

Mini plaque de refusion

pour l'assemblage ou la réparation de petits circuits CMS

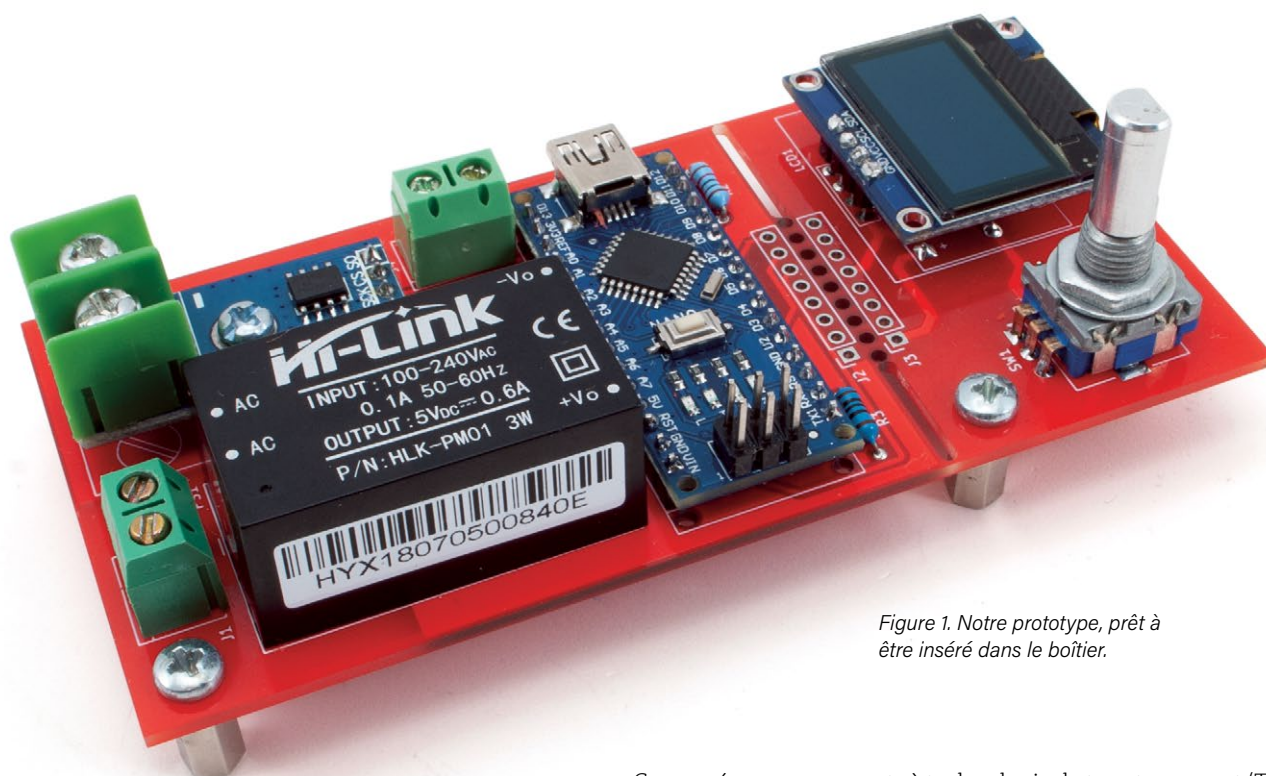


Figure 1. Notre prototype, prêt à être inséré dans le boîtier.

Massimo Divito (Italie)

Lors du passage aux composants miniaturisés CMS, beaucoup ont dû faire face à une série de difficultés d'assemblage. Outre le placement des composants, qui n'est pas si facile, il faut respecter des températures et des temps spécifiques pour le cycle de soudage. Les stations professionnelles de soudage par refusion sont coûteuses. Dans cet article, nous présentons une solution efficace, mais d'un prix raisonnable, pour l'assemblage et la réparation de nos cartes.

Comparé aux composants à technologie de trou traversant (THT), travailler avec des composants à technologie de montage en surface (CMS) peut être assez exigeant. Il faut une vue perçante, une main très sûre, un fer à souder avec une panne appropriée et, surtout, beaucoup de patience.

Lorsqu'un circuit comporte une majorité de composants montés en surface et, surtout, lorsqu'il s'agit de puces à plusieurs broches, la situation devient plus compliquée, d'autant plus si les éléments sont nombreux et rapprochés ; l'utilisation d'un fer à souder peut alors s'avérer impossible. C'est pourquoi nous avons imaginé un dispositif apte à répondre à vos besoins dans de telles situations : quelque chose de simple, peu coûteux et facile de mise en œuvre. La technique de soudage utilisée dans le milieu professionnel pour les CMS s'appelle le soudage par refusion et utilise habituellement des fours. Nous avons plutôt pensé à une plaque chauffante, adaptée aux petits circuits, qui prend peu de place dans notre laboratoire et peut servir si nécessaire de préchauffage pour les opérations de soudage et de dessoudage avec la station à air chaud.

