

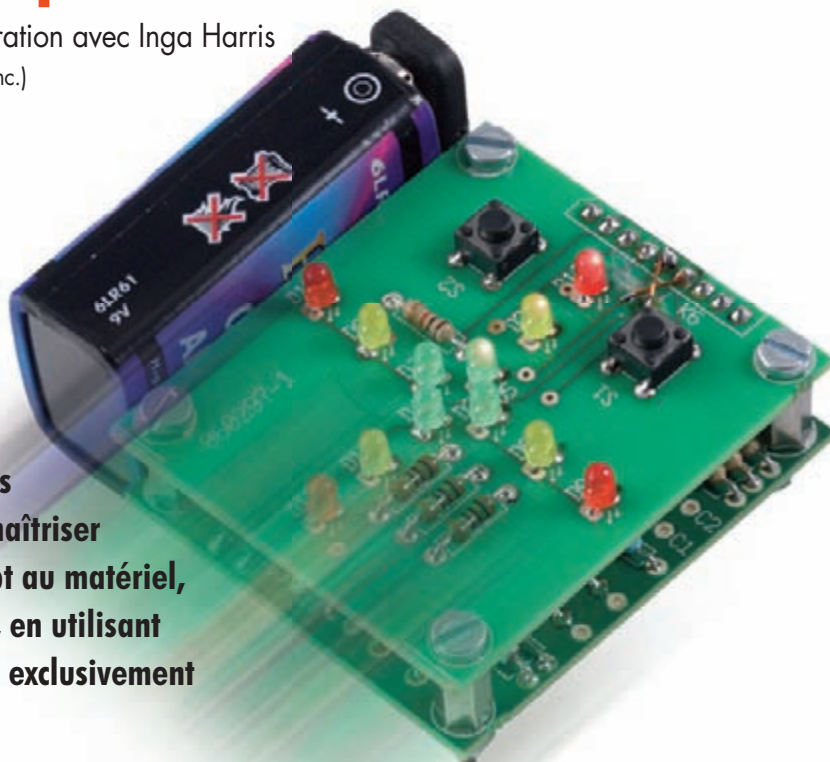
g Sur des LED

Accéléromètre bi-axe

avec SpYder et MC9S08 de Freescale

Jan Buiting & Luc Lemmens, en coopération avec Inga Harris
(Applications Engineer, Freescale Semiconductor Inc.)

Voici une application aussi instructive qu'amusante du microcontrôleur MC9S08 de Freescale. Il s'agit de la suite de l'article « SpYder attaque » du mois dernier. Elle devrait vous aider à maîtriser la technique de passage du concept au matériel, à travers un parcours peu onéreux, en utilisant trois instruments gratuits proposés exclusivement aux lecteurs d'Elektor.



Ce mois-ci, tout est déjà en place, la théorie, la pratique, les composants gratuits et le logiciel qui, réunis, constituent un projet de construction for-



Figure 1. Le Kit SpYder Discovery dans sa récente métamorphose (le mois dernier, nous vous en avions présenté la version bêta). À la faveur d'un arrangement spécial avec Freescale Semiconductor, vous pouvez acheter l'ensemble auprès d'Elektor au prix d'à peine 9,75 euros (+ frais de port et d'emballage) Notez qu'il faut vous procurer ou fabriquer vous-même le câble en nappe BDM à 6 conducteurs.

mateur et plaisant, que vous voudrez certainement installer sur votre voiture, votre modèle radiocommandé ou votre vélo. On peut même envisager de mettre en service cet accéléromètre sur un dragster, en vrai ou en modèle réduit ! Outre les deux articles du magazine (le second, vous l'avez entre les mains), toutes les pièces qui vont contribuer à édifier ce projet sont répertoriées dans le **tableau 1**. Si le kit SpYder et le module d'accéléromètre offert avec les deux platines du projet sont d'un prix aussi abordable, c'est le fruit de la mise en commun des efforts d'Elektor et de Freescale, en coopération exclusive, au bénéfice des lecteurs d'Elektor.

Objectifs du projet

Les acteurs cités ont décidé de publier ce projet avec, derrière la tête, des exigences ambitieuses et la mise à disposition de tout le nécessaire. Dressons la liste des unes et des autres, les solutions trouvées sont ici en *italique*.

1. Le microcontrôleur utilisé doit être

bon marché, gratuit si possible. *Le choix s'est porté sur le MC9S08 dont les échantillons sont disponibles gratuitement auprès de Freescale.*

2. Le microcontrôleur doit être accompagné de tous les outils de programmation et de débogage adaptés. *CodeWarrior et le BDM USB de SpYder feront parfaitement l'affaire.*
3. Le lecteur doit disposer d'un soutien en ligne. *Les forums de Freescale et d'Elektor sont là pour ça.*
4. Le projet doit être instructif et « libre ». *Tous les fichiers de code source, les feuillets de caractéristiques et les notes de développement sont disponibles gratuitement.*
5. Le projet doit se raccrocher à la réalité vécue. *Un accéléromètre à deux axes dans une gamme de 2g, avec écran LCD, c'est du concret.*
6. Par égard pour les étudiants, en particulier, le coût total doit rester aussi bas que possible. *Le module d'accéléromètre et le MC9S08 sont gratuits, les platines et le Kit de découverte SpYder sont peu coûteux.*

| Objet | Source / Fournisseur | Description | Comment l'obtenir | Coûts |
|----------------------|-------------------------|---|---|--|
| SpYder Discovery Kit | Elektor | Contient clé USB BDM, 1 échantillon MC9S08 8 broches, CD-ROM Code Warrior & utilitaires | Code commande 060296-91 (cf. e-choppe Elektor) | € 9,75 (+ port et emballage). Gratuit avec module ou kit commandé à l'e-choppe d'Elektor |
| MC9S08QG8CPBE | Freescale | Microcontrôleur PDIP 16 broches de Freescale | À demander via le service d'échantillons gratuits de Freescale | Gratuit |
| Set de platines | Elektor | Paire de platines pour réalisation personnelle avec composants standard | Code commande de 060297-71 (cf. e-choppe Elektor) | € 14,50 |
| MMA7260Q | Elektor | Accéléromètre Freescale monté sur platine gigogne | Deux (!) exemplaires inclus (gratuitement) avec set de platines 060296-71 | Gratuit |
| Autres composants | Fournisseur local / VPC | Différents composants standard tels que mentionnés dans la liste des composants | Acheter cher marchand de composant ou par VPC, Farnell par exemple | De l'ordre de € 7,50 |

Tableau 1. Quoi — Où — Comment

Récapitulatif des matériels/logiciels nécessaires à la réalisation du projet accéléromètre 2 axes

Une liste impressionnante à faire dresser l'oreille des dirigeants et des départements commerciaux un peu partout, mais aussi matière à faire grincer des dents à propos des artisans de ce projet. Avec la publication dans ce magazine, le défi peut s'étendre jusqu'à vous, si vous souhaitez démarrer dans la technologie enfouie, en partant de microcontrôleurs 8 bits à bas prix qui ont fait leurs preuves dans les applications automobiles (il est fort probable que votre prochaine nouvelle voiture en contiendra au moins un).

L'accéléromètre

Un accéléromètre est un appareil qui mesure et indique la force exercée sur un corps en mouvement, due à l'accélération (+g) ou la décélération (-g). Le projet actuel mesure ces forces jusqu'à 2 g dans deux plans : avant/arrière et gauche/droite.

L'affichage se compose de LED colorées, les valeurs les plus grandes sont indiquées par des LED rouges. L'instrument est alimenté sur piles et peut se loger dans un véhicule pour vous dévoiler, à vous ou à vos passagers (comme les enfants sur le siège arrière) un tas de choses sur vos habitudes de conduite.

SpYder de retour, mais par étapes

Avant de se lancer sur le projet et d'agripper ce cher fer à souder, quelques mots à propos de ce nouveau copain au pays des 8 bits enfouis. SpYder (**figure 1**) s'en remet totalement au BDC (*Background Debug Controller*) HCS08 et RS08, ce qui permet une programmation rapide et facile de la mémoire Flash embarquée et des autres mémoires. C'est l'interface de débogage de première ligne pour le

développement. Elle autorise l'accès non intrusif aux données en mémoire et les possibilités de débogage traditionnel, comme la modification du point d'arrêt dans le registre du processeur et la commande d'analyse instruction par instruction. Si vous avez utilisé le HC05 et les autres noyaux plus anciens de Freescale, sur lesquels nous avons trimé, vous apprécierez les progrès dont ce module vous gratifie.

L'outil SpYder se sert d'une interface USB standard pour communiquer avec le PC ainsi que de la tension présente sur le bus USB pour alimenter l'outil et le microcontrôleur, sans devoir recourir aux adaptateurs secteur encombrants de naguère, ce qui le rend carrément portable. Cette alimentation USB peut aussi desservir la platine cible, pour autant qu'elle ne compte pas trop de moteurs à faire tourner (jusqu'à 100 mA) !

Les cartes destinées à la construction du projet sont livrées avec deux capteurs d'accéléromètre MMA7260 gratuits, montés sur platines de soutien – un cadeau d'Elektor et de Freescale

Comme l'outil requiert l'usage de l'USB, le centre nerveux de SpYder est un microcontrôleur MC908JB16 de Freescale. Ce micro dispose en effet d'une interface USB (2.0 basse vitesse) et travaille à partir du 5 V fourni par l'USB. Et puisque l'outil accepte aussi les microcontrôleurs RS08, il faut du 12 V pour programmer les composants RS08. Le MC908JB16 est en mesure de reconnaître la cible et, par la commande de PTDO, peut autoriser le passage du signal de 12 V généré par une puce de conversion de CC à CC du type ST662.

