantes :

-pinMode

-digitalWrite

-digitalRead

-analogWrite

-analogRead

Les programmes présentés pourront être facilement complétés pour accomplir des tâches plus spécifiques (par exemple des lectures ou écritures séquentielles)¹⁸.

2.9 Les objets

Les interfaces graphiques sont composées d'objets ou widgets ou contrôles. Comme ils reçoivent des événements, en un sens, ce sont ces objets qui pilotent un programme ou qui le contrôlent.}. Ce paragraphe décrit les principales méthodes qui permettent de modifier le contenu et l'apparence des objets.

¹⁸ <https://www.f-legrand.fr/scidoc/docimg/sciphys/arduino/python/python.html>

Les exemples de codes des paragraphes qui suivent permettent de disposer les objets dans une fenêtre qui ne s'affichera pas sans les quelques lignes de code supplémentaires présentées au paragraphe Fenêtre principale et l'utilisation d'une méthode pack L'exemple suivant crée un objet :

```
zone_texte = tkinter.Label (text = "zone de texte")
```

Et pour l'afficher, il faut l'enrober :

```
Import tkinter      # import de tkinter
root = tkinter.Tk () # création de la fenêtre principale
# ...
obj = tkinter.Label (text = "zone de texte")
# ...
obj.pack ()        # on ajoute l'objet à la fenêtre principale
root.mainloop ()   # on affiche enfin la fenêtre principal et on attend
                    # les événements (souris, clic, clavier)
```

2.10 Zone de texte¶

Une zone de texte sert à insérer dans une fenêtre graphique une légende indiquant ce qu'il faut insérer dans une zone de saisie voisine comme le montre la figure qui suit. Une zone de texte correspond à la classe Label. Pour créer une zone de texte, il suffit d'écrire la ligne suivante :

```
zone_texte = tkinter.Label (text = "zone de texte")
```

Il est possible que le texte de cette zone de texte doive changer après quelques temps. Dans ce cas, il faut appeler la méthode config comme suit :

```
zone_texte = tkinter.Label (text = "premier texte")
# ...
# pour changer de texte
```



```
zone_texte.config (text = "second texte")
```

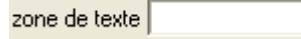


Figure2.14 : Exemple de zone de texte dans la fenêtre réalisé avec tkinter

La zone de texte ou Label est associée à une zone de saisie. La seconde image montre une zone de texte dans l'état DISABLED.

L'exemple suivant montre deux zones de texte. La seconde est grisée par rapport à la première. Pour obtenir cet état, il suffit d'utiliser l'instruction suivante :

```
zone_texte.config (state = tkinter.DISABLED)
```

Et pour revenir à un état normal :

```
zone_texte.config (state = tkinter.NORMAL)
```

Ces deux dernières options sont communes à tous les objets d'une interface graphique. Cette option sera rappelée au paragraphe Méthodes communes.

matplotlib

definition

matplotlib est une bibliothèque Python[14] capable de produire des graphes de qualité. matplotlib peut être utilisé dans des scripts Python, le shell Python et IPython, le notebook Jupyter, des serveurs d'application web et dans quatre outils d'interface graphique.

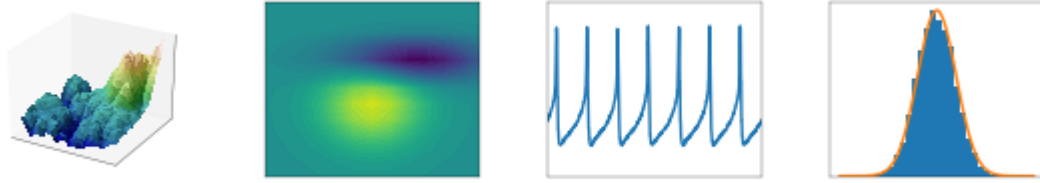


Figure2.15: les déferlants types de graphe réalisé par matplotlib

Matplotlib essaye de rendre les tâches simples “simples” et de rendre possible les choses compliquées. Vous pouvez générer des graphes, histogrammes, des spectres de puissance (lié à la transformée de Fourier), des graphiques à barres, des graphiques d’erreur, des nuages de dispersion, etc... en quelques lignes de code.

Pour des graphiques simples, le module `matplotlib.pyplot` fournit une interface comme MATLAB spécialement adaptée avec IPython.

2.11 Conclusion

Dans ce chapitre nous avons décrit les langages utilisés pour le code, ainsi que les IDE, les technologies nécessaires pour s’échanger les données entre les langages et nous avons terminé par quelques constructions courantes à propos des interfaces graphiques. Dans le chapitre suivant nous avons décrit la chaîne de conditionnement des capteurs.

Chapitre 3 : chaines de conditionnement des capteurs

3.1 Introduction

Dans ce chapitre on va parler au circuit de protection des capteurs (température max 6675, proximité ultrasons, couleur tcs3200) et comment protéger les captures par Arduino

3.2 Disjoncteurs

Définition :

Un disjoncteur est destiné à établir, supporter et interrompre des courants, sous sa tension assignée. Dans les conditions normales et anormales du réseau. Il est très généralement associé à un système de protection (relais), qui détecte le défaut et envoi des ordres au disjoncteur pour éliminer automatiquement le défaut ou pour remettre en service un circuit lorsque le défaut a été éliminé. Sa fonction principale est d'interrompre le flux de courant détecté lors d'un défaut. Le principe de base de tous les disjoncteurs est d'essayer de détecter le passage du courant par la valeur zéro et d'interrompre le flux de courant à ce moment. Le disjoncteur ne réussit pas souvent à interrompre le courant durant la première tentative, plusieurs cycles de la fréquence fondamentale du courant sont nécessaires pour

une interruption complète du flux de courant, ce qui affecte la vitesse du disjoncteur. Les disjoncteurs rapides utilisés dans la HT sont d'un cycle, par contre ceux utilisés dans la BT prennent 20 à 50 cycles pour ouvrir. De plus pour distinguer entre un défaut permanent et un défaut temporaire le concept d'auto-enclenchement est utilisé. Quand le disjoncteur déclenche il reste ouvert un certain temps ensuite il ferme automatiquement. Cette action permet au relais de vérifier si le défaut continu d'exister, et dans ce cas de déclencher de nouveau¹⁹.