Le soudage par refusion est un procédé dans lequel une pâte à braser, composée d'étain en poudre et de flux liquide, est utilisée à la place de l'alliage étain/plomb normal (ou du type sans plomb pour les applications conformes à la directive ROHS) pour souder les composants sur la carte de circuit imprimé.

La pâte à braser est déposée sur les plages du circuit imprimé à l'aide d'une seringue ou d'un pochoir spécial, puis tous les composants électroniques sont d'abord placés avant de faire subir à l'ensemble un traitement thermique contrôlé, qui fait fondre l'alliage et soude les composants de manière permanente.

Comment ça marche

Notre conception consiste en une plaque d'aluminium chauffée par des résistances électriques qui fonctionnent sous 230 V ; le circuit imprimé à souder est placé sur cette plaque. Les résistances sont actionnées par un relais à semi-conducteurs, commandé à son tour par un circuit basé sur un Arduino Nano, comme on peut le voir sur le circuit entièrement assemblé de la **figure 1**. Le circuit est relié à un thermocouple de type K avec un œillet de maintien, vissé à la plaque, qui mesure en permanence sa température.

La commande se fait par un encodeur rotatif qui permet de sélectionner le mode de fonctionnement parmi deux profils différents : l'un pour les alliages d'étain avec plomb et l'autre pour les alliages sans plomb. Le troisième mode permet de régler la plaque à une température fixe ; cette fonction est utile, par exemple, pour préchauffer un circuit sur lequel on va travailler avec un styilet à air chaud. Nous avons ensuite ajouté un petit écran OLED de 0,96 pouces sur lequel nous sélectionnons le profil que nous voulons lancer et qui, pendant les phases de refusion, nous indique la température réglée et la température atteinte par la plaque, ainsi que le temps restant jusqu'à la fin du processus. Comme déjà mentionné, pour effectuer correctement le processus de soudage, il est nécessaire de suivre une courbe de température précise afin de permettre à la pâte à braser de fondre correctement et, en même temps, d'éviter que les composants ne soient endommagés par une chaleur trop élevée ; en règle générale, chaque fabricant spécifie, pour ses propres intégrales, la courbe optimale. L'ensemble du processus est divisé en quatre étapes :

- préchauffage : les agrégats de la pâte à braser commencent à s'évaporer.
- trempage : les agrégats s'évaporent complètement, le flux fond et désoxyde les pastilles.
- refusion : la partie métallique de la soudure fond et crée une connexion mécanique et électrique permanente entre les composants et le circuit imprimé. Le flux s'évapore.
- refroidissement : l'étain se refroidit en se solidifiant.

Pour ce faire, il est nécessaire de connaître, à chaque instant du processus de refusion, la température que doit avoir la plaque. Étant donné que la température de la plaque (**figure 2**) est lue environ toutes les 250 ms, pour un seul profil d'une durée de 380 s, on doit connaître 1520 valeurs de température (380 000 / 250). Le stockage et l'accès à un si grand nombre de valeurs deviennent assez compliqués au niveau du micrologiciel, et tout changement ultérieur serait fastidieux. De plus, si comme dans notre cas, il y a deux profils (mais nous pourrions en ajouter d'autres si nous le souhaitions), le nombre de valeurs double. C'est pourquoi nous sommes arrivés à une solution alternative, grâce à laquelle nous

pouvons définir seulement certaines des températures et, à partir d'elles, calculer les autres.

Si l'on regarde les graphiques des profils à la **figure 3**, on remarque que les variations de température, tant à la hausse qu'à la baisse, sont linéaires. Nous pouvons alors découper chaque profil en plusieurs segments, dont il suffit de connaître les extrêmes. Chaque segment est défini par deux paires de valeurs qui indiquent son début et sa fin, et chaque paire est composée des valeurs de secondes et de température, respectivement ; ces valeurs sont facilement stockées dans un tableau à deux dimensions composé de six éléments : les six pics qui décrivent précisément le profil.

Ainsi, si l'on veut connaître la valeur de la température à fixer sur la plaque à un instant donné, il suffit d'appliquer la méthode d'interpolation linéaire, qui permet d'extrapoler une valeur inconnue entre

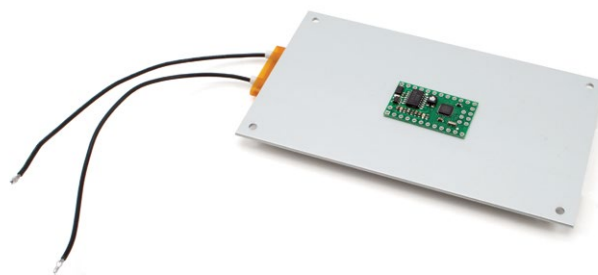


Figure 2. Pour le soudage, on peut placer une carte sur la plaque chauffante, après avoir enduit les pastilles de pâte à braser.

