

de l'idée au circuit avec l'ESP32-S3

construire un prototype avec les puces Espressif

Liu Jing Hui, Espressif

De nombreux produits et appareils – en particulier les exemplaires uniques et ceux produits en faible quantité – qui intègrent des puces Espressif sont initialement construits avec des cartes de développement. Les développeurs passent en suite à l'utilisation d'un module ou d'une puce nue s'ils ont besoin de produire un plus grand nombre de produits ou si le produit doit simplement être plus esthétique. Plusieurs facteurs doivent être pris en compte lors de cette migration. Dans cet article, nous abordons les risques les plus courants.

Nous prenons l'ESP32-S3 comme exemple de conception avec les puces Espressif. Cette puce contient une solution intégrée de système sur puce (SoC) à faible consommation, WiFi 2,4 GHz + Bluetooth LE (5). Elle dispose de deux coeurs rapides, d'une mémoire vive extensible en externe et d'une multitude d'interfaces qui prennent en charge de nombreuses applications. Il est important de noter que le SoC dispose également d'un matériel d'étalonnage intégré permettant de compenser les légers défauts de la configuration radio. Il est donc relativement facile de faire fonctionner votre circuit et il n'est pas nécessaire de recourir à un équipement de test spécialisé. L'ESP32-S3 est un choix idéal pour un large éventail d'applications d'intelligence artificielle (IA) et d'IA des objets (AIoT). L'ESP32-S3 est disponible sous forme de puce nue ou de module (comme l'ESP32-S3-WROOM-1) qui comprend l'ESP32-S3 ainsi que le quartz nécessaire, la mémoire flash, la commutation haute fréquence et, en option, une antenne sur circuit imprimé ou une connexion pour une antenne externe. Dans cet article, nous aborderons l'implémentation de la puce nue et du module.

Si vous souhaitez utiliser cette puce dans un circuit, il y a quatre éléments importants à prendre en compte :

- la conception du circuit et le dessin du schéma
- l'agencement du circuit imprimé
- le réglage RF et du quartz
- le téléchargement du micrologiciel et le dépannage.

Conception de circuits et dessin de schémas

Pour commencer, il est conseillé d'étudier attentivement la documentation de la puce ou du module à utiliser. Plusieurs documents sont disponibles en ligne [1]. Il s'agit notamment de la fiche technique, des

directives de conception matérielle et des conceptions de référence des modules, qui indiquent ce qu'il faut faire pour que la puce fonctionne de manière optimale.

Pour ceux qui utilisent une puce, nous présentons un circuit de base pour la puce ESP32-S3 avec tous les composants nécessaires dans la **figure 1**. Notez que contrairement aux puces sans radio intégrée, un découplage approprié est essentiel ici. Nous spécifions le nombre et les valeurs des condensateurs de découplage pour chaque broche d'alimentation ; il est important de les utiliser et de n'omettre aucun condensateur. En particulier, aux broches 2 et 3, le réseau de filtres CLC (C8, L1 et C9 dans le schéma) doit être utilisé. Sur la plupart des puces Espressif, des broches d'alimentation spécifiques sont affectées à la partie RF, où un CLC ou un filtre CCL doit toujours être utilisé pour supprimer les harmoniques élevées.

Un autre point à noter concernant l'alimentation de l'ESP32-S3 est qu'elle doit être capable de fournir au moins 500 mA. Cela s'applique fondamentalement à toutes les puces Espressif commercialisées jusqu'à présent. (La seule exception est l'ESP32-H2, qui se contente de 350 mA.) Une alimentation qui fournit un courant insuffisant conduira souvent à des réinitialisations de brownout et à d'autres phénomènes gênants, qui se produiront généralement lors de l'initialisation des connexions Wifi.

L'ESP32-S3 nécessite un quartz comme source de signal d'horloge pour l'ensemble du système. Utilisez toujours une bobine en série avec la borne XTAL_P pour supprimer les harmoniques. Les valeurs des condensateurs de découplage C1 et C4 dépendent du quartz, mais aussi de l'impédance parasite des pistes et des pastilles du circuit imprimé. Adafruit propose un bon guide pour déterminer les valeurs initiales de ces composants [2], mais pour obtenir la valeur optimale