1re étape : l'assemblage de la carte SpYder

Les pièces principales de l'outil SpYder ont déjà été installées à votre intention sur la carte, vous pourrez donc l'utiliser comme outil autonome avec le microcontrôleur présent dans le support.

Rien ne vous empêche d'employer un autre modèle de boîtier ou votre propre platine cible (comme dans le cas de l'application en accéléromètre). Il suffit d'ajouter une résistance de 0 Ω ou un court-circuit, à côté du champ référencé R2, pour connecter effectivement l'alimentation au support BDM de manière à mettre en service une autre platine cible. Vous obtiendrez du même coup un Kit de découverte SpYder transformé en module BDM, mais pas de tracas, in continuera à fonctionner en outil autonome. Si vous ne possédez pas encore de câble BDM, vous pouvez facilement vous en fabriquer un. Tout ce qu'il vous faut, c'est deux fiches IDC à 6 contacts avec frein de

câble (code de commande 1097021 chez Farnell) et du câble plat au pas de 1,27 mm (code 9187111 de Farnell). On presse les extrémités du câble sur les connecteurs IDC au moyen d'un petit étai. Ne dépassez pas une longueur de câble de 30 cm environ.

Rappelez-vous qu'il ne faut qu'un seul microcontrôleur branché à la fois : ou bien dans le support sur la carte SpYder ou bien sur la platine cible.

2e étape : installation du débogueur

Le Studio de développement CodeWarrior v5.1 pour les HC(S)08/RS08 de

Freescale est disponible, libre de charges, dans son avatar Édition spéciale, le CD SpYder en contient une copie. L'inventaire des différences entre cette édition spéciale et les éditions standard et professionnelle, vous le trouverez sur www.freescale.com/codewarrior. Sans numéro de licence, le logiciel tourne en mode de démonstration, il ne traite que 1 Koctet de code. Pour forcer cette limite, vous disposez de deux possibilités :

1. contacter Freescale pour demander un numéro de licence sans restriction de temps pour traiter jusqu'à 16 Koctets ;
2. contacter Freescale pour demander un numéro de licence limitée à 30 jours, mais sans plafond dans le volume à compiler.

3e étape : les pilotes

Dès que la platine se trouve physiquement dans l'état souhaité et que CodeWarrior est installé, il faut mettre en contact le SpYder et votre débogueur. Ensuite, il s'agit d'installer les pilotes SpYder disponibles sur le CD qui l'accompagne. Comment procéder, c'est ce que vous trouverez sur la face intérieure de la couverture. Lors de la première connexion de l'outil sur le PC, Windows reconnaît un nouvel appareil USB et affiche la fenêtre de dialogue de l'assistant d'installation d'un nouveau matériel pour vous demander le pilote nécessaire. Pour terminer l'installation, quoi de plus simple que de demander « Install Automatically » ?

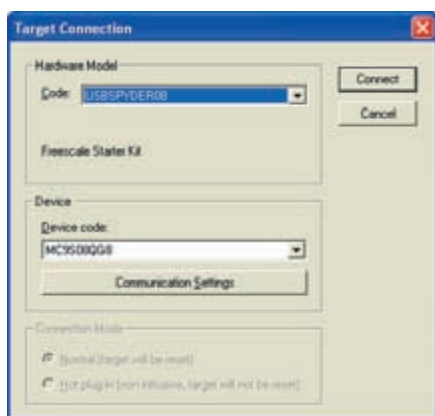


Figure 2. En CodeWarrior, assurez-vous de sélectionner la bonne connexion de débogage vers le PC.

Prêt pour partir à la découverte

Il importe de se rendre compte que SpYder utilise le microcontrôleur cible pour l'exécution en circuit et pas un émulateur, donc que les périphériques du microcontrôleur, tels que tempori-

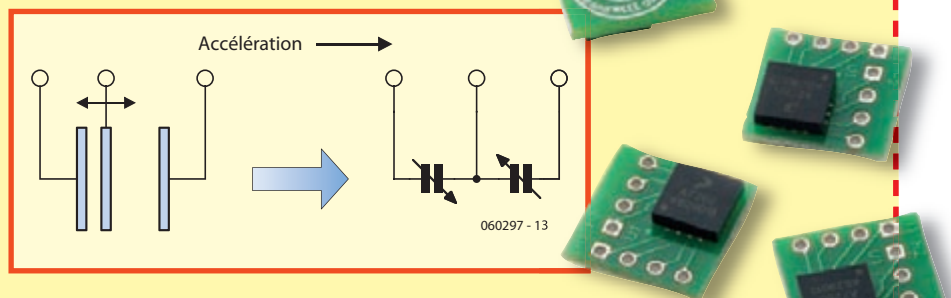
Le capteur d'accélération MMA7260Q

Comme Freescale s'est solidement implanté dans l'électronique automobile, il ne faut pas s'étonner que la société propose une imposante gamme de capteurs d'accélération. Plus que probablement, le feu stop de votre nouvelle voiture sera commandé par l'un d'entre eux !

La communauté électronique n'a pas tardé, elle non plus, à découvrir les avantages des capteurs de g de Freescale pour les incorporer aux modèles réduits, que ce soit des fusées, des modèles radiocommandés ou autres. Nous publierons dans le numéro de mai 2007 une application de ce genre.

Résultat d'un usinage de précision, le MMA7260 est un accéléromètre capacitif à prix comprimé, avec conditionnement du signal, filtre passe-bas à un pôle, compensation de température et sélection de gamme sur quatre sensibilités. Le décalage de zéro g à pleine échelle et le filtre de coupure sont compensés d'usine et ne demandent aucun composant extérieur. L'appareil dispose d'un mode de veille qui le rend particulièrement apprécié en électronique portable alimentée par piles.

La cellule g dans le MMA7260Q est une structure mécanique en matériau semi-conducteur (polysilicone) fabriquée selon les procédés des semi-conducteurs, par masque et gravure chimique. On peut le représenter comme un ensemble de rayons attachés à une masse centrale mobile qui se déplace entre des rayons fixes. Les rayons mobiles peuvent subir une déflexion par rapport à leur position de repos sous l'effet de l'accélération appliquée au système.



Si les rayons attachés à la masse centrale bougent, la distance entre eux et les rayons fixes augmente d'un côté de la même amplitude qu'elle décroît de l'autre côté. Le changement de distance est la mesure de l'accélération. Les rayons de la cellule g forment deux condensateurs en opposition. Si le rayon central bouge avec l'accélération, la distance entre les rayons se modifie et la capacité de chaque condensateur va changer aussi, comme l'exprime l'équation :

$$C = A \Sigma / D$$

dans laquelle A est l'aire du rayon, Σ est la constante diélectrique et D est la distance entre les rayons. L'ASIC sur la puce se sert de la technique de commutation de condensateurs pour mesurer les capacités de la cellule g et extraire la donnée d'accélération de la différence entre les deux condensateurs. L'ASIC conditionne aussi le signal et le filtre (condensateurs commutés) pour assurer un niveau de sortie élevé qui est ratiométrique (la mesure d'un rapport) et proportionnel à l'accélération.

Les deux appareils MMA7270Q à montage en surface, installés sur des platines porteuses, sont compris dans la livraison des platines pour le projet décrit ici. L'appareil est normalement vendu au prix de 5,95 \$US la pièce par quantité de 1 000 ou plus. Il est alors livré sans platine de soutien (!). Pour votre facilité, les fiches de caractéristiques du MMA7270Q sont incluses dans le téléchargement gratuit de ce mois-ci pour le projet.

Vous trouverez des notes d'application (et des vidéos !) de cet appareil attrayant sur : http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=MMA7260QT&nodeId=01126911184209

sateur, convertisseur A/N, modules de communication série ou autres, ne sont pas reconstitués par le logiciel ou un quelconque appareil externe.

À la première entrée en session de débogage, une boîte de dialogue pour la configuration du microcontrôleur s'ouvre et vous demande de sélectionner la connexion du matériel de débogage vers le PC. Assurez-vous que la connexion choisie est du type

« USBSPYDER08 » (figure 2). À ce moment, CodeWarrior efface et reprogramme la mémoire et ajuste également l'oscillateur s'il y en a un de disponible sur le microcontrôleur choisi. Vous disposez à présent de tout le nécessaire pour entamer le débogage de votre code, le tout est de savoir comment y procéder. CodeWarrior permet toute une gamme de méthodes pour analyser le déroulement du pro-

Liste des composants

Résistances :

- R1 = 10 Ω
- R2 = 4kΩ7
- R3 à R7, R9, R12, R13 = 1 kΩ
- R8, R10, R11 = 100 Ω

Condensateurs :

- C1 à C7 = 100 nF

Semi-conducteurs :

- D1 = diode zener 18 V/500 mW
- D2 à D5 = LED 3 mm faible courant verte
- D6 à D8, D11 = LED 3 mm faible courant ambre ou jaune
- D9, D10, D12, D13 = LED 3 mm faible courant rouge
- T1 à T4 = BC547
- IC1 = TS2950CT-3.3
- IC2 = MC9S08QG8CPBE (PDIP 16 broches; service d'échantillons de Freescale)
- IC3 = MMA7260Q capteur d'accélération monté sur carte gigogne (gratuit, cf. platine ci-après)

Divers :

- K1 = embase à 2 contacts pour connexion de la pile
- K2 = embase à 2 rangées de 3 contacts
- K3 = connecteur à 1 rangée de 9 contacts pour enfilage de K3
- K4 = embase à 1 rangée de 9 contacts pour enfilage de K3
- S1, S2 = bouton-poussoir unipolaire à contact travail (taille 6 mm)
- 4 entretoises de 10 mm
- le set de platines comprend 2 modules accéléromètres MMA7260Q (IC3) gratuits. Code de commande **060297-71**, cf. e-shoppe Elektor logiciel du projet et documents additionnels téléchargeables gratuitement # **060297-11.zip (www.elektor.fr)**

gramme à l'aide de points d'arrêt, de points d'attente et d'un tampon pour suivre à la trace l'exécution. Toutes ces opérations font usage du périphérique de débogage du microcontrôleur cible. Voici ce que vous verrez à l'écran :

- une fenêtre qui affiche le code ;
- une fenêtre d'assemblage qui montre ce que le compilateur a fait de votre code source ;
- une fenêtre des registres dans laquelle vous pourrez vérifier le contenu des registres ;
- une fenêtre de mémoire pour surveiller n'importe quelle adresse ou forcer les bits à une autre valeur ;
- d'autres fenêtres de données, de procédures et de commandes.