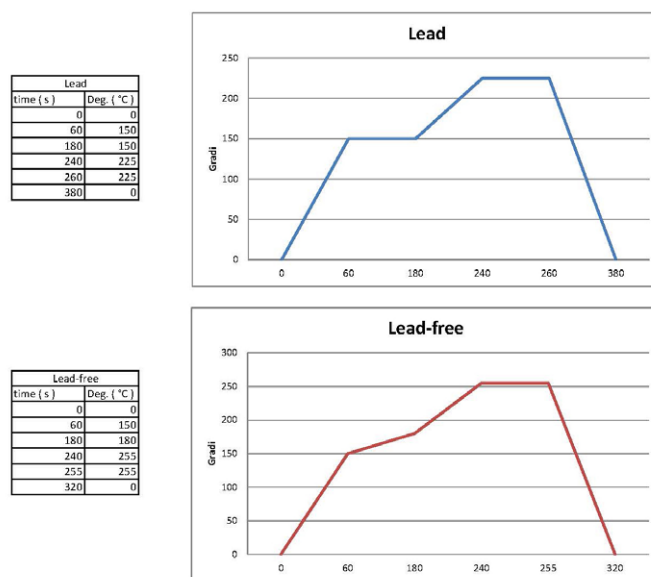


Figure 3. Courbes de température pour le soudage avec des alliages avec et sans plomb.

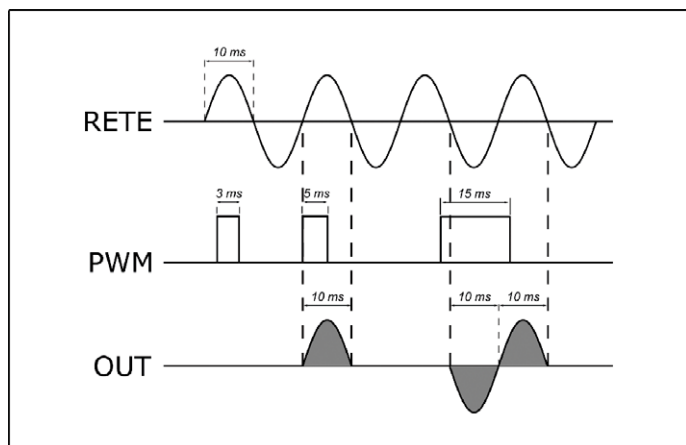


Figure 4. Principe de fonctionnement de la méthode de déclenchement au passage par zéro pilotée par MLI (RETE = tension secteur).

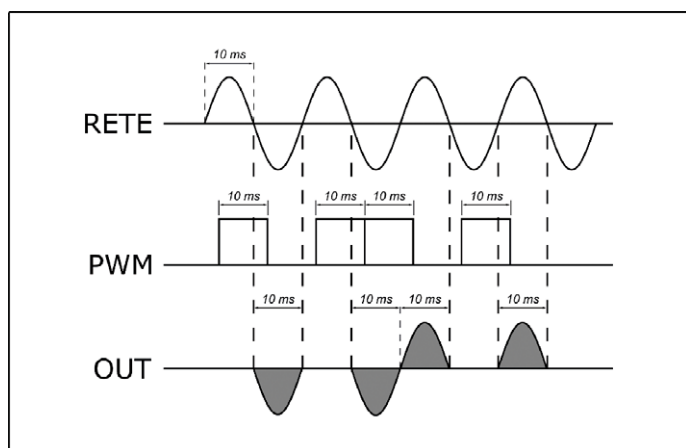


Figure 5. Comportement de la méthode de déclenchement au passage par zéro pilotée par PWM à 1 Hz (RETE = tension secteur).

deux valeurs connues. Connaissant la valeur de la température que la plaque devrait avoir et sa température réelle, nous pouvons l'activer ou la désactiver en conséquence. Pour que la température suive plus précisément ce qui est indiqué par le profil, nous utiliserons la technique de commande PID. Ce type de commande ajuste simplement une quantité en la surveillant constamment et en agissant sur certaines variables, en essayant de la maintenir proche de la valeur de consigne. Il est clair que l'opération n'est pas si triviale, en fait, elle suit des formules mathématiques précises. Heureusement, il existe des bibliothèques spéciales pour Arduino qui facilitent grandement cette opération.

Comme déjà mentionné, pour allumer la plaque, nous utilisons un relais à semi-conducteurs doté d'une fonction de détection du passage par zéro, qui fait que le changement d'état de sa sortie - c'est-à-dire la transition entre « on » et « off » et vice versa - ne se produit que lorsque l'onde de tension du réseau, dont on se souvient qu'elle est de 50 Hz, passe par 0 V. Cela signifie qu'entre le moment où nous excitons ou désexcitons le relais et le moment où il allume ou éteint la charge, il peut s'écouler quelques millisecondes.