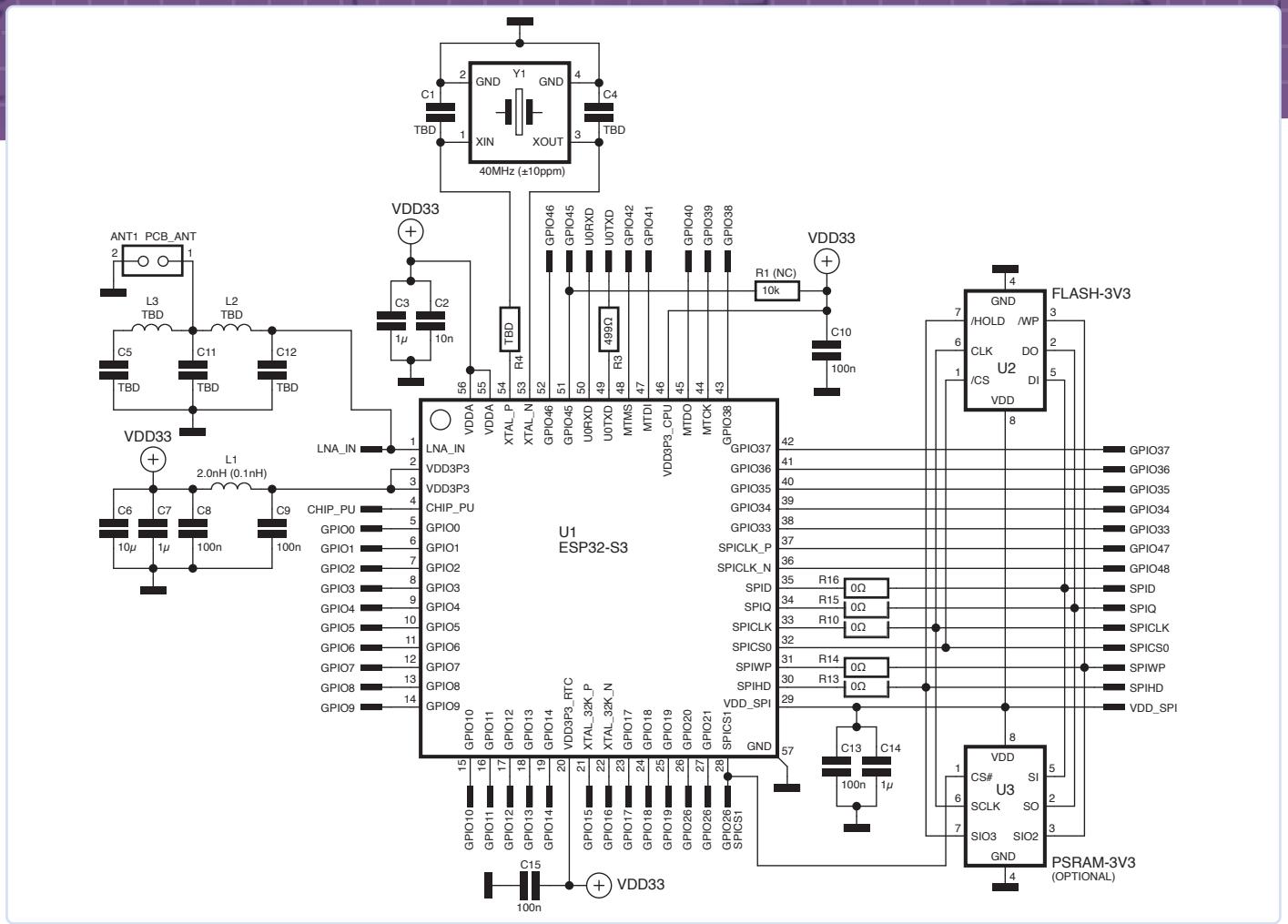


Figure 1. La puce ESP32-S3 et tous les composants qu'elle nécessite généralement. Le découplage est très important ici.

et certifiée, vous devrez les mesurer et les modifier une fois que vous aurez accès au circuit imprimé réel. Plus d'informations à ce sujet dans la suite de cet article.

L'ESP32-S3 nécessite une mémoire flash pour stocker son logiciel ; de la PSRAM peut éventuellement être ajoutée si davantage de mémoire est nécessaire. Certaines variantes de l'ESP32-S3 disposent d'une mémoire flash et/ou PSRAM dans le SoC. Dans ce cas, il est possible d'omettre U2 et/ou U3. Cela signifie qu'il n'est pas nécessaire de réservé de l'espace sur le circuit imprimé pour ces puces, ce qui permet de réduire la taille du circuit imprimé. Dans ce cas, il faut connecter des condensateurs de découplage à VDD_SPI, car ils sont nécessaires pour découpler l'alimentation de la mémoire flash/PSRAM.