On peut changer le format des données et la fréquence de leur rafraî-

chissement en les modifiant après un clic droit dans la fenêtre. Vous pouvez enregistrer les modifications en suivant File → Save Configuration. Consacrez quelques instants à rechercher et jouer avec les boutons Lancer/Continuer, Pas à pas, Passer, Sortir, Un pas d'assembleur, Pause et Mise à zéro de la cible. Une aide efficace pour se lancer et retrouver son chemin dans l'environnement de débogage, vous la trouverez dans les deux documents suivants : *AN3335 - Introduction to HCS08 Background Debug Mode et AN2616 - Getting Started with HCS08 et CodeWarrior Using C.*

On peut brancher un oscilloscope dans le champ de broches à côté du microcontrôleur pour observer les sorties pendant que le débogueur fonctionne en temps réel.

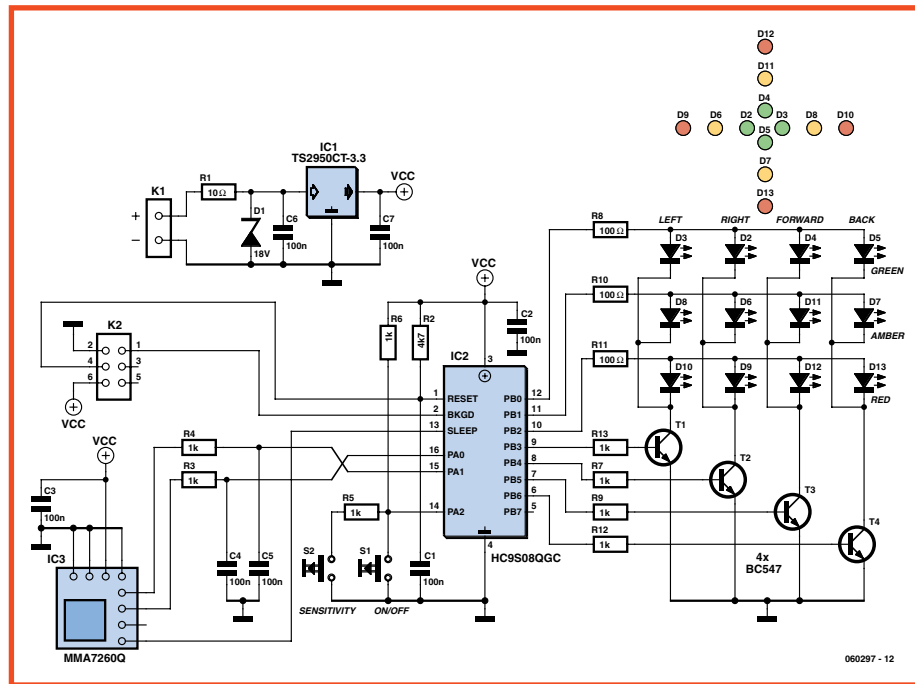


Figure 3. Schéma du circuit des deux accéléromètres à 2 axes.

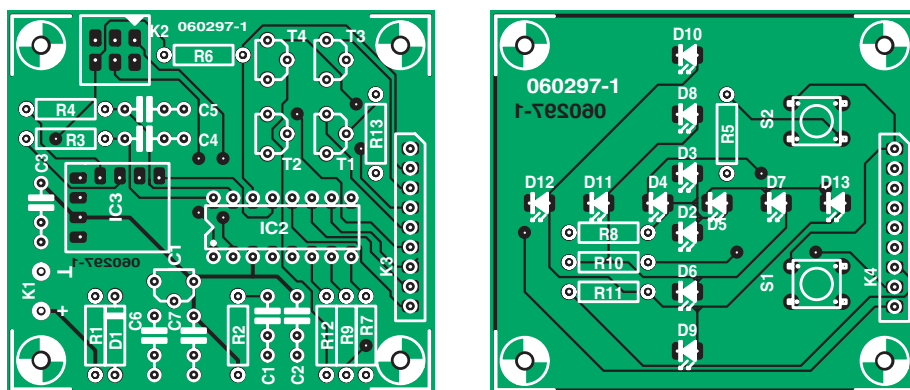


Figure 4. Tracé des pistes et disposition des composants des deux platines à meubler avant leur assemblage à impériale.

Votre premier projet : un accéléromètre

La figure 3 représente le schéma de l'accéléromètre animé par le MC9S08. Le microcontrôleur IC2 est un PDIP à 16 broches dont vous pouvez obtenir en ligne des échantillons gratuits, comme nous l'avons dit le mois dernier. Ce micro vous est livré vierge et il faudra manifestement le programmer avec le micrologiciel. C'est la mission de SpYder.

Le capteur de l'accéléromètre du type MMA7260Q est monté, spécialement pour ce projet Elektor, sur une platine de soutien de 12x12 mm. Vous en recevrez gratuitement deux avec les platines pour le projet (cf. liste des composants). Pour le reste, il faudra un peu plus que les 12 LED positionnées dans les quatre directions, allumées

Modifiez cet accéléromètre !

Il est extrêmement facile de modifier l'accéléromètre pour qu'il affiche en permanence le nombre de g maximum atteint sur chacun des axes. Commencez par ouvrir le projet Accéléromètre dans l'environnement CodeWarrior (5.1). Le fichier-source, main.c, se trouve à l'endroit où se trouve le code de l'application.

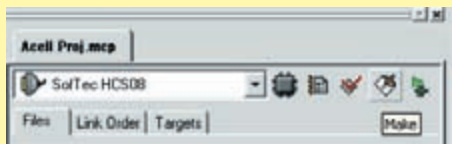
Les modifications à effectuer se situent au niveau de la fonction Byte2LED. Le code ci-après remplace le code de la version précédente :

Le code utilise le paramètre exporté « Val » pour échantillonner la gamme du spectre G et déterminer si elle est neutre, positive (G, A ou R) ou négative (G, A ou R). Quand il a trouvé la gamme correcte, il efface l'axe opposé, mais laisse allumée la LED maximale. Il cherche ensuite si Val est un nouveau maximum et le met à jour si nécessaire (toujours si rouge). La routine affiche alors la grandeur par le maximum sur cet axe.

N'oubliez pas d'ajouter les définitions du nouveau « LED Bitmap » en tête du fichier et « LEDMax bitmap » à l'ensemble des variables dans main.c.

Vous pouvez alors enregistrer le projet sous un autre nom, si vous le désirez, en suivant File→Save A Copy As... Dès que le projet est créé, il est enregistré et donc la version précédente du projet est perdue. Vous pouvez aussi enregistrer les fichiers sources individuellement par File→Save As... quand vous êtes dans la fenêtre du code source.

Après toute modification, vous devez repasser le projet par « compile/make ». Si aucune erreur n'a été introduite (croisons les doigts) vous pouvez alors déboguer le projet en utilisant le bouton à droite (flèche verte avec insecte).



Assurez-vous que USBSpYder08 est sélectionné comme « Hardware Model » et MC9S08QG8 comme « Device ». CodeWarrior va alors effacer et reprogrammer l'appareil avec le nouveau projet.