Il est clair que cette particularité ne permet pas d'avoir un contrôle exact de la sortie. En effet, piloter le relais avec une impulsion d'une durée inférieure à celle d'une demi-onde, que nous aurons peut-être envoyée une milliseconde après le passage à zéro, aura

pour conséquence de ne pas allumer la charge. En revanche, si nous envoyons l'impulsion exactement au point de transition zéro, la charge s'allumera, mais elle ne s'éteindra qu'au prochain passage à zéro. Si l'on prend l'exemple de la **figure 4**, après une MLI qui a piloté le relais pendant un total de 23 ms, la sortie est active pendant un total de 30 ms. Une solution consisterait à utiliser un circuit détecteur de passage à zéro qui, comme son nom l'indique, peut détecter tous les moments où la tension du réseau passe par zéro, afin de synchroniser correctement la sortie du MLI. Dans notre cas, afin de ne pas compliquer davantage le circuit, nous avons résolu ce problème en utilisant une fréquence MLI très basse, plus précisément 1 Hz. De cette manière, nous aurons une période totale d'une seconde, ou mieux, de 1000 millisecondes, divisée en 100 étapes de 10 millisecondes chacune. En pratique, pour chaque variation de 1 % du rapport cyclique, nous aurons une variation de la longueur de l'impulsion de 10 ms, ce qui correspond exactement à la durée d'une demi-onde de la fréquence du réseau. De cette manière, la valeur du rapport cyclique que nous fixons sera correctement transmise à la sortie du relais, puisque nous sommes certains que le même retard appliqué à l'activation de la charge sera également appliqué à sa désactivation, la seule différence étant un retard possible allant de 1 à 9 millisecondes, qui décalera toute l'onde vers l'avant (**figure 5**). Cette conception ne prévoit que la régulation du chauffage, tandis que le refroidissement s'effectue naturellement par dissipation de la chaleur. La courbe réelle ne sera donc pas identique à la courbe théorique. Beaucoup dépend de l'environnement dans lequel nous opérons, mais nous pouvons vous assurer qu'après plusieurs essais, nous n'avons jamais eu de problèmes pour terminer le processus de refus avec d'excellents résultats.

Schéma de câblage

L'utilisation d'Arduino (U1), simplifie le circuit (**figure 6**) ; en effet, en plus de celui-ci sur la carte, il n'y a qu'un module avec un MAX6675. On utilise cette puce pour amplifier le signal présent aux bornes du thermocouple de type K, le convertir d'analogique en numérique et le restituer via l'interface SPI. En effet, la lecture de ce signal via les entrées analogiques serait plus compliquée, car les amplitudes impliquées sont de l'ordre du microvolt et, de ce fait, devraient être correctement amplifiées, et le bruit électrique pourrait affecter la mesure. C'est pourquoi nous avons décidé d'utiliser cette puce et, plus précisément, un module pré-assemblé (BRD1) qui l'inclut, puisque le MAX6675 n'est produit qu'en format CMS !

Les signaux provenant du thermocouple sont régulièrement convertis et envoyés à travers les broches SCK, CS et SO connectées aux entrées numériques de U1, qui les traite et décide, en fonction, d'augmenter ou de diminuer la température de la plaque, dont l'élément chauffant est commandé par un relais à semi-conducteurs. Dans notre cas, nous avons utilisé un relais FOTEK SSR-25-DA - logé dans le même boîtier et non visible sur les schémas - qui supporte des charges jusqu'à 25 A. Le pilotage de ce relais se fait via une sortie numérique à travers la résistance R1, comme on le fait habituellement pour allumer une LED. Dans notre cas, nous allons piloter un opto-isolateur intégré, situé dans la section d'entrée.