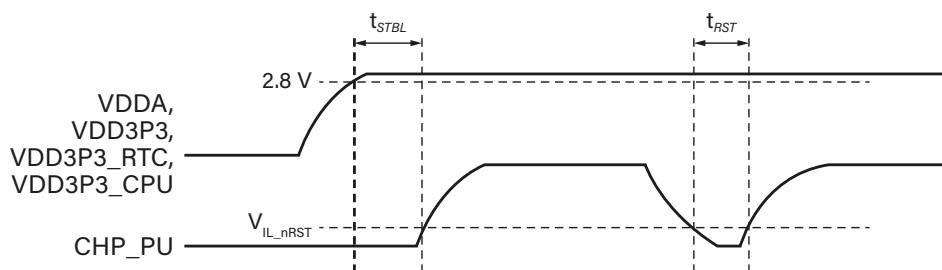
Pour obtenir la meilleure portée radio et vous assurer que le circuit est conforme aux exigences de certification CEM, vous devez vous assurer que les impédances de la piste entre l'ESP32-S3 et l'antenne sont compatibles. En général, la ligne de transmission entre l'ESP32 et l'antenne a une impédance de $50\ \Omega$, mais la pastille ESP32-S3 LNA_IN n'a pas cette impédance et l'antenne que vous utilisez peut ne pas l'avoir non plus. En d'autres termes : la connexion de l'ESP32 à la ligne de transmission nécessite un circuit d'adaptation π -CLC à proximité de l'ESP32-S3 pour adapter son impédance à $50\ \Omega$. La connexion d'une antenne à la piste peut également nécessiter un circuit d'adaptation CLC à proximité, mais vous pouvez l'omettre si vous pouvez confirmer, par exemple par simulation, que l'impédance de l'antenne est de $50\ \Omega$. Notez que dans le schéma, le réseau CLC (composé de C11, L2 et C12) n'a pas de valeurs assignées. En effet, les valeurs nécessaires dépendent du circuit et du matériau du circuit imprimé et doivent être déterminées après la phase de conception. Vous trouverez plus de détails à ce sujet dans la suite de cet article.

Si vous utilisez un module, vous n'avez pas à vous préoccuper des points susmentionnés, car le module comprend déjà les composants et les valeurs corrects.

Si vous utilisez un module ou une puce : La broche CHIP_PU (marquée EN sur le module) sert à la fois de broche d'activation et de réinitialisation ; elle ne doit pas être laissée flottante. Afin de démarrer de manière fiable la puce ESP32-S3, la broche CHIP_PU ne doit être activée que lorsque l'alimentation électrique est stable. (Voir l'encadré **Paramètres de temporisation pour la mise sous tension et la réinitialisation.**)

Généralement, un circuit RC est ajouté à cette broche pour générer un délai. Certains produits doivent fonctionner dans un scénario où la rampe d'alimentation est très lente, ou nécessite des mises sous tension et hors tension fréquentes, ou encore lorsque l'alimentation est instable. (Dans ce cas, utiliser uniquement un circuit RC peut être insuffisant pour respecter les délais de la séquence de mise sous tension et hors tension, ce qui peut conduire à un échec de démarrage de la puce. Dans ce cas, nous suggérons d'utiliser d'autres moyens pour répondre aux exigences, tels que l'utilisation d'une puce externe de supervision de l'alimentation ou d'une puce chien de garde. Si votre circuit le permet, vous pouvez également contrôler la broche CHIP_PU par un autre microcontrôleur, ou simplement ajouter un bouton de réinitialisation. La dernière étape pour faire fonctionner l'ESP32-S3 avec succès est de faire attention aux broches de strapping. À chaque démarrage ou réinitialisation, les puces Espressif ont besoin de certains paramètres de configuration initiaux, tels que le mode de démarrage de la puce, la tension de la mémoire flash, etc. Ces paramètres sont sélectionnés via les broches de strapping. Après la réinitialisation, les broches de strapping fonctionnent comme des broches d'E/S normales. Vous trouverez tous les détails dans la fiche technique de la puce.

Paramètres temporels pour le démarrage et la réinitialisation



Visualisation des paramètres temporels pour le démarrage et la réinitialisation. (Source [5])

Paramètre	Description	Min (μs)
t_{STBL}	Temps réservé à la stabilisation des rails 3,3 V avant que la broche CHIP_PU ne soit tirée vers le haut pour activer la puce.	50
t_{RST}	Temps réservé à CHIP_PU pour rester inférieur à V_{IL_nRST} afin de réinitialiser la puce	50

À ce stade, notre ESP32-S3 peut démarrer avec succès. Cependant, vous voudrez probablement la connecter à d'autres puces, capteurs et actionneurs. Quels GPIOs pouvez-vous utiliser pour cela ? Bonne nouvelle : toutes les puces Espressif disposent d'une fonction appelée GPIO Matrix. Elle facilite grandement la connexion à l'ESP32-S3, car elle permet d'affecter TOUT GPIO disponible à n'importe quel signal de la plupart des périphériques généraux (par exemple, I2C, SPI, SDIO, etc.). Bien sûr, certains périphériques utilisent encore des GPIO fixes (par exemple, ADC). Reportez-vous à la fiche technique de la puce pour plus d'informations si nécessaire. D'une manière générale, n'oubliez pas de lire d'abord les documents techniques – il vaut mieux relire des informations que vous connaissez déjà que de devoir refaire un circuit imprimé à cause d'une erreur évitable.