Si vous allez à la fonction Byte2LED et que vous placez un point d'arrêt à la première instruction « if » d'un clic à droite, CodeWarrior va arrêter le programme quand il arrivera à ce point. Dans la fenêtre de données, vous pourrez lire la valeur de Val et en avançant pas à pas dans la fonction, vous observerez le déroulement du programme dans la gamme correcte. Libre à vous d'ajouter d'autres points d'arrêt et de les supprimer d'un clic droit de la souris à d'autres endroits du programme.

```
void Byte2LED(char Val, char Dir, char SenMode){

    /* Positive Green */
    if ((Val < PosA[SenMode])&&(Val >= PosG[SenMode])) {
    //if (PosA[SenMode] > Val >= PosG[SenMode]) {
        LEDMag[NegYLED + Dir] = (LEDOFF | LEDMax[NegYLED + Dir]);
        if(LEDMag[PosYLED + Dir] > LEDMax[PosYLED + Dir]) {
            LEDMax[PosYLED + Dir] = LEDG;
        }
        LEDMag[PosYLED + Dir] = (LEDG | LEDMax[PosYLED + Dir]);
    }

    /* Positive Amber */
    if ((Val < PosR[SenMode])&&(Val >= PosA[SenMode])) {
    //if (PosR[SenMode] > Val >= PosA[SenMode]) {
        LEDMag[NegYLED + Dir] = (LEDOFF | LEDMax[NegYLED + Dir]);
        if (LEDMag[PosYLED + Dir] > LEDMax[PosYLED + Dir]) {
            LEDMax[PosYLED + Dir] = LEDA;
        }
        LEDMag[PosYLED + Dir] = (LEDGA | LEDMax[PosYLED + Dir]);
    }

    /* Positive Red */
    if (Val >= PosR[SenMode]) {
        LEDMag[NegYLED + Dir] = (LEDOFF | LEDMax[NegYLED + Dir]);
        LEDMag[PosYLED + Dir] = LEDGAR;
        LEDMax[PosYLED + Dir] = LEDR;
    }

    /* Negative Green */
    if ((Val < NegG[SenMode])&&(Val >= NegA[SenMode])) {
    //if (NegG[SenMode] > Val >= NegA[SenMode]) {
        LEDMag[PosYLED + Dir] = (LEDOFF | LEDMax[PosYLED + Dir]);
        if (LEDMag[NegYLED + Dir] > LEDMax[NegYLED + Dir]) {
            LEDMax[NegYLED + Dir] = LEDG;
        }
        LEDMag[NegYLED + Dir] = (LEDG | LEDMax[NegYLED + Dir]);
    }

    /* Negative Amber */
    if ((Val < NegA[SenMode])&&(Val >= NegR[SenMode])) {
    //if (NegA[SenMode] > Val >= NegR[SenMode]) {
        LEDMag[PosYLED + Dir] = (LEDOFF | LEDMax[PosYLED + Dir]);
        if (LEDMag[NegYLED + Dir] > LEDMax[NegYLED + Dir]) {
            LEDMax[NegYLED + Dir] = LEDA;
        }
        LEDMag[NegYLED + Dir] = (LEDGA | LEDMax[NegYLED + Dir]);
    }

    /* Negative Red */
    if (NegR[SenMode] >= Val) {
        LEDMag[PosYLED + Dir] = (LEDOFF | LEDMax[PosYLED + Dir]);
        LEDMag[NegYLED + Dir] = LEDGAR;
        LEDMax[NegYLED + Dir] = LEDR;
    }

    /* Neutral */
    if ((Val < PosG[SenMode])&&(Val > NegG[SenMode])) {
    //if (PosG[SenMode] > Val > NegG[SenMode]) {
        LEDMag[PosYLED + Dir] = (LEDOFF | LEDMax[PosYLED + Dir]);
        LEDMag[NegYLED + Dir] = (LEDOFF | LEDMax[NegYLED + Dir]);
    }
}
```

et éteintes, en forme de constellation matricielle, par les lignes PB0 à PB6 du microcontrôleur MC9S08.

Le circuit est alimenté sur piles, un groupe qui doit fournir au moins 4,5 V (par ex. trois AA ou AAA en série) et consomme environ 25 mA quand trois LED d'une rangée sont allumées. Un régulateur TS2950CT-3.3 embarqué abaisse à 3,3 V la tension pour la circuiterie. Il y a aussi deux boutons poussoirs : S1 commande la mise en/hors service et S2 choisit la sensibilité. Remarquez la méthode futée et économique d'interroger les interrupteurs.

Le circuit se répartit sur deux platines, étagées à l'aide d'entretoises de 10 mm en PCB. Même si le tracé des pistes est dûment reproduit aux figures 4a et 4b, la gravure et le perçage à domicile sont largement concurrencés par le prix minime des cartes toutes faites, d'autant que les deux capteurs y sont joints gracieusement. Qui s'en plaindrait ?

De prime abord, nous avions pensé à une exécution totalement montée en surface, mais à la réflexion, nous avons opté pour un microcontrôleur dans un bon vieux boîtier PDIP à 16 broches et des composants traditionnels soudés sur l'autre face, montés sur deux platines à trous métallisés de 55x55 mm. La carte porteuse du capteur s'installe dans l'aire « IC3 ». Le connecteur K2 permet une liaison à 6 voies vers le BDM de SpYder.

Le tableau de bord avec poussoirs et LED est installé au-dessus de la carte contrôleur à l'aide d'entretoises en PCB. La connexion électrique entre eux s'opère par une embase SIL. Vous avez certainement remarqué la photo du prototype terminé et opérationnel, tandis que les platines séparées sont visibles à la figure 5.



Programmation

Croyez-le ou non, nous recevons des réclamations de certains de nos lecteurs selon lesquelles les projets d'Elektor ne marchent pas, malgré l'achat de « puces d'origine en provenance de la firme Xyz, insérées à la bonne place sur une carte approuvée par un expert qui totalise 40 ans d'expérience en électronique ».

C'est que les microcontrôleurs vierges doivent passer en programmation avant de pouvoir servir à quelque chose.

Téléchargez le fichier archivé 060297-11.zip du site d'Elektor et décompressez-le dans un répertoire approprié que vous pourriez nommer Accéléromètre, par exemple.

Reliez ensemble SpYder, le matériel de l'accéléromètre et CodeWarrior et faites démarrer le tout. Dans IDE de CodeWarrior, suivez le chemin File → Open Project, puis naviguez jusqu'au répertoire que vous venez de créer pour y trouver « Accel Proj.mcp ». Chargez ce projet, sélectionnez le bon appareil pour programmer (MC9S08QG), compilez, créez le code objet et programmez le microcontrôleur par la liaison BDM. C'est un excellent exercice pour se familiariser avec tous les aspects de l'environnement des microcontrôleurs Freescale, surtout qu'il n'y a aucun risque de bousiller quoi que ce soit, puisque vous pourrez toujours tout effacer et recommencer. Dites-nous comment ça a marché, vous trouverez un sujet sur le forum en ligne d'Elektor pour communiquer avec d'autres adeptes et avec les concepteurs.

Utilisation pratique

Maintenez le bouton de mise en marche appuyé jusqu'à ce que les LED s'allument. Un défilement lumineux en séquence vous convaincra que le microcontrôleur est éveillé, qu'il exécute le micrologiciel et que toutes les LED fonctionnent. L'instrument est prêt à l'emploi. Basculez vivement l'appareil à la main dans différentes directions pour vérifier que l'indicateur de g réagit sous tous les axes.

On peut actionner le bouton de sensibilité pour choisir entre trois gammes :

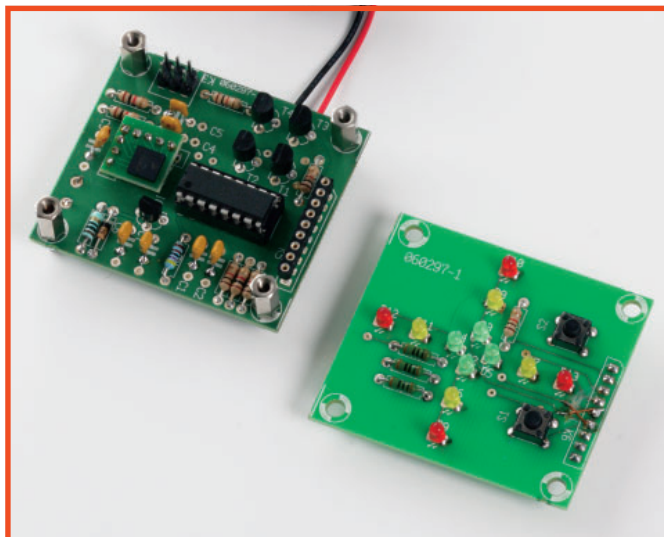


Figure 5. Les platines montées avant leur assemblage par entretoises en PCB dans les quatre coins.

| Sensibilité | min | moy | max [g] |
|-------------|------|------|---------|
| Vert | 0,45 | 0,29 | 0,15 g |
| Jaune | 0,80 | 0,53 | 0,27 g |
| Rouge | 1,15 | 0,97 | 0,39 g |

On arrête l'instrument en gardant le bouton de marche enfoncé jusqu'à ce que les LED commencent à former un curseur en forme de croix.

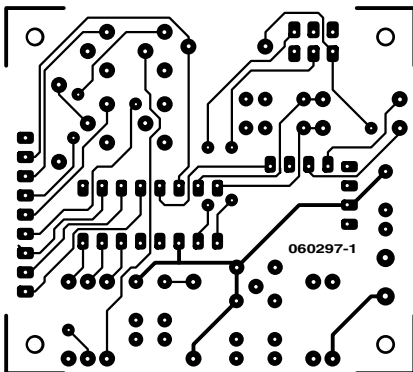
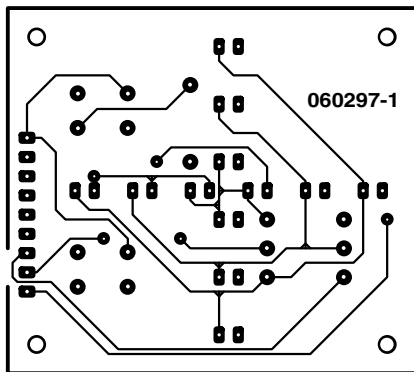
Vous pouvez vous procurer une description détaillée du fonctionnement du micrologiciel de l'accéléromètre dans un document séparé inclus dans le téléchargement gratuit de cet article. Son contenu est destiné aux programmeurs chevronnés.