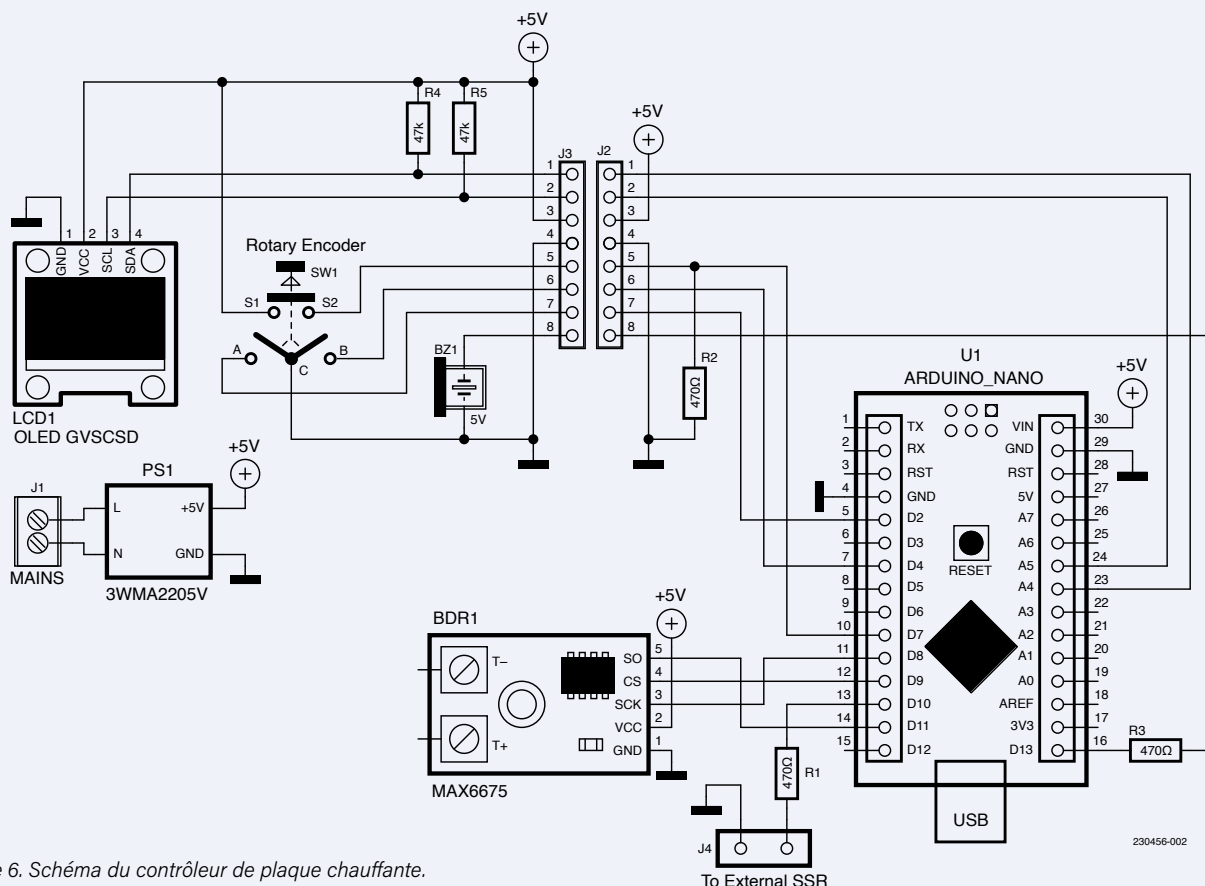
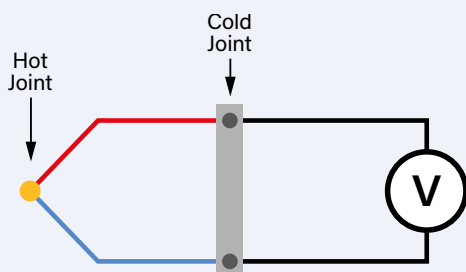


Figure 6. Schéma du contrôleur de plaque chauffante.

Le thermocouple



Les thermocouples sont des transducteurs de température couramment utilisés dans de nombreux domaines : industriel, scientifique, ingénierie, etc. Grâce à leur petite taille et à leur temps de réponse rapide, ils peuvent être utilisés même dans des environnements particulièrement critiques, car ils sont capables de mesurer avec précision une large gamme de températures. Leurs grands avantages sont qu'ils sont très simples, durables et peu coûteux.

Les thermocouples sont conçus pour mesurer les températures élevées des corps physiques, des composants et des substances.

Chaque thermocouple est constitué de deux pièces métalliques différentes reliées en un point appelé « jonction chaude ». Ce point est mis en contact avec l'objet ou la substance dont la température doit être mesurée. L'autre extrémité, celle formée par les deux fils libres, est appelée « jonction froide » et, si elle est connectée à un voltmètre, la différence de température entre les deux jonctions crée une différence de potentiel proportionnelle à la température (effet Seebeck). On trouve des thermocouples sous des formes très différentes les uns des autres, mais leur principe de fonctionnement reste fondamentalement le même. La soudure froide provient de

la différence entre le métal des fils du thermocouple et celui des fils de connexion.

Il existe différents types de thermocouples, mais le type K est certainement le plus utilisé dans l'industrie. Il est constitué d'une combinaison de fils à base de nickel et les plages de mesure vont de -200 à +1260 degrés Celsius. De plus, la résistance à la corrosion et à l'oxydation du nickel lui permet d'être utilisé dans une large gamme d'applications.

Le choix d'utiliser un tel composant est dicté par le fait que la charge qui lui est appliquée est pilotée en MLI, c'est-à-dire par une séquence d'impulsions qui allume et éteint la plaque plusieurs fois par seconde. Un relais électromécanique ordinaire, outre qu'il n'aurait pas la vitesse de commutation requise, ne durerait pas longtemps et ses contacts seraient irrémédiablement endommagés par l'arc électrique.

Sur les broches SDA et SCL de l'Arduino, nous avons connecté LCD1, un petit écran d'un pouce basé sur la puce SSD1306, en ajoutant les résistances de rappel haut R4 et R5. En outre, nous avons le codeur rotatif en quadrature SW1, avec ses broches A et B connectées aux entrées numériques de U1 : on voit sur le schéma que nous n'avons pas ajouté de résistances de rappel haut, mais avons activé via le logiciel celles situées à l'intérieur du microcontrôleur ; la broche C est connectée à GND à la place. Le bouton-poussoir intégré à l'arbre du codeur, qui permet les différentes sélections, est connecté d'un côté à Vcc et de l'autre à la broche 10 de U1, avec la résistance de rappel bas R2.