Disposition du circuit imprimé

Si vous utilisez une puce, vous devriez avoir un schéma parfait pour dessiner la disposition. Pour les produits Wi-Fi et Bluetooth, l'agencement du circuit imprimé est essentiel pour obtenir des performances RF qualifiées. Même si vous n'utilisez qu'une puce Espressif comme microcontrôleur, sans ajouter d'antenne ou utiliser la radio, les directives ci-dessous vous permettront d'obtenir un système stable et une

fiabilité accrue. Notez que tous les détails exacts sont disponibles dans les lignes directrices pour la conception du matériel. Nous nous contenterons ici de généraliser les points importants. Commençons par l'empilement des couches. Quatre couches ou plus sont recommandées car, ainsi, nous pouvons avoir un plan de masse large et continu adjacent à la couche de la puce comme couche de masse de référence. Si vous souhaitez vraiment utiliser deux couches, n'oubliez pas de vous assurer que vous disposez toujours d'un bon plan de masse.

Le deuxième point important est le routage des pistes d'alimentation. Nous suggérons de les router en étoile sur une couche intérieure. Chaque condensateur de découplage et circuit de filtrage CLC doit être aussi proche que possible des broches d'alimentation (**figure 2**). Le troisième point consiste à prendre soin de vos tracés de pistes RF. Vous devez demander à l'usine d'avoir une impédance contrôlée de $50\ \Omega$ pour tous les pistes RF. Cela commence au stade de la disposition du circuit imprimé, où vous êtes censé calculer la largeur des pistes et l'écart par rapport à la masse sur la base de votre empilage. L'usine peut ensuite les ajuster légèrement. Si les composants sont dans le format 0201 SMD, veuillez utiliser un stub dans la conception du circuit RF adapté à proximité de la puce.

Vient ensuite le quartz. La bobine en série doit être très proche de la puce ESP32-S3, et les deux condensateurs de charge doivent être placés de deux côtés du quartz (**figure 3**).

Une dernière chose à laquelle il faut prêter attention sont les pistes de la flash et de l'UART, ainsi que les pistes qui vont vers d'autres pastilles GPIO périphériques. Ces pistes peuvent générer des harmoniques à haute fréquence, il est donc conseillé de les acheminer sur les couches internes et de les entourer de masse autant que possible.

Si vous utilisez un module, le point critique est de savoir comment le placer pour obtenir les meilleures performances RF. Les lignes directrices suivantes ne concernent que les modules dotés d'une antenne sur le circuit imprimé ; si vous utilisez un module doté d'une antenne externe, vous disposez d'une plus grande marge de manœuvre quant à l'emplacement de celle-ci sur le circuit imprimé.

Comme indiqué dans la **figure 4**, nous vous suggérons de placer le module dans un coin de la carte. (Notez également que le point d'alimentation ne se trouve pas du même côté pour tous les types de modules ; là encore, vérifiez la fiche technique avant de concevoir le circuit imprimé). Nous recommandons que l'antenne soit complètement à l'extérieur de la carte. Si cela n'est pas possible, assurez-vous qu'il y a un espace libre sous et au moins 15 mm autour de l'antenne :

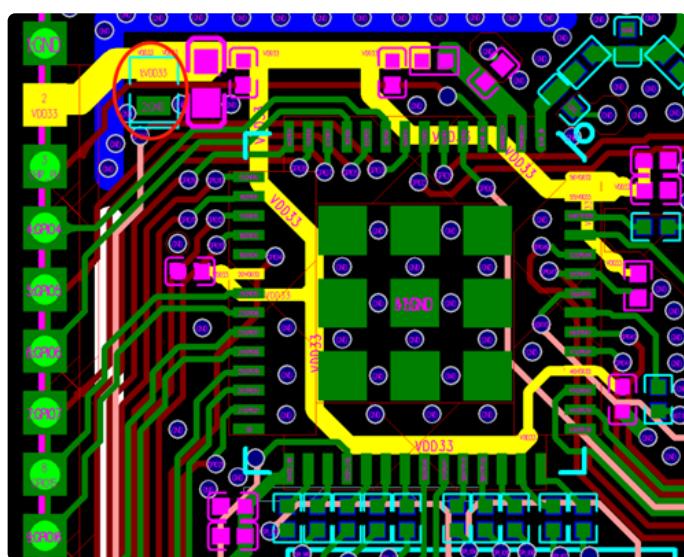


Figure 2. Chaque condensateur de découplage doit être aussi proche que possible des broches d'alimentation.

$$dW = \int [\psi(x_0, y_0, z_0)] dV = 4.4 \text{ mJ}$$

cela signifie qu'il ne doit pas y avoir de pistes, de plans, de vias, ou de composants à cet emplacement (**figure 5**).