Si le défaut a disparu, le relais ne fonctionne pas et la ligne va rester en service.

Le disjoncteur HT est caractérisé essentiellement par la technique utilisée pour la coupure :

- ◆ Les disjoncteurs à l'huile.
- ◆ Les disjoncteurs à air comprimé.
- ◆ Les disjoncteurs utilisant le gaz SF6.
- ◆ Les disjoncteurs à vide.

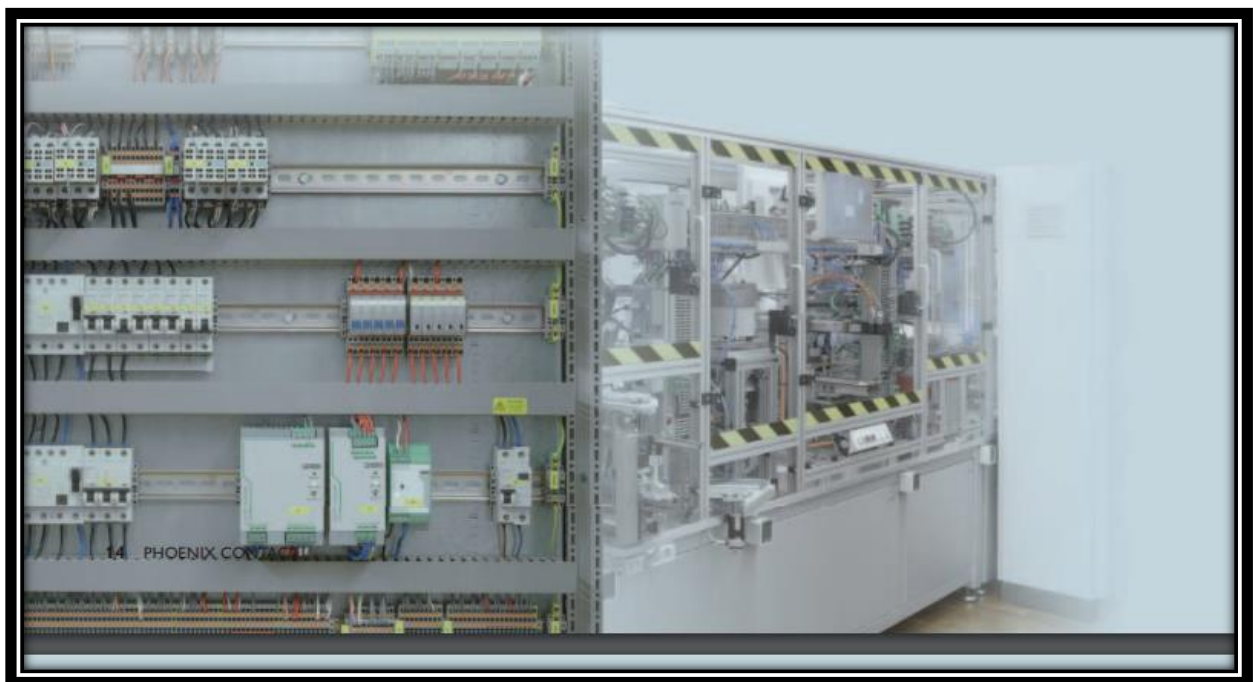


Figure3.1: exemple de disjoncteur

Principaux secteurs

¹⁹ <http://thesis.univ-biskra.dz/1873/3/chapitre%2001.pdf>

Les disjoncteurs de protection de circuits magnétothermiques sont utilisés dans le domaine des technologies de l'information, de la communication et des procédés. Grâce aux différentes courbes de déclenchement, ils sont polyvalents. Le réarmement et la signalisation à distance instantanée de l'état de fonctionnement garantissent une disponibilité élevée. Les disjoncteurs de protection de circuits électroniques sont utilisés dans les techniques de communication mais aussi d'automatisation. La limitation de courant active permet de conserver la tension de sortie sur le bloc d'alimentation, et tous les autres circuits restent en service. Les disjoncteurs de protection de circuits thermiques protègent de manière optimale les charges inductives contre les surcharges dans les systèmes de distribution des armoires électriques et des installations. De plus, ils sont insensibles aux courants de démarrage élevés, comme lors du démarrage d'un moteur ou d'un transformateur.

3.3 Relais de protection

Les relais de protection sont des appareils qui reçoivent un ou plusieurs informations à caractère analogique (courant, tension, puissance,... etc.) et le transmettent à un ordre de fermeture ou ouverture d'un circuit de commande lorsque ces informations reçues atteignent des valeurs supérieures ou inférieures à certaines limites qui sont fixées d'avance. En effet, le rôle des relais de protection consiste à détecter tout phénomène anormal pouvant se produire sur un réseau électrique tel que les défauts de court-circuit. Les relais peuvent être classés en trois familles (Figure)

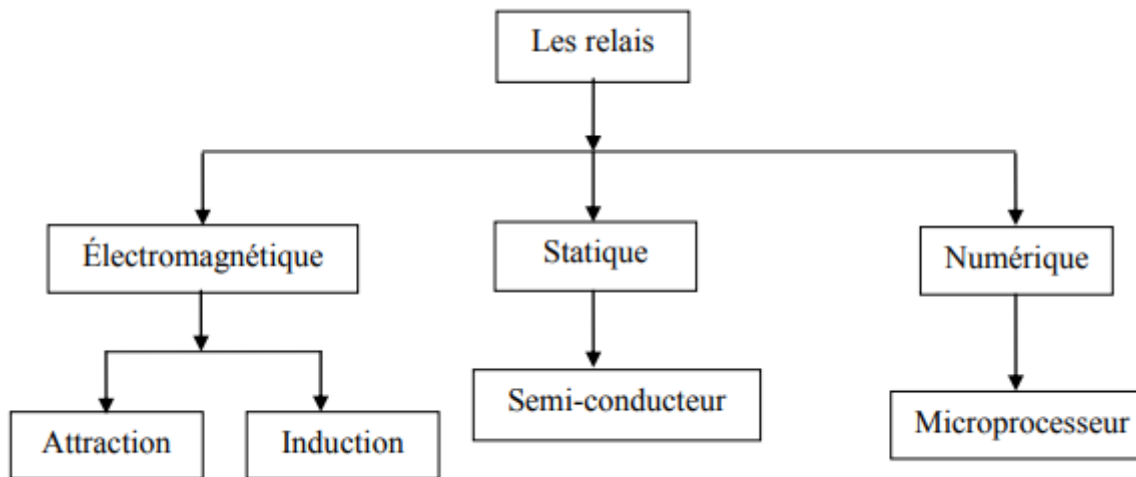


Figure3.2: Type des relais

Les relais de protection se présentent comme des produits spécifiques en fonction des applications. Les types standards sont : Sous station, jeu de barres, transformateur, moteur, condensateur, générateur²⁰.