Enfin, il y a le buzzer, BZ1, connecté à la broche 9 de U1 via la résistance R3. L'ensemble du circuit est alimenté par le module PS1, un

convertisseur CA/CC de Hi Link qui se monte directement sur le circuit imprimé. Ce module a une entrée de 230 V CA et une sortie de 5 V, 600 mA. Cette solution évite l'emploi d'une alimentation externe, en facilitant le logement de l'ensemble du circuit dans un seul boîtier.

Firmware

Le croquis Arduino, téléchargeable à l'adresse [1], est assez volumineux, aussi nous ne commenterons ici que certaines parties, celles qui sont directement pertinentes à comprendre et qui vous permettent de personnaliser le projet à votre guise. Pour commencer, voici les quatre bibliothèques que nous utilisons :

- Affichage (U8g2 par Oliver)
- MAX6675 (MAX6675_Thermocouple par Yurii Salimo)
- PID (PID par Brett Beauregard)
- PWM (MLI) (<https://github.com/maxint-rd/FastPwmPin>)

Ensuite, les broches connectées au MAX6675 et à l'encodeur rotatif sont définies, ainsi que deux symboles graphiques personnalisés qui composeront l'icône du thermomètre sur l'écran. Notez que dans les déclarations initiales, le tableau `profiles`, qui, comme nous l'avons mentionné, définit les deux profils de soudage :

```
int profiles [2][6][2] = {
  {{0, 0}, {60, 150}, {180, 150}, {240, 225},
   {260, 225}, {380, 0}}, // Lead profile
  {{0, 0}, {60, 150}, {180, 180}, {240, 255},
   {255, 255}, {320, 0}} // Lead-free profile
};
```

Ici, vous devrez agir si vous voulez changer ou ajouter un profil, en modifiant les valeurs du tableau de manière appropriée et en vous rappelant que, pour chaque paire de valeurs, la première se réfère au temps en millisecondes et la seconde aux degrés ; les tableaux de la figure 3 clarifieront le concept. Après ce tableau se trouve un autre bloc de variables qu'il peut être utile de modifier :

```
int tempMinPreheater = 60;
//set the minimum temperature that
// can be set as a preheater
int tempMaxPreheater = 250;
//set the maximum temperature
// that can be set as the preheater
int tempStepPreheater = 5;
//set the value of the setting
// steps for the preheater
int tempAmbient = 50;
//set the maximum ambient temperature,
// if the plate is warmer, the
// remelting process cannot be started
int tempAdjust = 0;
//correction of the
// temperature read by the probe
```

Toutes ces lignes de code sont accompagnées de commentaires, de sorte qu'il est facile de voir que les deux premières indiquent

les valeurs minimales et maximales qui peuvent être définies ; la troisième indique la valeur qui sera ajoutée ou soustraite aux degrés définis pour chaque rotation de l'encodeur ; ces valeurs se réfèrent uniquement à la fonctionnalité de préchauffage. La variable `tempAmbiente` indique la valeur maximale que la plaque peut avoir avant de commencer le processus de refusion et est appelée température ambiante, car la plaque prend généralement la température de l'environnement dans lequel elle se trouve. Cette méthode permet d'éviter de commencer un processus de soudage immédiatement après en avoir terminé un autre. En effet, dans ce cas, cela fausserait la courbe de refusion avec le risque de faire échouer toute la procédure. La variable `tempAdjust` permet de corriger la lecture de la température de la plaque.

Le reste du croquis ne nécessite que peu de commentaires. La lecture des boutons et de l'encodeur se fait sans utiliser d'interruptions : pour lire l'appui long sur P1 et pour mettre à jour périodiquement l'affichage, on utilise plutôt quelques temporisateurs. Dans le code qui gère la fonctionnalité de refusion, nous pouvons voir comment nous utilisons la formule d'interpolation linéaire pour calculer à quelle température régler la plaque :

```
reflowNowTime = (unsigned long)(millis() - reflowStartTime);
// Calculates how many milliseconds
// the reflow cycle started;
for (int i = 0; i < 6; i++) {
  if ((reflowNowTime >=
    (profiles[profile][i][0] * second)) &&
    ( reflowNowTime <
    (profiles[profile][i + 1][0] * second)) {
    xa = profiles[profile][i][0] * second;
    xb = profiles[profile][i + 1][0] * second;
    ya = profiles[profile][i][1];
    yb = profiles[profile][i + 1][1];
    tempTarget = (((reflowNowTime - xa) *
      (yb - ya)) / (xb - xa)) + ya;
    // y0=( ( (x0-xa)*(yb-ya) ) / (xb-xa) ) + ya
    // Linear interpolation formula

    phaseActual = i;
    ruleHeater();
  }
}
```

Dans la boucle for, nous parcourons le tableau des profils pour obtenir les valeurs connues, sur l'axe x, les millisecondes, et sur l'axe y, les degrés centigrades. Nous obtenons ainsi les valeurs `xa`, `ya`, `xb`, `yb`, tandis que `x0` sera égal aux millisecondes écoulées depuis le début de la phase de refusion : avec ces valeurs, nous pouvons calculer `y0`, qui représente la valeur de la température à régler pour notre plaque chauffante.