Bien qu'il soit assez facile de souder la plupart des connexions de nos modules Wroom avec un fer à souder standard, il y a une connexion de masse en dessous qui est normalement inaccessible à la main. Si vous avez la possibilité de le faire (par exemple, en utilisant un four à refusion ou une station de soudage à air chaud), nous vous conseillons de la connecter à un plan de masse, car cela permet de dissiper la chaleur et d'obtenir une meilleure mise à la masse pour la radio, mais dans des circonstances normales, le module fonctionnera également sans cette connexion.

Réglage RF et du quartz

Si vous utilisez une puce : si vous avez suivi tous les critères ci-dessus et que vous avez reçu un circuit imprimé fonctionnel de l'usine, pour une meilleure fiabilité et pour passer la certification CEM, vous aurez malheureusement besoin d'équipements professionnels pour régler les valeurs d'adaptation et les condensateurs du quartz. Si vous avez accès à ces équipements (ou si vous disposez de produits peu coûteux comme le MicroVNA), nous allons vous expliquer en quoi consiste cette procédure.

Vous avez probablement choisi les valeurs des condensateurs de charge en fonction de la capacité parasite estimée des pistes. Il est maintenant temps de les ajuster : parce que le 2,4 GHz principal est généré par ce quartz, l'optimisation de son fonctionnement vous permettra d'obtenir une meilleure portée et des harmoniques plus faibles. Nous allons optimiser cela en utilisant l'outil ESP RF Test que vous pouvez télécharger sur notre site [3].

1. Sélectionnez le mode de *TX tone* avec l'outil *Certification and Test*.
 2. Observez le signal de 2,4 GHz avec un analyseur de communication radio ou un analyseur de spectre et démodulez-le pour obtenir le décalage de fréquence réel.
 3. Réglez le décalage de fréquence à ± 10 ppm (recommandé) en ajustant la valeur de la capacité de charge externe.
- Lorsque le décalage de la fréquence centrale est positif, cela signifie que la capacité de charge équivalente est trop faible et que la capacité de charge externe doit être augmentée.
 - Lorsque le décalage de la fréquence centrale est négatif, cela signifie que la capacité de charge équivalente est trop grande et que la capacité de charge externe doit être réduite.
 - Les condensateurs de charge externe des deux côtés sont généralement égaux, mais dans des cas particuliers, ils peuvent avoir des valeurs légèrement différentes.

Une fois les valeurs des condensateurs de charge du quartz réglées , nous passons au réseau d'adaptation de l'antenne. Dans le circuit d'adaptation, nous définissons le port proche de la puce comme Port 1 et le port proche de l'antenne comme Port 2. S11 décrit le rapport entre la puissance du signal réfléchi par le Port 1 et la puissance du signal d'entrée, et S21 est utilisé pour décrire la perte de transmission du signal du Port 1 au Port 2. Pour la série de puces ESP32-S3, si S11 est inférieur ou égal à -10 dB et S21 est inférieur ou égal à -35 dB lors de la transmission des signaux 4,8 GHz et 7,2 GHz, le circuit d'adaptation peut satisfaire aux exigences de transmission.

Connectez les deux extrémités du circuit d'adaptation à l'analyseur de réseau (voir l'encadré « Circuit d'adaptation ») et testez le paramètre de

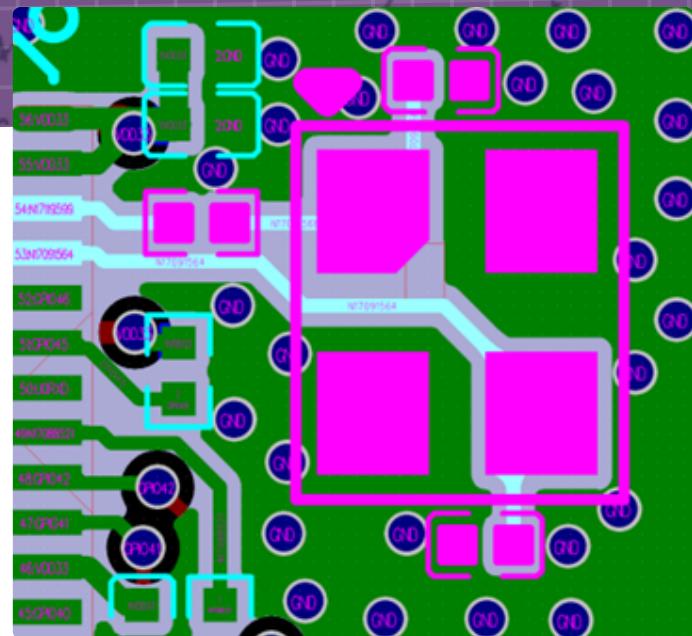


Figure 3. Les deux condensateurs de charge doivent être placés de part et d'autre du quartz.