3.3.1 Relais électromécaniques

Sont basés sur le principe d'un disque d'induction actionné par des bobines alimentées par les transformateurs de courant et de tension (Figure I.7). Un ressort de rappel réglable détermine la limite de l'action du disque sur un déclencheur. Les équipements électromécaniques sont des assemblages de fonctions : détection de seuils et temporisation. Ils avaient l'avantage d'être robustes, de fonctionner sans source d'énergie auxiliaire et d'être peu sensibles aux perturbations électromagnétiques. Ces relais se démarquent par leur solidité et leur grande fiabilité, pour cette raison, leur entretien est minime. Ils sont réputés pour leur fiabilité dans les environnements de travail les plus délicats.

3.3.2 Relais statiques

La technologie statique analogique, apparue vers 1970, qui utilise des circuits intégrés analogiques et logiques a fait apparaître les relais analogiques qui sont composés grossièrement de trois blocs :

²⁰ <http://thesis.univ-biskra.dz/1873/3/chapitre%2001.pdf>

- ◆ Un bloc d'adaptation et de filtrage, constitué de petits transformateurs, d'impédances et de filtres passe-bas destinés à éliminer les composantes transitoires rapides ;
- ◆ Un bloc de traitement et de détection, composé d'un circuit analogique adapté, transformant la grandeur surveillée en une tension ou un courant continu proportionnel, et d'une bascule servant à détecter le passage d'un seuil ;
- ◆ Un bloc de sortie, comprenant un temporisateur, par exemple un circuit RC, et un relais de sortie électromécanique. Les principaux avantages des relais analogiques sur les relais électromagnétiques sont leur sensibilité, leur précision, leur faible puissance de fonctionnement (quelques VA), et permettent de réduire les dimensions des transformateurs de courant. Par contre, ils nécessitent souvent une alimentation auxiliaire et leurs circuits analogiques sont affectés par les interférences électromagnétiques.

3.3.3 Relais numériques

La technologie numérique a fait son apparition au début des années 1980. Avec le développement des microprocesseurs et des mémoires, les puces numériques ont été intégrées aux équipements de protection. Les protections numériques, sont basées sur le principe de la transformation de variables électriques du réseau, fournies par des transformateurs de mesure, en signaux numériques de faible tension. Ces dispositifs nécessitent une source auxiliaire, offrent un excellent niveau de précision et un haut niveau de sensibilité. Ils procurent de nouvelles possibilités, comme l'intégration de plusieurs fonctions pour réaliser une fonction de protection complète dans une même unité, le traitement et le stockage de données et l'enregistrement des perturbations du réseau (perturbographe). Cette génération intègre des possibilités d'autotest et d'autocontrôle qui augmente leur continuité de fonctionnement tout en réduisant la durée et la fréquence des opérations de maintenance.



Figure 3.3 : Exemple de relais numérique (SEPAM de Schneider Electric)

3.3.4 Principes de fonctionnement des relais de protection

Tous les paramètres d'un réseau électrique peuvent être utilisés pour sa surveillance et la détection de défauts. Il s'agit le plus souvent de mesure du courant et de la tension du réseau. En général, quand un défaut de court-circuit se produit le courant augmente et la tension baisse. A travers la variation de ces deux grandeurs, d'autres paramètres varient également et on obtient des mesures de paramètres plus complexes :

- ◆ Déphasage par comparaison des phases ;
- ◆ Puissance apparente en effectuant le produit du courant par la tension ;
- ◆ Puissances active et réactive à partir de la puissance apparente et du déphasage ;
- ◆ Impédance en effectuant le quotient de la tension par le courant ;
- ◆ Composante homopolaire par addition et composante inverse par des circuits déphaseurs.

Le principe de fonctionnement d'un relais est basé sur la détection de ces variations à l'intérieur de sa zone de protection. Les relais sont caractérisés par leurs grandeurs d'entrée auxquelles ils répondent.

3.4 Fusible

Définition

Les fusibles permettent d'interrompre automatiquement un circuit parcouru par une surintensité pendant un intervalle de temps donné. L'interruption du courant est obtenue par la fusion d'un conducteur métallique calibré. Ils sont surtout efficaces pour la protection contre les courts circuits, vis-à-vis desquels ils agissent, le plus souvent, en limiteurs de la valeur crête du courant de défaut. Le fusible est ainsi un excellent dispositif pour l'élimination des défauts mais il présente un certain nombre d'inconvénients qui limitent son utilisation : Ils exigent malheureusement d'être remplacés après chaque fonctionnement ; En régime triphasé, ils n'éliminent que les phases parcourues par un courant de défaut, ce qui peut présenter un danger pour le matériel et le personnel ; Leur calibre doit être bien adapté pour éviter un fonctionnement intempestif en cas de surcharge momentanée. Les fusibles peuvent être associés à des interrupteurs ou à des contacteurs avec lesquels ils constituent des combinés capables d'assurer automatiquement la coupure des phases saines lors du fonctionnement de l'un d'eux de façon à éviter la marche monophasée dangereuse²¹.

3.5 Arduino et protection circuit

3.5.1 Précautions d'utilisation

Il est facile d'utiliser les nombreux exemples disponibles sur internet sans véritablement connaître toutes les notions relatives à ces projets. Cependant il existe un certain nombre de notions de base qui vous permettront de mieux comprendre vos réalisations dans le but de

²¹ <http://thesis.univ-biskra.dz/1873/3/chapitre%2001.pdf>

les adapter plus précisément à vos besoins. Aussi, avant de se lancer dans vos premiers circuits, il est recommandé de prendre connaissance des connaissances qui suivent²².