Réalisation pratique

Un circuit imprimé spécial a été conçu pour le contrôleur, ce qui facilite son montage et le rend compact et soigné. Ce circuit imprimé nous donne la possibilité de détacher la partie encodeur et affichage pour permettre son installation dans une grande variété de configurations ; si vous décidez de scinder le circuit

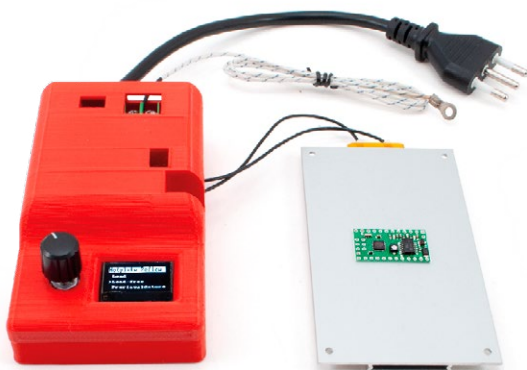


Figure 7. Le système complet, y compris le thermocouple, vissé à l'aide de la cosse à œillet.

imprimé, vous devrez évidemment connecter les deux parties via les connecteurs J2 et J3. L'emplacement optimal du circuit est dans un boîtier en plastique qui permet une isolation totale. Une solution pourrait consister à utiliser une boîte de jonction pour les installations électriques, dont le couvercle serait précisément percé pour l'affichage et l'encodeur rotatif.

Étant donné la présence de la tension réseau, on peut envisager de rassembler dans le boîtier le circuit, le relais statique et la prise de 230 V pour la connexion de la plaque et la sonde thermocouple de type K. Pour permettre une lecture précise de la température, il est important de positionner cette dernière correctement, éventuellement en la fixant solidement à la plaque chauffante avec un support métallique. Une solution pourrait consister à l'insérer dans une cosse à œillet en aluminium, puis à la serrer à l'aide d'une pince sans exagérer la pression pour éviter de l'endommager. De cette façon, la sonde (**figure 7**) peut être fixée à la plaque avec une simple vis, en ajoutant éventuellement une pâte thermoconductrice à la connexion mécanique, comme celle utilisée pour les processeurs de PC, qui favorise la conduction thermique pour améliorer la précision de la lecture.

Une autre solution consisterait à la laisser « libre » et à la coller sur le circuit imprimé que vous soudez, à chaque fois que vous effectuez un processus de refusion, à l'aide d'un morceau de ruban adhésif en kapton.

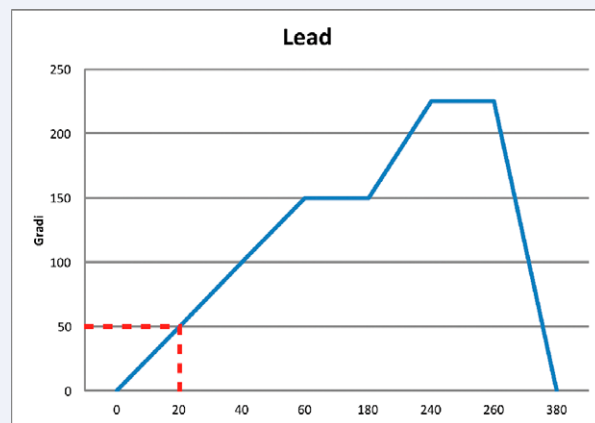


Avertissement : Apportez le plus grand soin à la fabrication et au câblage, compte tenu de la présence de la tension du secteur, qui est très dangereuse : N'oubliez pas de relier le fil de terre à la fiche et à toutes les parties métalliques de l'appareil, y compris la plaque.

Étalonnage et utilisation

Une fois l'Arduino programmé et le projet assemblé, après avoir soigneusement vérifié le câblage (en particulier celui du 230 V), nous pouvons mettre le circuit sous tension et vérifier son bon fonctionnement. Toutefois, avant de l'utiliser, il convient d'effectuer un étalonnage simple mais utile. Dans la pratique, selon l'endroit où l'on place la sonde de type K, il peut y avoir une légère variation

Interpolation linéaire



En mathématiques, l'interpolation désigne une méthode permettant de trouver de nouveaux points dans le plan cartésien à partir d'un ensemble fini de points de données. Elle permet ainsi de déduire, de manière approximative, la tendance d'une courbe en n'en connaissant que quelques points. Dans l'interpolation linéaire, qui est aussi la méthode d'interpolation la plus simple, il suffit de deux points adjacents, identifiés par (x_a, y_a) et (x_b, y_b) , en les joignant pour tracer un segment.