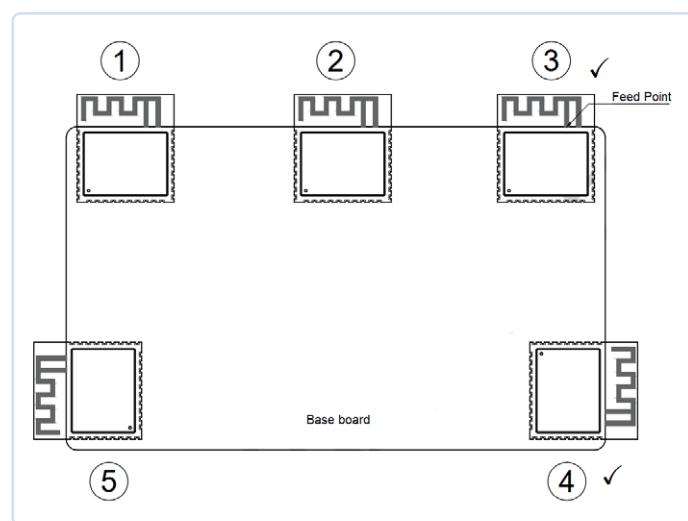


Figure 4. Montage d'un module sur une carte de base. (Source : [6])

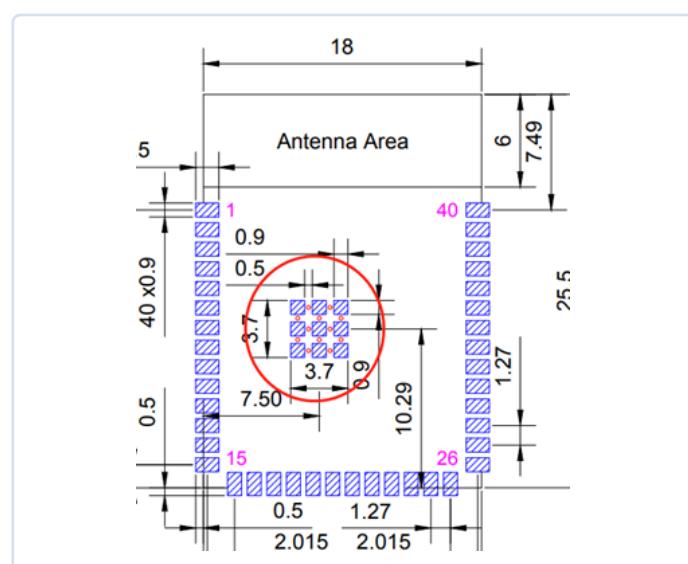
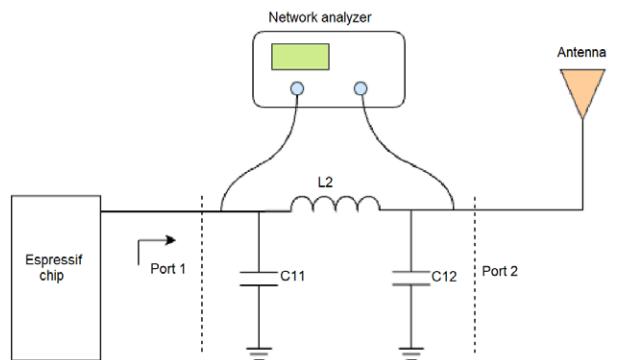


Figure 5. Circuit imprimé recommandé pour l'ESP32-S3-WROOM-1. (Source : [6])

Matching Circuit



Source [7]

Indicateur de référence	Valeur recommandée	Numéro de série
C11	1.2 ~ 1.8 pF	GRM0335C1H1RXBA01D
L2	2.4 ~ 3.0 nH	LQP03TN2NXPB02D
C12	1.8 ~ 1.2 pF	GRM0335C1H1RXBA01D

réflexion S11 et le paramètre de transmission S21 du signal. Ajustez les valeurs des composants du circuit jusqu'à ce que S11 et S21 répondent aux exigences. Si la conception du circuit imprimé de la puce respecte strictement les directives de conception du circuit imprimé et que l'antenne est bien conçue, vous pouvez vous référer aux plages de valeurs dans l'encadré pour déboguer le circuit d'adaptation.