Conclusion

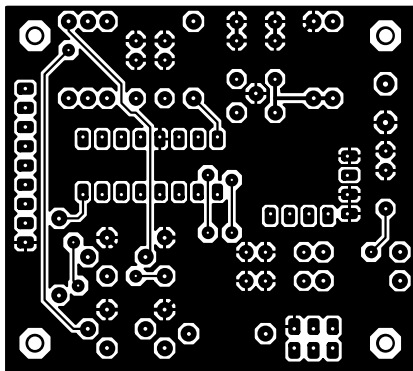
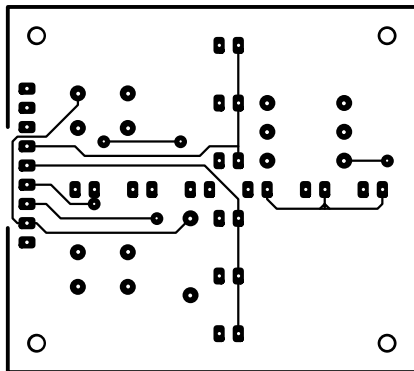
Le Kit de découverte de SpYder est un outil étonnamment souple qui répond à tous les besoins d'un développement à budget serré. Il s'occupe directement des puces S08 PDIP à 8 broches de Freescale, mais on peut étendre le nombre de broches par l'utilisation de l'option Chip « target ». Code Warrior est un outil très puissant, quelque peu intimidant au premier abord, mais facile à dominer dès qu'on a pris la peine de découvrir tous les boutons nécessaires. Vous avez en outre la certitude d'apprendre les autres capacités qu'il recèle en vous familiarisant avec l'outil, Freescale vous procure, en ligne, l'aide de documents et un d'entraînement adéquats.

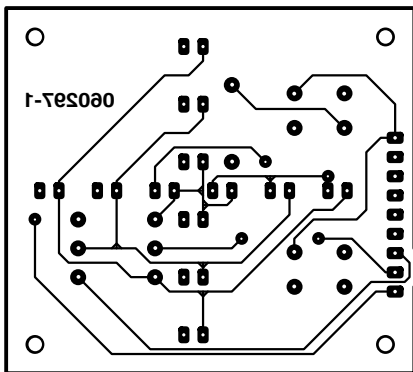
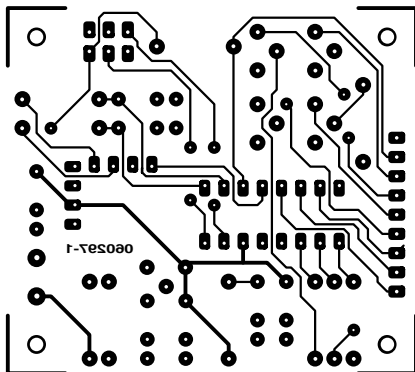
D'autres projets faisant appel à SpYder sont en préparation chez Elektor.

(060297-1)

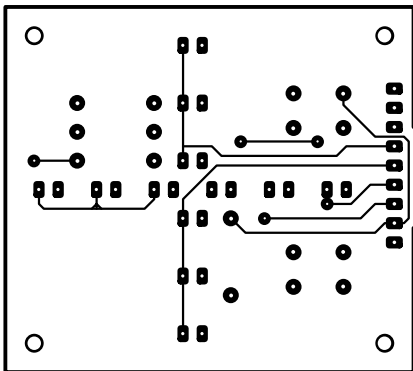
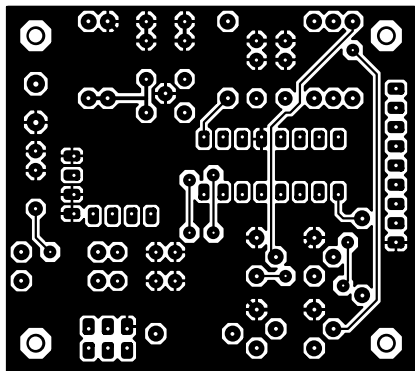


non reflected





reflected





±1.5g - 6g Three Axis Low-g Micromachined Accelerometer

The MMA7260QT low cost capacitive micromachined accelerometer features signal conditioning, a 1-pole low pass filter, temperature compensation and g-Select which allows for the selection among 4 sensitivities. Zero-g offset full scale span and filter cut-off are factory set and require no external devices. Includes a Sleep Mode that makes it ideal for handheld battery powered electronics.

Features

- Selectable Sensitivity (1.5g/2g/4g/6g)
- Low Current Consumption: 500 μ A
- Sleep Mode: 3 μ A
- Low Voltage Operation: 2.2 V – 3.6 V
- 6mm x 6mm x 1.45mm QFN
- High Sensitivity (800 mV/g @ 1.5g)
- Fast Turn On Time
- Integral Signal Conditioning with Low Pass Filter
- Robust Design, High Shocks Survivability
- Pb-Free Terminations
- Environmentally Preferred Package
- Low Cost

Typical Applications

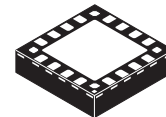
- HDD MP3 Player: Freefall Detection
- Laptop PC: Freefall Detection, Anti-Theft
- Cell Phone: Image Stability, Text Scroll, Motion Dialing, E-Compass
- Pedometer: Motion Sensing
- PDA: Text Scroll
- Navigation and Dead Reckoning: E-Compass Tilt Compensation
- Gaming: Tilt and Motion Sensing, Event Recorder
- Robotics: Motion Sensing

| ORDERING INFORMATION | | | |
|----------------------|-------------------|-----------------|---------------------|
| Device Name | Temperature Range | Package Drawing | Package |
| MMA7260QT | -20 to +85°C | 1622-02 | QFN-16, Tray |
| MMA7260QR2 | -20 to +85°C | 1622-02 | QFN-16, Tape & Reel |

MMA7260QT

**MMA7260QT: XYZ AXIS
 ACCELEROMETER**
 ± 1.5g/2g/4g/6g

Bottom View



**16 LEAD
 QFN
 CASE 1622-02**

Top View

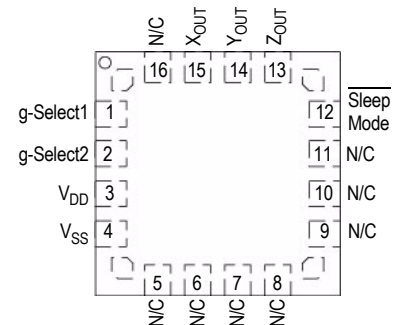


Figure 1. Pin Connections

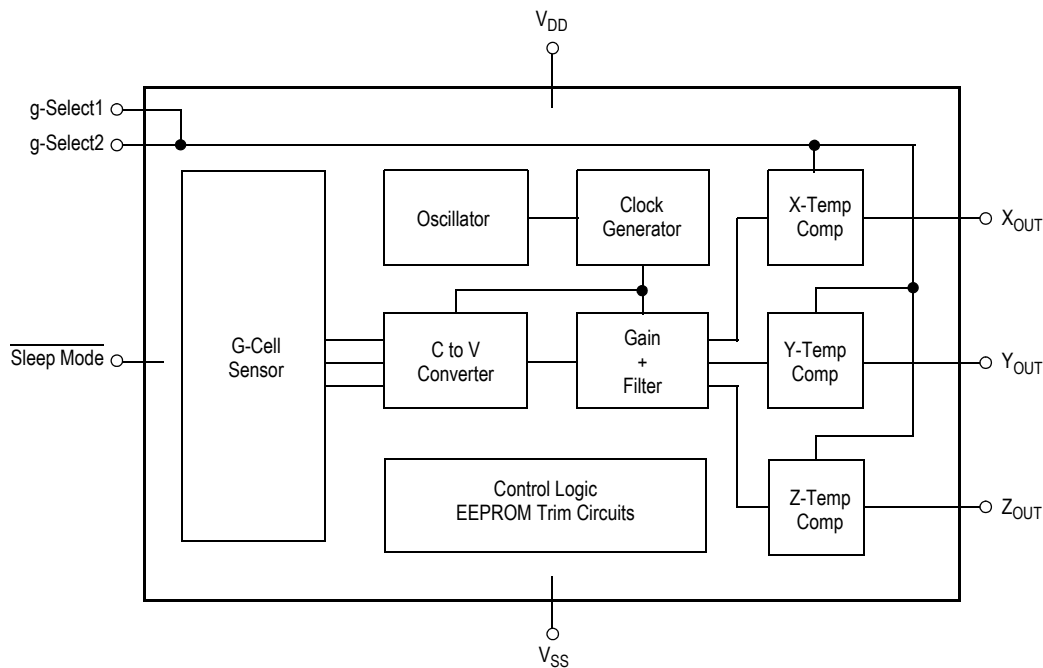


Figure 2. Simplified Accelerometer Functional Block Diagram

Table 1. Maximum Ratings

(Maximum ratings are the limits to which the device can be exposed without causing permanent damage.)

| Rating | Symbol | Value | Unit |
|---------------------------------|------------|--------------|------|
| Maximum Acceleration (all axis) | g_{max} | ± 2000 | g |
| Supply Voltage | V_{DD} | -0.3 to +3.6 | V |
| Drop Test ⁽¹⁾ | D_{drop} | 1.8 | m |
| Storage Temperature Range | T_{stg} | -40 to +125 | °C |

1. Dropped onto concrete surface from any axis.

ELECTRO STATIC DISCHARGE (ESD)

WARNING: This device is sensitive to electrostatic discharge.

Although the Freescale accelerometer contains internal 2000 V ESD protection circuitry, extra precaution must be taken by the user to protect the chip from ESD. A charge of over 2000 volts can accumulate on the human body or associated test equipment. A charge of this magnitude can

alter the performance or cause failure of the chip. When handling the accelerometer, proper ESD precautions should be followed to avoid exposing the device to discharges which may be detrimental to its performance.