3.5.2 Savoir reconnaître les composants électroniques

Chaque composant possède une forme particulière ainsi qu'un code couleur ou un numéro d'identification qui vous permet de connaître sa fonction et ses valeurs électriques. Chaque composant possède une documentation technique communément appelée Datasheet que l'on peut facilement trouver sur internet. Cette documentation présente les caractéristiques précises du composant que vous souhaitez utiliser. Attention : certains composants se ressemblent mais n'ont pas la même fonction. Il faut toujours bien regarder les symboles qu'ils comportent et se référer à leur documentation technique.

3.5.3 Représentations graphiques

Il vous sera utile de savoir déchiffrer les schémas techniques ou **schematics**, qui sont utilisés pour décrire les circuits électroniques. Ces schémas utilisent des symboles standardisés pour chacun des composants. Il existe deux standards en concurrence : le standard Américain et le standard Européen (voir « Les bases de l'électronique »).

3.5.4 Protection du port USB

Tout d'abord, il faut savoir que le port USB de votre ordinateur délivre une quantité limitée de courant électrique. En général un port USB peut fournir au maximum 500 mA en 5 V. Si vous souhaitez réaliser un montage qui nécessite plus de courant, il s'agira de prévoir une **alimentation externe suffisamment puissante pour répondre à votre besoin.**

3.5.5 Protection de la carte Arduino

Sachez que chacune des entrées/sorties de la carte ne peut pas délivrer plus de 20 mA. Cette quantité de courant est relativement faible mais permet, par exemple, de contrôler une diode électroluminescente DEL ou LED en anglais ainsi que des actionneurs de faible puissance tel qu'un piézoélectrique ou encore un petit servomoteur (voir « Circuit de commande »).

²² <https://fr.flossmanuals.net/arduino/precaution-dusage/>

3.5.6 Protection des entrées numériques

Comme leur configuration dépend du programme résident dans la carte Arduino, ce que l'on considère dans un montage comme des entrées numériques peuvent accidentellement être configurées en sortie (avec la commande `pinMode`) et réglées à 0 V (LOW). Dans ce cas, si cette « entrée » reçoit du 5 V, il se produira un court-circuit qui risque d'endommager partiellement ou intégralement la carte.

Pour éviter ce genre de mésaventure, lorsqu'on câble une entrée numérique, il est prudent d'y ajouter une résistance de 100 Ω qui limite le courant en cas de fausse manœuvre

3.5.7 Protection des composants

Chaque composant possède ses propres conventions d'utilisation. Par exemple, il existe des composants qui possèdent un sens de branchement à respecter. L'on dit que ces composants sont polarisés. C'est le cas des LED, de certains condensateurs, des diodes, etc.

La plupart des composants ne peuvent pas fonctionner seuls, par exemple une LED a besoin d'une résistance appropriée pour ne pas s'user ou « brûler ». Cette résistance permet de limiter le courant qui traverse la LED. Le courant supprimé est alors dissipé en chaleur par la résistance (voir « Loi d'Ohm » dans le chapitre « Bases de l'électronique »).

3.5.8 Circuits de commande et de puissance

Lorsqu'un système comporte des moteurs ou des actionneurs demandant un certain niveau de puissance que le microcontrôleur ne peut pas fournir, il faut avoir recours à deux circuits distincts interconnectés, appelés « circuit de commande » et « circuit de puissance ».

Suivant la puissance présente, les interconnexions entre les deux circuits nécessiteront des précautions particulières.

3.5.9 Circuit de commande

Comme son nom l'indique, c'est dans ce circuit que sont rassemblés tous les éléments de contrôle comme les boutons, les interfaces et le microcontrôleur. Il est alimenté en basse tension : moins de 50 V, souvent 12 V, ou avec la carte Arduino 5 V. L'on pourrait l'assimiler

au système nerveux d'un organisme : c'est ici que se prennent les décisions mais peu d'énergie y circule.

La manière la plus simple de relayer les commandes émergeant de ce circuit pour les transmettre au circuit de puissance est d'utiliser des transistors ou encore des relais.

Lorsque les tensions d'alimentation des deux circuits sont plus importantes ou si l'on veut protéger la commande de retours accidentels de courant provenant de la puissance, des optocoupleurs (plutôt que des transistors) assurent une isolation galvanique : l'information est transmise sous forme de lumière. Ainsi, les deux circuits sont complètement isolés électriquement.

3.5.10 Circuit de puissance

Ce circuit alimente les composants nécessitant beaucoup d'énergie (habituellement les moteurs et autres actionneurs). Sa tension d'alimentation dépend des composants en question.

En fonction de cette tension et des conditions d'utilisation, les précautions à prendre sont variables :

- dans tous les cas, une protection contre les courtes-circuits est conseillée : fusible ou disjoncteur pour éviter de détruire les composants ;
- au-dessus de 50 V, la tension peut menacer directement les humains : protéger les pièces nues sous tension ;
- le 230 V nécessitent un interrupteur différentiel qui protège les humains des contacts accidentels (en général tout local est doté de ce dispositif). Ne pas manipuler de 230 V sans connaissances appropriées.

Il est important de noter que ces deux circuits sont montés distinctement et sont isolés l'un de l'autre. Toutefois, il est primordial de lier leur masse (« Gnd ») afin qu'ils partagent le même potentiel de référence.

3.5.11 Court-circuit

Votre carte ne doit pas être posée sur un support conducteur car elle possède sur son verso des zones nues qui ne doivent pas être mises en contact afin de ne pas court-circuiter les composants entre eux. Il faut aussi ne jamais connecter directement le port noté « Gnd » (pôle négatif) avec la broche 5 V (pôle positif).

3.6 Conclusion

Dans ce chapitre nous avons brièvement présenté le circuit de protection des capteurs, comment protéger les captures par l'Arduino, les composants électriques et ses principes de fonctionnement. Nous présentons dans le chapitre suivant l'implémentation et la validation du système.

Chapitre 4 : Réalisation et Résultats

4.1 Introduction :

Dans le but de réaliser notre interface graphique, ce chapitre est organisé en 2 parties :
L'interface graphique des résultats donnés, l'interface des capteurs électriques dans le temps réel.