Alors, que faire lorsque vous voulez intégrer une puce ESP32-S3 dans votre carte, mais que vous ne disposez pas des équipements sophistiqués pour le faire correctement ? Si vous avez suivi tous les autres points de cet article, vous pourrez peut-être utiliser les valeurs estimées des condensateurs de charge et ne pas utiliser de réseau CLC. (Si vous en avez quand même conçu un dans votre circuit imprimé, vous pouvez supprimer C11/C12 et remplacer L2 par une résistance de zéro ohm. Ainsi, vous pourrez ajouter un réseau CLC plus tard sans avoir à modifier la disposition du circuit imprimé). D'après notre expérience, l'absence d'un réseau d'adaptation d'impédance n'affecte pas trop la portée ; sa principale utilité est de réduire l'EMV (Error Magnitude Vector) des appareils, ce qui est nécessaire pour satisfaire aux exigences de certification.

Si vous utilisez un module : aucune des opérations susmentionnées n'est nécessaire, car votre module est livré avec tous ses composants pré-réglés pour des performances optimales.

Téléchargement du micrologiciel et résolutions de problèmes

Si vous utilisez un module ou une puce : avant que votre ESP32-S3 ne réalise quoi que ce soit, vous devrez charger un micrologiciel dans sa mémoire flash (ce qu'on appelle la programmation ou le téléchargement d'un micrologiciel). Pour toutes les puces Espressif, le processus de téléchargement est le suivant : réinitialiser la puce en mode téléchargement – télécharger – réinitialiser et faire fonctionner la puce en mode SPI Boot.

Vous pouvez programmer les puces Espressif généralement de deux façons : toutes les puces Espressif supporteront le téléchargement de micrologiciel via l'UART, mais les puces les plus récentes ont généralement aussi un périphérique USB qui vous permet de télécharger le micrologiciel directement via une connexion USB à votre ordinateur. Vous trouverez ci-dessous les étapes détaillées pour l'ESP32-S3.

Pour télécharger via UART :

- Avant le téléchargement, assurez-vous de mettre la puce ou le module en mode *Download Boot*, c'est-à-dire que la broche de strapping GPIO0 (tirée vers le haut par défaut) est tirée vers le bas et que la broche GPIO46 (tirée vers le bas par défaut) est laissée flottante ou tirée vers le bas. Configurez la broche GPIO45 de manière appropriée selon le tableau des broches de strapping, ou laissez-la flottante si vous utilisez un module.
- Mettez la puce ou le module sous tension et vérifiez s'il est entré en mode UART Download via le port série UART0. Si le journal indique « *waiting for download* », la puce ou le module est entré en mode *Download Boot*.
- Téléchargez votre micrologiciel dans la mémoire flash via UART. Vous pouvez utiliser n'importe quelle option de programmation que votre environnement de programmation (Arduino, ESP-IDF, VSCode...) vous donne ; ou vous pouvez utiliser l'outil autonome *Flash Download Tool*.
- Une fois le micrologiciel téléchargé et la puce entrée dans le mode *USB Download*, la procédure est terminée. Si ce n'est pas le cas, tirez IO0 vers le haut ou laissez-la flottante pour vous assurer que la puce ou le module entre en mode SPI Boot. Réinitialisez ou mettez sous tension la puce ou le module, et il lira et exécutera le nouveau micrologiciel pendant l'initialisation. Si vous ne parvenez pas à télécharger le micrologiciel via l'UART, vérifiez le journal de l'UART0 avec un outil de terminal série. Le message de démarrage de l'ESP32-S3 vous indiquera la valeur réelle



À propos de l'auteur

Liu Jinghui, diplômée en ingénierie électronique, a rejoint Espressif et s'est consacrée à la résolution des problèmes de matériel. Pendant cinq ans, elle a été la représentante des produits Espressif et de leurs utilisations. Elle a une bonne compréhension des caractéristiques matérielles des produits et des problèmes qui peuvent survenir lors de leur utilisation par les clients.

de verrouillage des broches de strapping, ce qui vous permettra de déterminer quelle valeur est erronée. La valeur après `boot`: montre les valeurs des broches de strapping en hexadécimal : bit 2 = GPIO46, bit 3 = GPIO0, bit 4 = GPIO45, bit 5 = GPIO3 :

`ets Jun 8 2016 00:22:57`

`rst:0x1 (POWERON_RESET),boot:0x3 (DOWNLOAD_BOOT(UART0/UART1/SDIO_RESET,REO_V2))`

Si vous ne reconnaissiez pas le périphérique USB ou si la connexion USB est instable, essayez d'abord de passer en mode téléchargement, puis de télécharger. Rappelez-vous également que du côté de l'ordinateur, un seul programme doit ouvrir le port série à la fois : essayer de flasher tout en ayant un terminal série ouvert sur le même port est voué à l'échec.

Notez que si le passage en mode téléchargement par USB est généralement fiable, il existe quelques scénarios dans lesquels il peut échouer :

- USB PHY est désactivé par l'application ;
- Le port USB est reconfiguré pour d'autres tâches, par exemple, hôte USB, appareil standard USB ;
- Les GPIO USB sont reconfigurés, par exemple comme sorties pour LEDC, SPI, ou comme GPIO génériques.