Table 2. Operating CharacteristicsUnless otherwise noted: $-20^{\circ}\text{C} \leq T_A \leq 85^{\circ}\text{C}$, $2.2\text{ V} \leq V_{\text{DD}} \leq 3.6\text{ V}$, Acceleration = 0g, Loaded output⁽¹⁾

| Characteristic | Symbol | Min | Typ | Max | Unit |
|---|--------------------------|----------------------|------------|----------------------|--------------------------------|
| Operating Range ⁽²⁾ | | | | | |
| Supply Voltage ⁽³⁾ | V_{DD} | 2.2 | 3.3 | 3.6 | V |
| Supply Current | I_{DD} | — | 500 | 800 | μA |
| Supply Current at Sleep Mode ⁽⁴⁾ | I_{DD} | — | 3.0 | 10 | μA |
| Operating Temperature Range | T_A | -20 | — | +85 | $^{\circ}\text{C}$ |
| Acceleration Range, X-Axis, Y-Axis, Z-Axis | | | | | |
| g-Select1 & 2: 00 | g_{FS} | — | ± 1.5 | — | g |
| g-Select1 & 2: 10 | g_{FS} | — | ± 2.0 | — | g |
| g-Select1 & 2: 01 | g_{FS} | — | ± 4.0 | — | g |
| g-Select1 & 2: 11 | g_{FS} | — | ± 6.0 | — | g |
| Output Signal | | | | | |
| Zero g ($T_A = 25^{\circ}\text{C}$, $V_{\text{DD}} = 3.3\text{ V}$) ⁽⁵⁾ | V_{OFF} | 1.485 | 1.65 | 1.815 | V |
| Zero g ⁽⁴⁾ | V_{OFF}, T_A | — | ± 2.0 | — | $\text{mg}/^{\circ}\text{C}$ |
| Sensitivity ($T_A = 25^{\circ}\text{C}$, $V_{\text{DD}} = 3.3\text{ V}$) | | | | | |
| 1.5g | $S_{1.5\text{g}}$ | 740 | 800 | 860 | mV/g |
| 2g | $S_{2\text{g}}$ | 555 | 600 | 645 | mV/g |
| 4g | $S_{4\text{g}}$ | 277.5 | 300 | 322.5 | mV/g |
| 6g | $S_{6\text{g}}$ | 185 | 200 | 215 | mV/g |
| Sensitivity ⁽⁴⁾ | S, T_A | — | ± 0.03 | — | $\%/^{\circ}\text{C}$ |
| Bandwidth Response | | | | | |
| XY | $f_{-3\text{dB}}$ | — | 350 | — | Hz |
| Z | $f_{-3\text{dB}}$ | — | 150 | — | Hz |
| Noise | | | | | |
| RMS (0.1 Hz – 1 kHz) ⁽⁴⁾ | n_{RMS} | — | 4.7 | — | mVrms |
| Power Spectral Density RMS (0.1 Hz – 1 kHz) ⁽⁴⁾ | n_{PSD} | — | 350 | — | $\mu\text{g}/\sqrt{\text{Hz}}$ |
| Control Timing | | | | | |
| Power-Up Response Time ⁽⁶⁾ | t_{RESPONSE} | — | 1.0 | 2.0 | ms |
| Enable Response Time ⁽⁷⁾ | t_{ENABLE} | — | 0.5 | 2.0 | ms |
| Sensing Element Resonant Frequency | | | | | |
| XY | f_{GCELL} | — | 6.0 | — | kHz |
| Z | f_{GCELL} | — | 3.4 | — | kHz |
| Internal Sampling Frequency | f_{CLK} | — | 11 | — | kHz |
| Output Stage Performance | | | | | |
| Full-Scale Output Range ($I_{\text{OUT}} = 30\ \mu\text{A}$) | V_{FSO} | $V_{\text{SS}}+0.25$ | — | $V_{\text{DD}}-0.25$ | V |
| Nonlinearity, X_{OUT} , Y_{OUT} , Z_{OUT} | NL_{OUT} | -1.0 | — | +1.0 | %FSO |
| Cross-Axis Sensitivity ⁽⁸⁾ | $V_{\text{XY, XZ, YZ}}$ | — | — | 5.0 | % |

1. For a loaded output, the measurements are observed after an RC filter consisting of a 1.0 k Ω resistor and a 0.1 μF capacitor on $V_{\text{DD}}\text{-GND}$.
2. These limits define the range of operation for which the part will meet specification.
3. Within the supply range of 2.2 and 3.6 V, the device operates as a fully calibrated linear accelerometer. Beyond these supply limits the device may operate as a linear device but is not guaranteed to be in calibration.
4. This value is measured with g-Select in 1.5g mode.
5. The device can measure both + and - acceleration. With no input acceleration the output is at midsupply. For positive acceleration the output will increase above $V_{\text{DD}}/2$. For negative acceleration, the output will decrease below $V_{\text{DD}}/2$.
6. The response time between 10% of full scale Vdd input voltage and 90% of the final operating output voltage.
7. The response time between 10% of full scale Sleep Mode input voltage and 90% of the final operating output voltage.
8. A measure of the device's ability to reject an acceleration applied 90° from the true axis of sensitivity.

PRINCIPLE OF OPERATION

The Freescale accelerometer is a surface-micromachined integrated-circuit accelerometer.

The device consists of two surface micromachined capacitive sensing cells (g-cell) and a signal conditioning ASIC contained in a single integrated circuit package. The sensing elements are sealed hermetically at the wafer level using a bulk micromachined cap wafer.

The g-cell is a mechanical structure formed from semiconductor materials (polysilicon) using semiconductor processes (masking and etching). It can be modeled as a set of beams attached to a movable central mass that move between fixed beams. The movable beams can be deflected from their rest position by subjecting the system to an acceleration (Figure 3).

As the beams attached to the central mass move, the distance from them to the fixed beams on one side will increase by the same amount that the distance to the fixed beams on the other side decreases. The change in distance is a measure of acceleration.

The g-cell beams form two back-to-back capacitors (Figure 3). As the center beam moves with acceleration, the distance between the beams changes and each capacitor's value will change, ($C = A\epsilon/D$). Where A is the area of the beam, ϵ is the dielectric constant, and D is the distance between the beams.

The ASIC uses switched capacitor techniques to measure the g-cell capacitors and extract the acceleration data from the difference between the two capacitors. The ASIC also signal conditions and filters (switched capacitor) the signal, providing a high level output voltage that is ratiometric and proportional to acceleration.

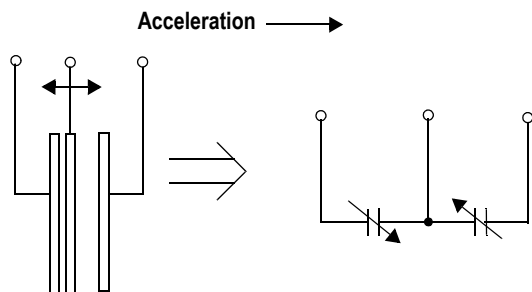


Figure 3. Simplified Transducer Physical Model

SPECIAL FEATURES

g-Select

The g-Select feature allows for the selection among 4 sensitivities present in the device. Depending on the logic input placed on pins 1 and 2, the device internal gain will be changed allowing it to function with a 1.5g, 2g, 4g, or 6g sensitivity (Table 3). This feature is ideal when a product has applications requiring different sensitivities for optimum performance. The sensitivity can be changed at anytime during the operation of the product. The g-Select1 and g-Select2 pins can be left unconnected for applications requiring only a 1.5g sensitivity as the device has an internal pull-down to keep it at that sensitivity (800mV/g).

Table 3. g-Select pin Descriptions

| g-Select2 | g-Select1 | g-Range | Sensitivity |
|-----------|-----------|---------|-------------|
| 0 | 0 | 1.5g | 800 mV/g |
| 0 | 1 | 2g | 600 mV/g |
| 1 | 0 | 4g | 300 mV/g |
| 1 | 1 | 6g | 200 mV/g |

Sleep Mode

The 3 axis accelerometer provides a Sleep Mode that is ideal for battery operated products. When Sleep Mode is active, the device outputs are turned off, providing significant reduction of operating current. A low input signal on pin 12 (Sleep Mode) will place the device in this mode and reduce the current to 3 μ A typ. For lower power consumption, it is recommended to set g-Select1 and g-Select2 to 1.5g mode. By placing a high input signal on pin 12, the device will resume to normal mode of operation.

Filtering

The 3 axis accelerometer contains onboard single-pole switched capacitor filters. Because the filter is realized using switched capacitor techniques, there is no requirement for external passive components (resistors and capacitors) to set the cut-off frequency.

Ratiometricity

Ratiometricity simply means the output offset voltage and sensitivity will scale linearly with applied supply voltage. That is, as supply voltage is increased, the sensitivity and offset increase linearly; as supply voltage decreases, offset and sensitivity decrease linearly. This is a key feature when interfacing to a microcontroller or an A/D converter because it provides system level cancellation of supply induced errors in the analog to digital conversion process.