4.2 Logo d'interface :

En a utilisé le logo de Saad Dahleb Blida comme un logo d'interface en format (.ICO)



Figure4.1:saad Dahleb logo dans la fenêtré de l'interface

On l'import la photo dans notre IDE pycharm dans cette manière :

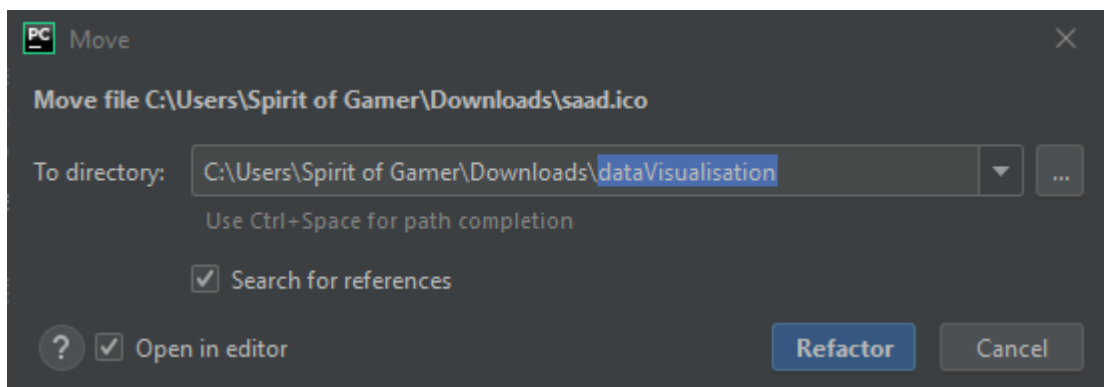


Figure4.2: le chemin d'importation du logo sur le code

Utilisent la syntaxe suivante :

```
window.iconbitmap("icon.ico")
```

```
window.iconbitmap("icon.ico")
```

Le logo doit être affiché dans cette forme



Figure4.3:Exemple de résultat de logo

4.3 Le Botton « email me » :

C'est un Botton qui doit ouvrir le navigateur web et guider vers ce lien « mon email »

(<https://mail.google.com/mail/u/0/?tab=mm#sent/FMfcgzGkbDdIVCwGZFmHwmlDsQrhJQKr?compose=new>)

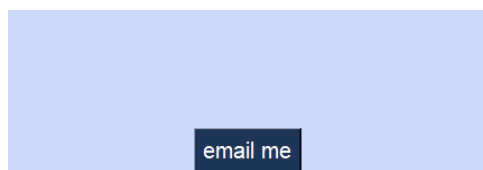


Figure4.4Exemple de Botton sur l'interface

4.4 Les Botton principales

4.4.1 Le 1^{er} Botton :

C'est un Botton qui devait afficher un plot hors hasard, le code utilise les variables qui été stocké dans la case mémoire sans modification (les valeurs dans la case mémoire à l' instant qui elle être crée)

En utilisant la fonction Nambay du python, nous avons choisi des échantillons aléatoires entre 4900 et 5000 et dessiné le graphe à barres en utilisant le code suivant :

Def graph():

```
World=np.random.normal(5000,20, 45)
```

```
plt.hist(world,4)
```

```
plt.show()
```

NOTE : la fonction numpay nous permet de dessinée plusieurs modèles des graphes (graphes simple, graphe a barres, etc.)