Dans ce cas, il est nécessaire d'entrer manuellement en mode téléchargement. C'est pourquoi, au moins pendant le développement du micrologiciel, nous suggérons d'avoir toujours un moyen (boutons, cavaliers, ...) de configurer les broches de strapping pour forcer un démarrage en mode téléchargement.

Une autre chose à noter : parfois, certains constatent que leur carte ne démarre pas correctement (c'est-à-dire qu'elle ne commence pas à exécuter leur programme en flash) à moins qu'ils ne la réinitialisent manuellement. Cela peut se produire lorsqu'un condensateur est placé sur une broche de strapping (par exemple, GPIO0 sur l'ESP32-S3). Un tel condensateur est parfois placé pour amortir un bouton, mais lorsque l'appareil est mis sous tension, le niveau du signal du GPIO0 augmente lentement, ainsi, l'ESP32-S3 le considère comme une valeur basse.

Les modules font la magie

Comme vous pouvez le voir dans cet article, bien qu'il soit possible de réaliser un projet fonctionnel, concevoir une carte intégrant un ESP32-S3 avec des performances optimisées et une certification réussie n'est pas trivial et nécessite des connaissances approfondies ainsi que

des outils. Ainsi, si vous disposez de l'espace nécessaire, nous vous suggérons d'utiliser plutôt un module ; toute la magie RF est déjà faite par Espressif, ce qui rend l'intégration d'un module dans votre projet beaucoup plus facile.

Si tout cela vous semble encore trop compliqué, comme indiqué au début de l'article, Espressif propose également une large gamme de cartes de développement, des plus simples qui déplient toutes les GPIO vers des connecteurs faciles à utiliser comme l'ESP32-S3-DevkitC-1, aux plus puissantes cartes avec la reconnaissance vocale intégrée comme l'ESP32-S3-Box-3 et l'ESP32-S3-Eye, dotée d'un appareil photo. En général, les schémas et les designs de ces cartes de développement sont disponibles, donc même si vous ne prévoyez pas de les utiliser dans votre produit, ils peuvent constituer un excellent point de départ pour votre projet. ↗

230563-04

Questions ou commentaires ?

Envoyez un courriel à l'auteure (liujinghui@espressif.com) ou contactez Elektor redaction@elektor.fr).



Produits

- **ESP32-S3-WROOM-1**
www.elektor.fr/20696

LIENS

[1] Documents techniques d'Espressif :

https://espressif.com/en/support/documents/technical-documents?keys=&field_type_tid%5B%5D=842

[2] Adafruit, "Choosing the Right Crystal and Caps for your Design," 2012 :

<https://blog.adafruit.com/2012/01/24/choosing-the-right-crystal-and-caps-for-your-design>

[3] ESP RF Test Tool : <https://espressif.com/en/support/download/other-tools>

[4] Schéma de l'ESP32-S3-DevKitC-1 : https://dl.espressif.com/dl/schematics/SCH_ESP32-S3-DevKitC-1_V1.1_20221130.pdf

[5] Fiche technique de la série ESP32 : https://espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

[6] ESP32-H2-Series Hardware Design Guidelines :

https://espressif.com/sites/default/files/documentation/esp32-h2_hardware_design_guidelines_en.pdf

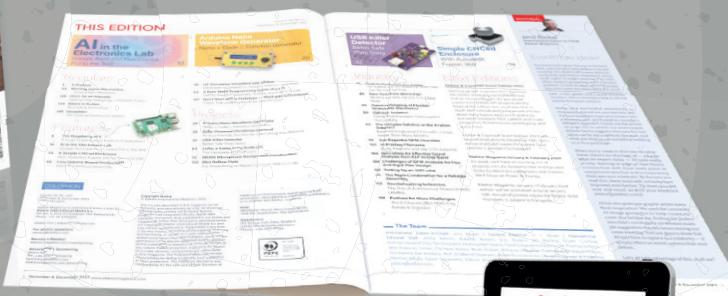
[7] Fiche technique de l'ESP32-S3-WROOM-1 :

https://espressif.com/sites/default/files/documentation/esp32-s3-wroom-1_wroom-1u_datasheet_en.pdf

20%
de réduction
sur la première année de
votre abonnement

Rejoignez la communauté **Elektor**

Devenez membre
maintenant !



- accès à l'archive numérique depuis 1978 !
- 8x magazine imprimé Elektor
- 8x magazine numérique (PDF)
- 10 % de remise dans l'e-choppe et des offres exclusives pour les membres
- accès à plus de 5000 fichiers Gerber
- Livraison gratuite en France



www.elektormagazine.fr/gold-member

Utilisez le code promo :

ESPRESSIF20



ESPRESSIF

 **Elektor**