BASIC CONNECTIONS

Pin Descriptions

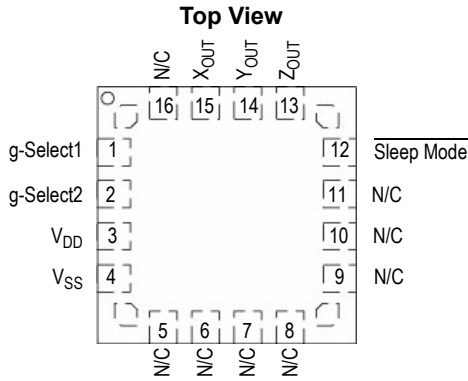


Figure 4. Pinout Description

Table 4. Pin Descriptions

| Pin No. | Pin Name | Description |
|---------|------------------|--|
| 1 | g-Select1 | Logic input pin to select g level. |
| 2 | g-Select2 | Logic input pin to select g level. |
| 3 | V _{DD} | Power Supply Input |
| 4 | V _{SS} | Power Supply Ground |
| 5 - 7 | N/C | No internal connection. Leave unconnected. |
| 8 - 11 | N/C | Unused for factory trim. Leave unconnected. |
| 12 | Sleep Mode | Logic input pin to enable product or Sleep Mode. |
| 13 | Z _{OUT} | Z direction output voltage. |
| 14 | Y _{OUT} | Y direction output voltage. |
| 15 | X _{OUT} | X direction output voltage. |
| 16 | N/C | No internal connection. Leave unconnected. |

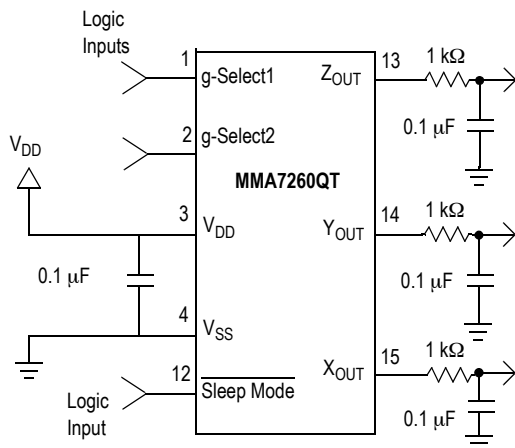


Figure 5. Accelerometer with Recommended Connection Diagram

PCB Layout

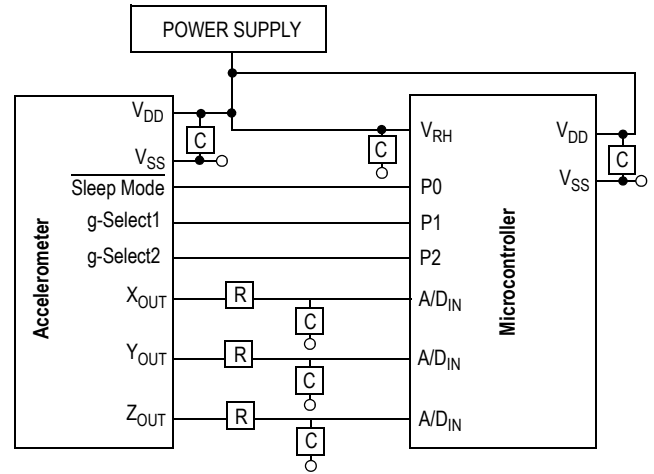
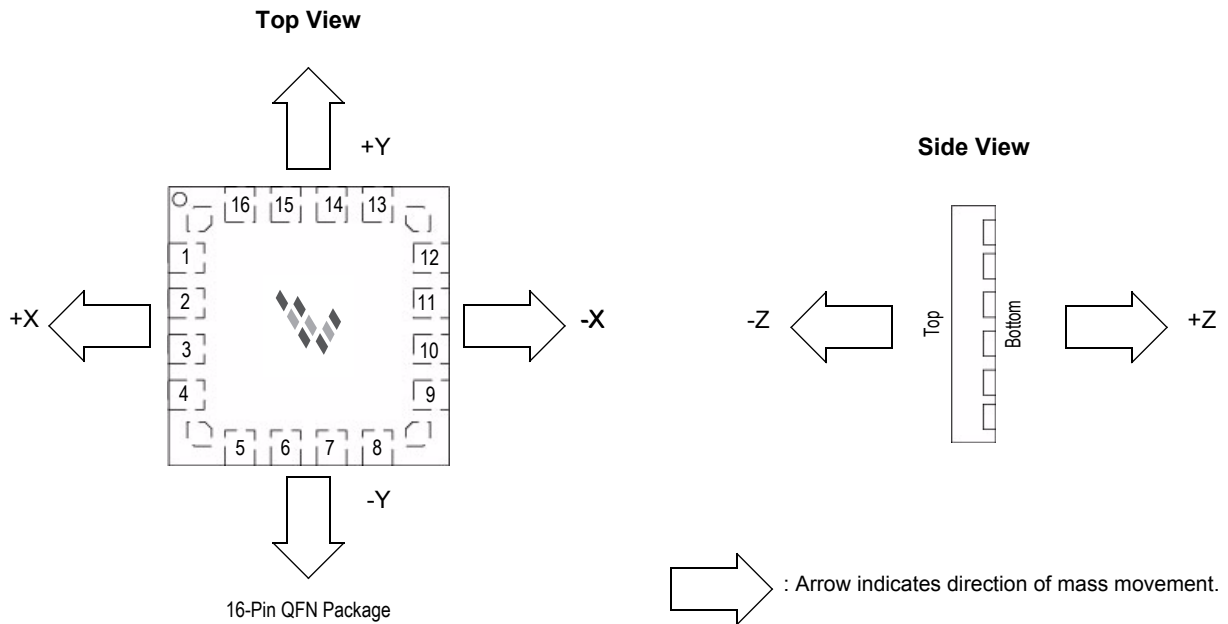


Figure 6. Recommended PCB Layout for Interfacing Accelerometer to Microcontroller

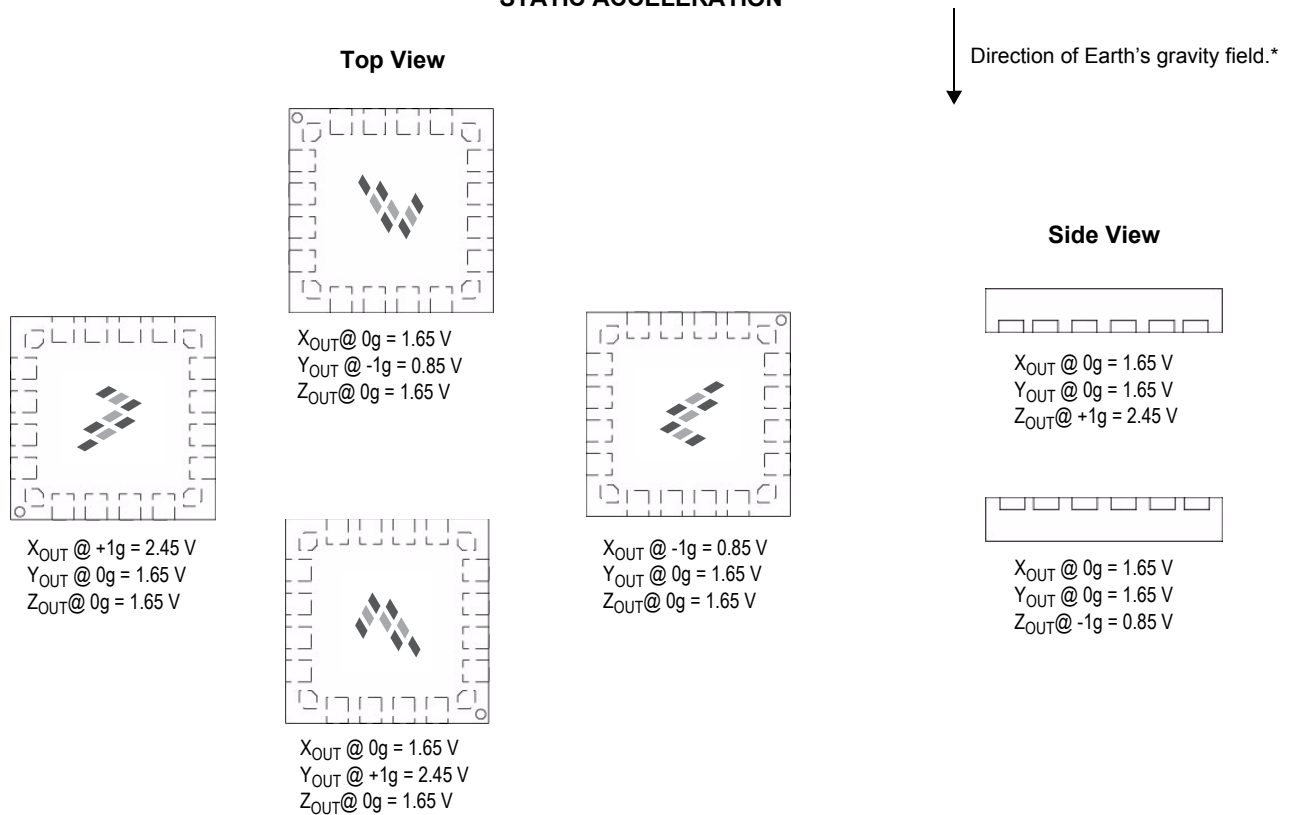
NOTES:

1. Use 0.1 μF capacitor on V_{DD} to decouple the power source. Do not exceed capacitor values of 2.2 or 3.3 μF.
2. Physical coupling distance of the accelerometer to the microcontroller should be minimal.
3. The flag underneath the package is internally connected to ground. It is not recommended for the flag to be soldered down.
4. Place a ground plane beneath the accelerometer to reduce noise, the ground plane should be attached to all of the open ended terminals shown in Figure 6.
5. Use an RC filter with 1.0 kΩ and 0.1 μF on the outputs of the accelerometer to minimize clock noise (from the switched capacitor filter circuit).
6. PCB layout of power and ground should not couple power supply noise.
7. Accelerometer and microcontroller should not be a high current path.
8. A/D sampling rate and any external power supply switching frequency should be selected such that they do not interfere with the internal accelerometer sampling frequency (11 kHz for the sampling frequency). This will prevent aliasing errors.
9. PCB layout should not run traces or vias under the QFN part. This could lead to ground shorting to the accelerometer flag.