Tracer une Random Courbe

Figure4.5:Exemple de 1er main Botton

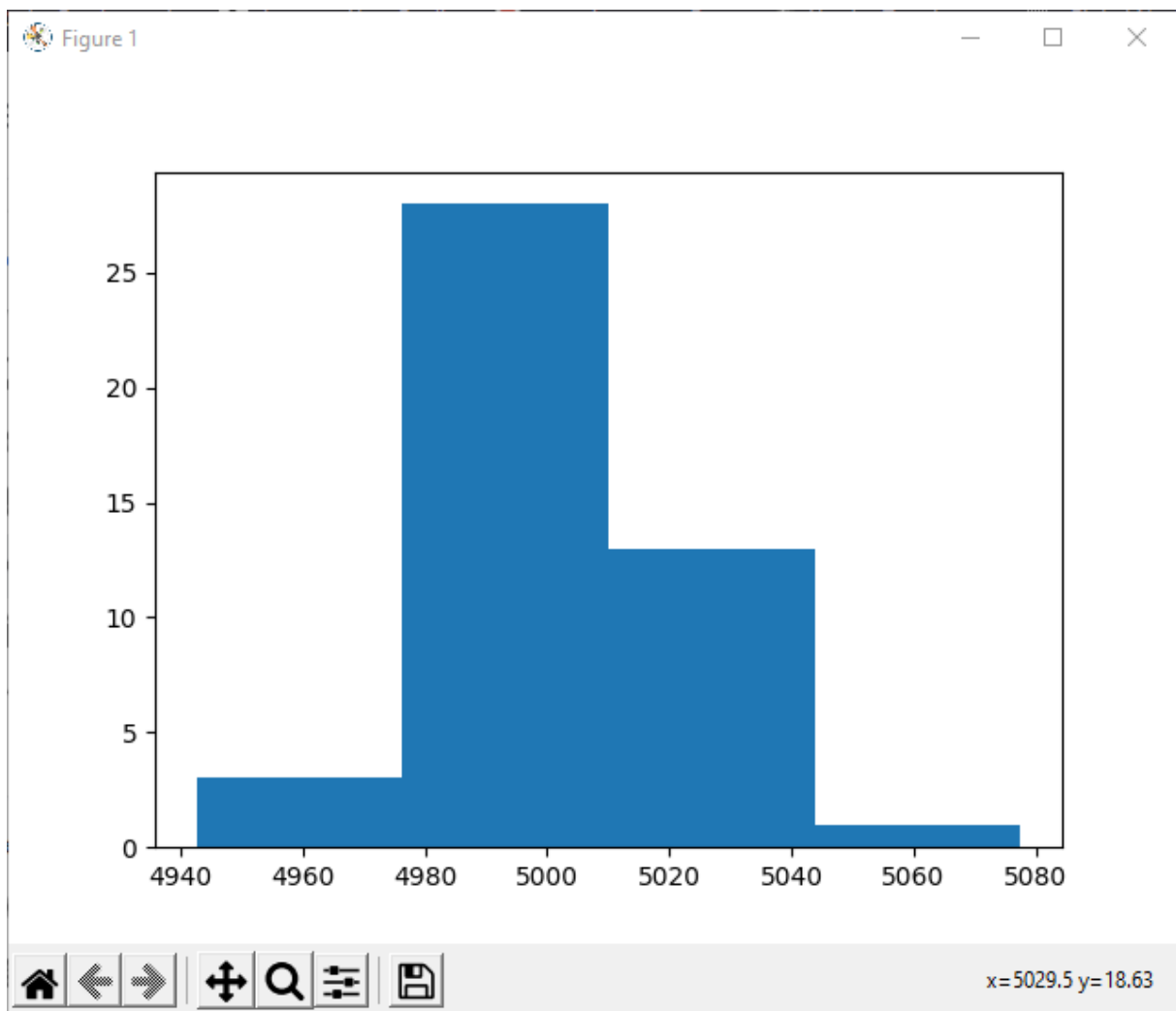


Figure4.6: Résultat de trace de 1er principal Botton

4.3.2 Le 2em Bouton :

Le Graph De Capteur De Temp Donneé de TAB

Figure4.7: Exemple de 1er principal Botton

Dans ce Botton on a pris les résultats du TP « étude de modélisation d'un capteur de température » ils ont calculé les valeurs de température et de voltage par rapporte au temps à l'aide de l'Arduino, et on a obtenu le tableau suivant :

Temps écoulé(en min)	0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30
Température thermomètre étalon (°C)	22	23	24	26	27	29	31	33	3	37	39	41	43	45	47	49
Tension délivrée par le capteur à modéliser(V)	2.39	2.14	2.52	0.61	2.71	2.82	2.94	3.06	3.16	3.27	3.38	3.47	3.56	3.65	3.73	3.8

Tableau4.1 : résultats prélevés de mesures d'un TP de modélisation d'un capteur de température²³

Le plot est de ce forme :

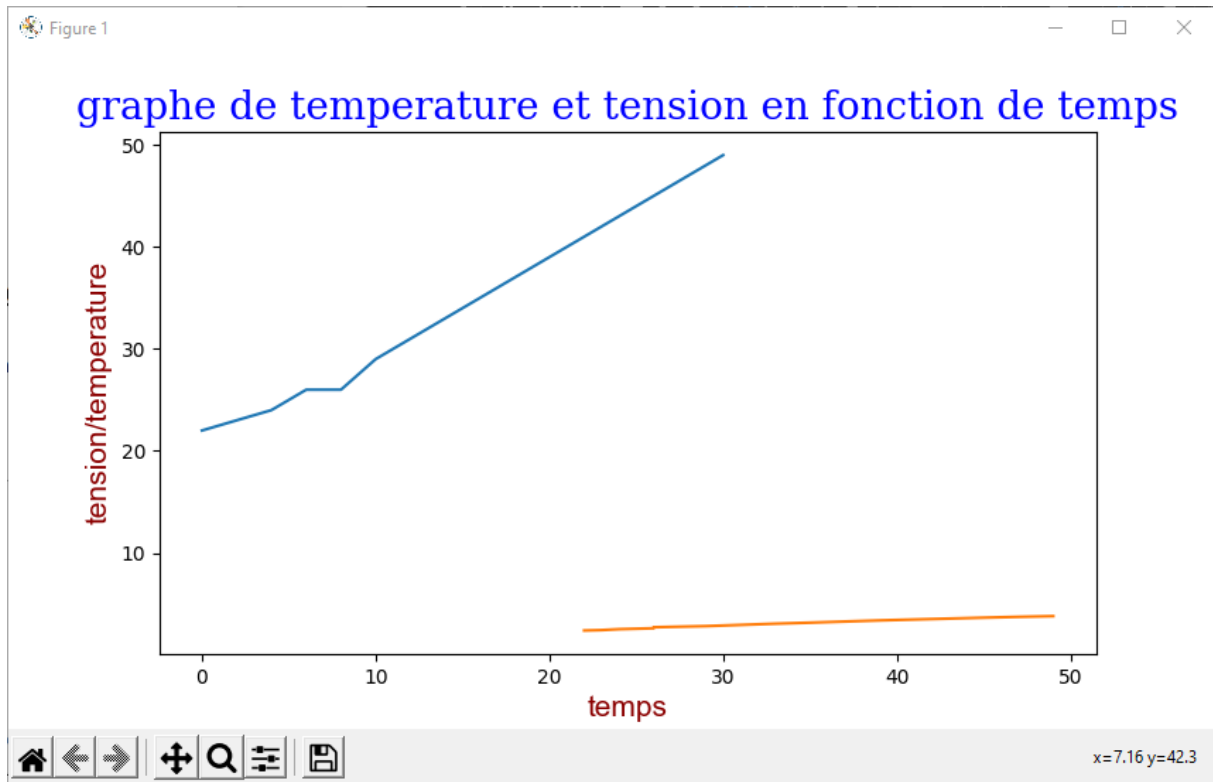


Figure4.7 : Résultat de trace de tableau

4.3.3 Le 3eme Botton :

Le Graph De Capteur De Temp Donneé fot

Figure4.7: Exemple de 3er Botton principal

Dans le troisième Botton on a pris les résultats du TP « étude de mise en œuvre de capteur de température RTD »

La fonction de transfert de la sonde de pt100 et décrit de cette forme :

-une pour les températures au-dessus de 0°C :

$$R_{RTD}(t) = R_0 \cdot (1 + A \cdot t + B \cdot t^2)$$

-une pour les températures au-dessus de 0°C :

$$R_{RTD}(t) = R_0 \cdot (1 + A \cdot t + B \cdot t^2)$$

Avec A, B, C est constante :

$$A = 3.9083 \times 10^{-3} , \quad B = -5.775 \times 10^{-7} , \quad C = -4.183 \times 10^{-12}$$