Par conséquent, pour calculer le point (x_0, y_0) , intermédiaire aux points connus, nous utilisons la formule :

$$y_0 = \frac{(x_0 - x_a)(y_b - y_a)}{x_b - x_a} + y_a$$

Donnons un exemple en nous référant à la figure et en substituant à la formule les points du premier segment, à savoir $(0,0)$ et $(60,150)$, en supposant que nous voulons connaître la valeur de y_0 étant donné que $x_0 = 20$

$$y_0 = \frac{(20 - 0)(150 - 0)}{60 - 0} + 0$$

$$y_0 = \frac{20 \times 150}{60}$$

$$y_0 = \frac{3000}{60}$$


$$y_0 = 50$$

des relevés ; cela est dû à la disposition des résistances chauffantes, qui sont généralement centrées, et c'est précisément à ces endroits que l'on obtiendra une lecture plus précise. Au fur et à mesure que nous nous déplaçons, les relevés présentent de petites différences qui augmentent à mesure que la distance par rapport à la source de chaleur s'accroît.

Pour corriger la lecture, nous avons prévu la variable `tempAdjust` qui, par défaut, vaut 0. Si vous disposez d'un moyen de lire la température de la plaque à l'aide d'un thermomètre infrarouge ou autre, vous pourrez vérifier la température lue par notre circuit et celle relevée avec le thermomètre. Vous devrez affecter la valeur résultant de la différence entre les deux à la variable ci-dessus. De cette façon, vous pourrez corriger la lecture à une valeur plus proche de

la réalité. Une fois cette étape franchie, vous serez prêt à souder vos circuits. Nous vous recommandons toutefois de commencer par des circuits de petite taille, par exemple de petits adaptateurs SOIC-DIL – vous devrez vous familiariser avec la quantité de pâte à braser à déposer sur les pistes et le positionnement correct des composants.

Utilisation pratique

En plus d'être utilisé pour le soudage par refusion, ce modèle peut également servir de préchauffage de circuit imprimé pour le soudage et le dessoudage à l'aide d'une station à air chaud. Dans ce dernier cas, le circuit imprimé doit être porté à environ 150 °C, puis les composants à souder sont chauffés avec la station à air chaud, ce qui fait fondre le flux et la soudure. 

VF : Denis Lafourcade — 230456-04

Questions ou commentaires ?

Contactez Elektor (redaction@elektor.fr)



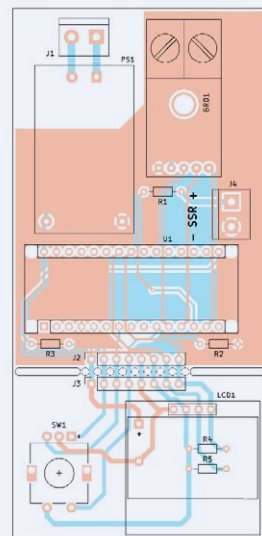
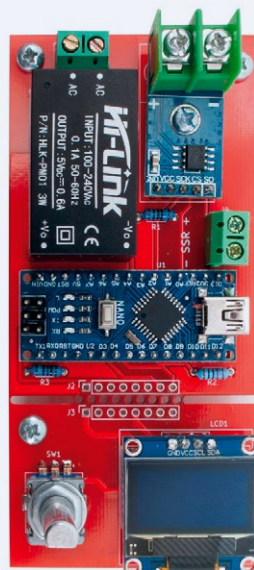
Produits

- > **Four à refusion infrarouge T-962 amélioré (version Elektor) (SKU 20346)**
www.elektor.fr/20346
- > **Station de retouche CMS à air chaud 2-en-1 ZD-8922 (SKU 20141)**
www.elektor.fr/20141

LIEN

[1] Téléchargement du logiciel :
<https://elektormagazine.fr/230456-04>

Liste des composants



Résistances

R1...R3 = 470 Ω

R2 = 10 kΩ

R4...R5 = 47 kΩ

Modules

U1 = Arduino Nano

PS1 = 3WMA2205V (Module PSU)

BRD1 = MODMAX6675 (pour thermocouples de type K)

LCD1 = OLEDGVSCSD (0.96 pouce OLED)

Divers

BZ1 = buzzer 5 V (sans oscillateur)

Barrette femelle, 4 broches

Barrette femelle, 15 broches

Barrette femelle, 5 broches

Bornier à vis, 2 pôles

Encodeur rotatif avec bouton-poussoir

Elektor
Webinars

Exploring the ESP32 P4 Module and the Guest-Edited Elektor Mag

Animé par Anant Raj Gupta et Brian Tristram Williams

Saisissez la chance de gagner
une carte ESP32-C3-DevKitM-1 !



www.elektormagazine.fr/esp32w2



ESPRESSIF



elektor

DÉC
14
2023
16:00 CET