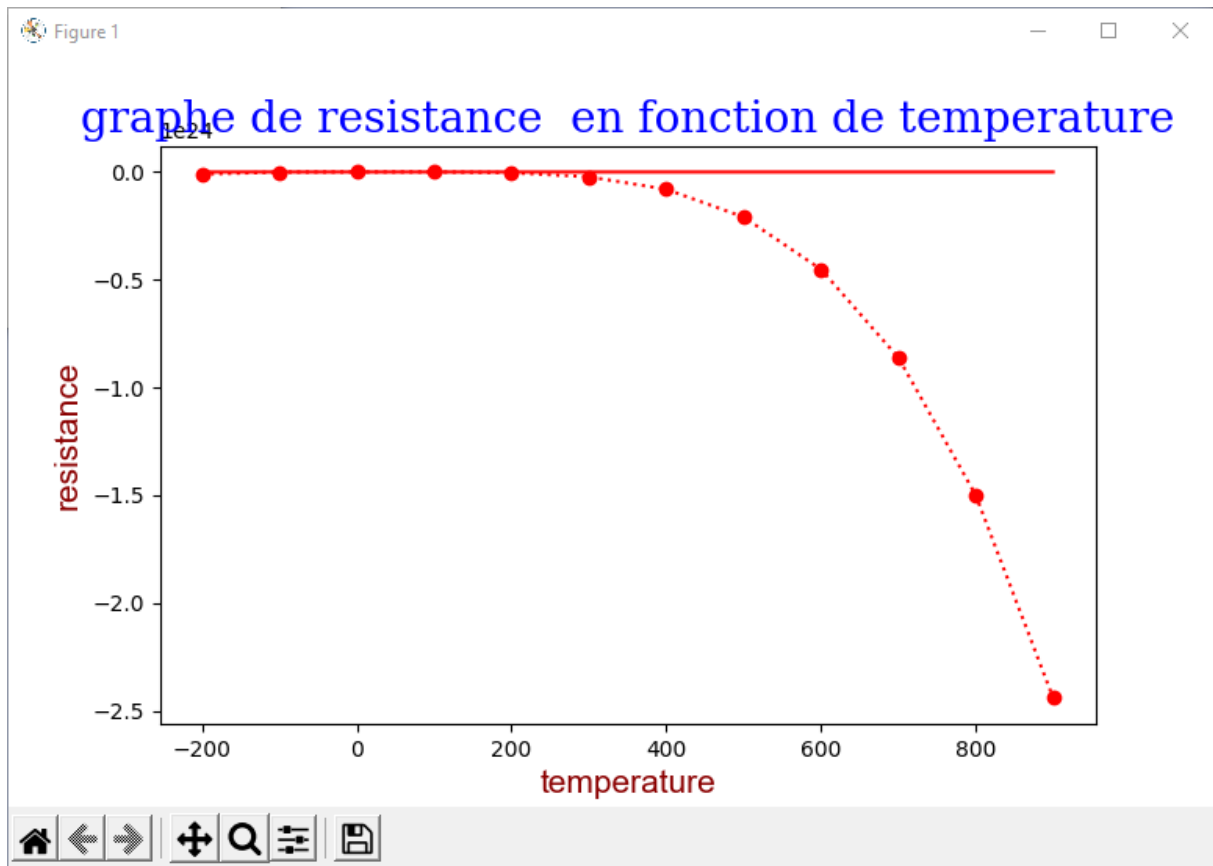


Figure4.8 : Résultat de trace de la fonction

Une fenêtre menu :



Figure4.9 : Exemple de fenêtres en cascade

Pour crée un sub-fenêtre il faut utiliser la fonction « menu_bar »

L'interface finale :

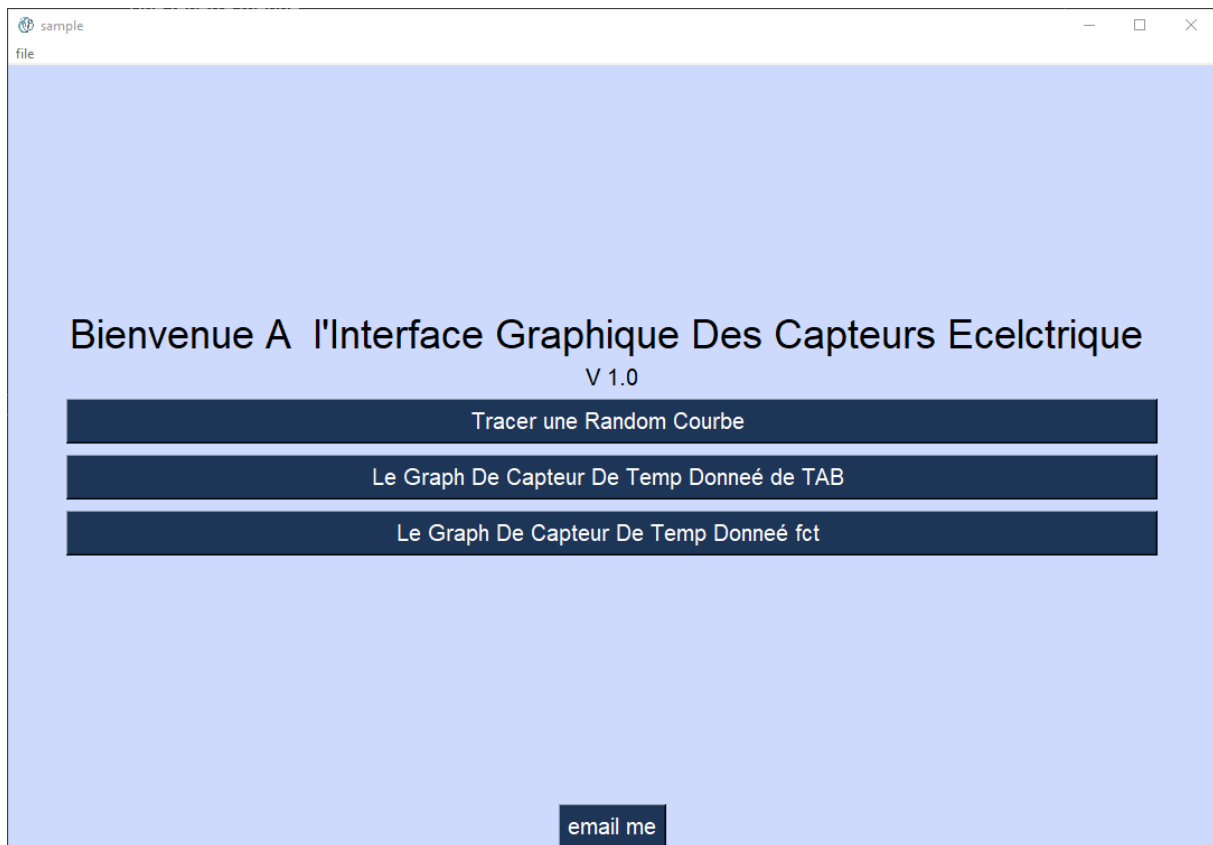


Figure4.10: Fenêtre finale de interface

4.5 Real time data Reading window:

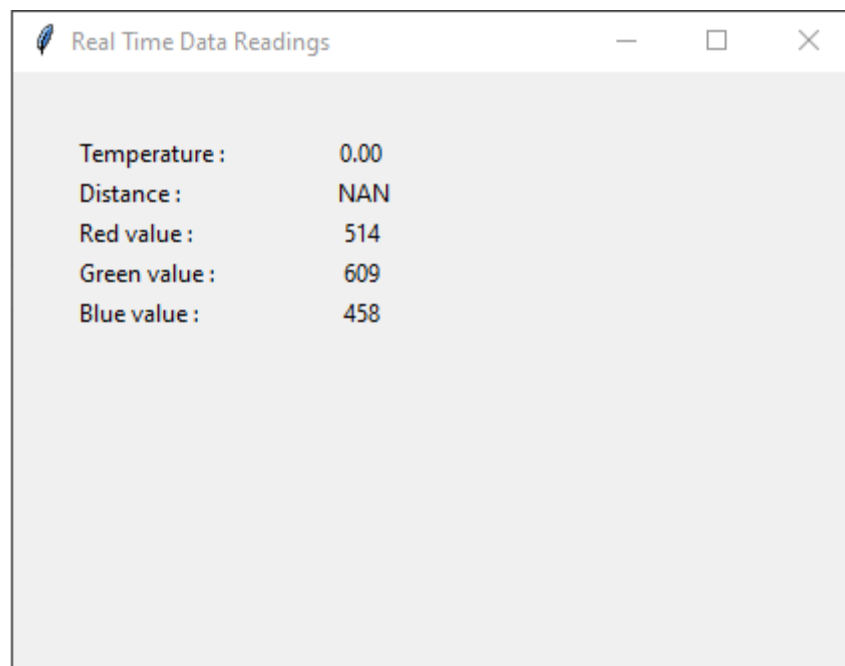


Figure4.11 : Fenêtre de l'interface a résultats en temps réel

4.6 Conclusion

Dans ce chapitre nous avons brièvement présenté les résultats et les figures et ainsi les boutons de notre simulation et notre réalisation et l'organisation d'une interface graphique.

4.7 Conclusion générale

Un capteur est un dispositif transformant l'état d'une grandeur physique observée en une grandeur utilisable, telle qu'une tension électrique, une hauteur de mercure, une intensité ou la déviation d'une aiguille.

Notre travail consiste à un affichage graphique à des capteurs électrique, on fait par 4 étapes.

La première étape consiste à l'étude de généralité sur les capteurs et on choisit les 3 capteurs qui on utilise (capteur de température max 6675, capteur de proximité ultrasons, capteur de couleur tcs3200) et leur caractéristique et fonctionnement.

La deuxième étape consiste à du langage de développement de notre interface qui on choisit le python et On montre comment commander un Arduino depuis un script Python tournant sur un Ordinateur. et ainsi parle à des bibliothèques dans le python.

La troisième étape on parle à chaînes de conditionnement et méthode de protection et comment Arduino protège les capteurs.

La quatrième étape on fait des résultats réels à notre simulation et l'organisation d'une interface graphique.

- [01] http://philippe.berger2.free.fr/automatique/cours/cpt/les_capteurs.htm
- [02] Christophe Pijolat, étude des capteurs microélectroniques : actifs et passifs, Ecole Nationale Supérieure des Mines de Saint-Etienne (2004)
- [03] <http://electroniqueamateur.blogspot.com/2020/09/capteur-de-couleurs-tcs3200-gy-31-et.htm>
- [04] <https://www.microsonic.de/fr/support/capteurs-%C3%A0-ultrasons/principe.htm>
- [05] <https://gaz-tarif-reglemente.fr/lexique-gaz/definition-thermocouple.html>
- [06] <https://arduino103.blogspot.com/2020/02/mod-tc-utiliser-un-thermocouple-type-k.html>
- [07] <http://tiptopboards.com/339-thermocouple-de-type-k-avec-module-d-amplificateur-num%C3%A9rique-max6675.html>
- [08] <https://www.positron-libre.com/electronique/arduino/arduino.php>
- [09] <https://cours-informatique-gratuit.fr/dictionnaire/interface-graphique/>
- [10] https://fr.wikibooks.org/wiki/Programmation_Python/Biblioth%C3%A8ques_pour_Python
- [11] <http://infohost.nmt.edu/tcc/help/pubs/tkinter/web/index.html>
- [12] <https://www.fil.univ-lille1.fr/~marvie/python/chapitre6.html>
- [13] <https://www.scipy.org/about.html>
- [14] https://books.google.fr/books?id=4wMn27KSnTsC&printsec=frontcover&dq=Matplotlib&hl=fr&sa=X&ei=H87fUpvZCpSw7Aa9loDQBw&redir_esc=y#v=onepage&q=Matplotlib&f=false
- [15] <https://matplotlib.org/stable/users/license.html>
- [16] <https://wiki.python.org/moin/PySerial>
- [17] <https://www.f-legrand.fr/scidoc/docimg/sciphys/arduino/python/python.html>
- [18] <http://thesis.univ-biskra.dz/1873/3/chapitre%2001.pdf>
- [19] <http://thesis.univ-biskra.dz/1873/3/chapitre%2001.pdf>
- [20] <http://thesis.univ-biskra.dz/1873/3/chapitre%2001.pdf>
- [21] <https://fr.flossmanuals.net/arduino/precaution-dusage/>
- [22] <https://fr.flossmanuals.net/arduino/precaution-dusage/>