DYNAMIC ACCELERATION



STATIC ACCELERATION



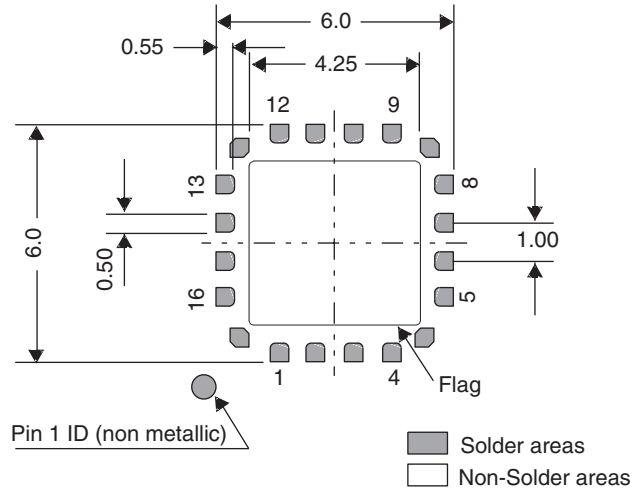
* When positioned as shown, the Earth's gravity will result in a positive 1g output.

MINIMUM RECOMMENDED FOOTPRINT FOR SURFACE MOUNTED APPLICATIONS

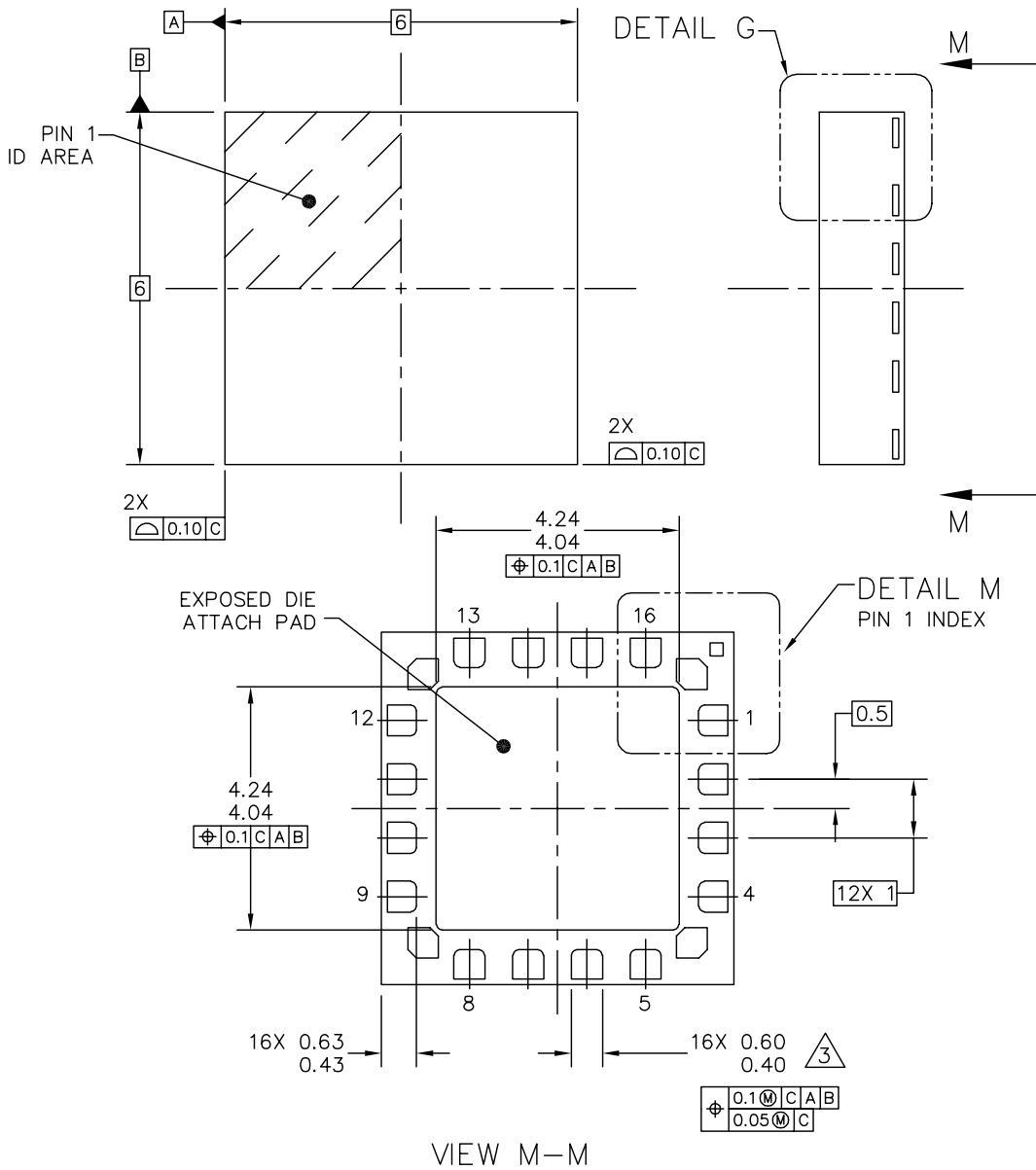
Surface mount board layout is a critical portion of the total design. The footprint for the surface mount packages must be the correct size to ensure proper solder connection interface between the board and the package.

With the correct footprint, the packages will self-align when subjected to a solder reflow process. It is always recommended to design boards with a solder mask layer to avoid bridging and shorting between solder pads.

The flag underneath the package is internally connected to ground. It is not recommended for the flag to be soldered down.



PACKAGE DIMENSIONS

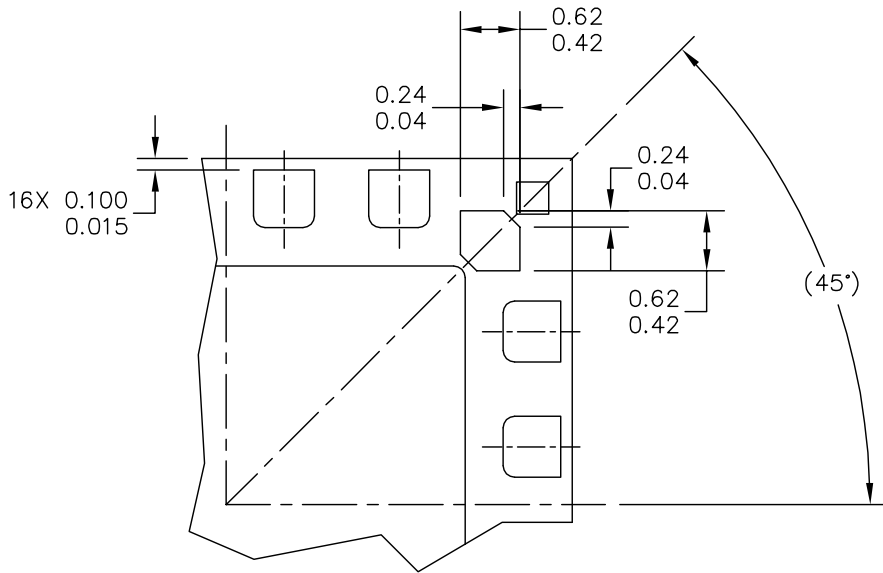


| | | | |
|---|---------------------------|----------------------------|--|
| © FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED. | MECHANICAL OUTLINE | PRINT VERSION NOT TO SCALE | |
| TITLE: QUAD FLAT NON-LEADED PACKAGE (QFN) FOR SENSORS 16 TERMINAL, 1.0 PITCH (6 X 6 X 1.45) | DOCUMENT NO: 98ASA10651D | REV: B | |
| | CASE NUMBER: 1622-02 | 27 SEP 2006 | |
| | STANDARD: NON-JEDEC | | |

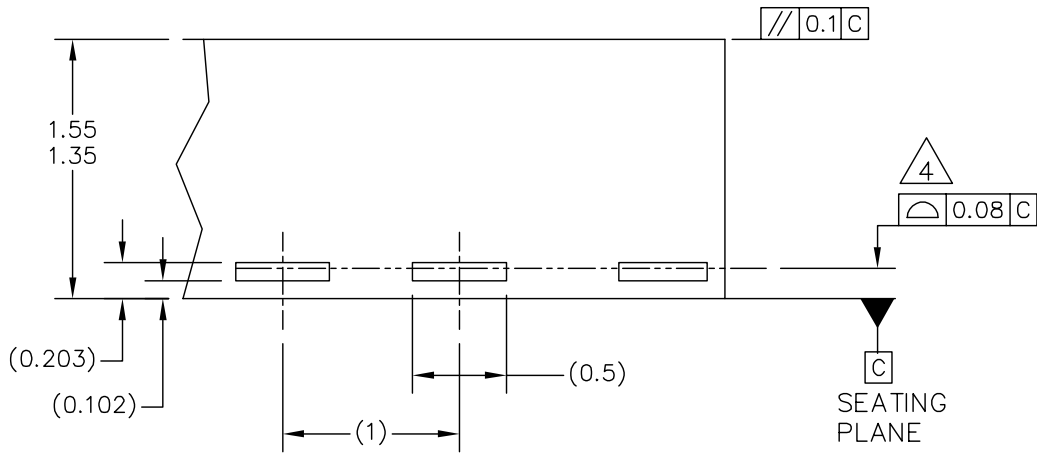
PAGE 1 OF 3

**CASE 1622-02
ISSUE B
16-LEAD QFN**

PACKAGE DIMENSIONS



DETAIL M



DETAIL G
VIEW ROTATED 90° CW

| | | | |
|---|---------------------------|----------------------------|--|
| © FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED. | MECHANICAL OUTLINE | PRINT VERSION NOT TO SCALE | |
| TITLE: QUAD FLAT NON-LEADED PACKAGE (QFN) FOR SENSORS 16 TERMINAL, 1.0 PITCH (6 X 6 X 1.45) | DOCUMENT NO: 98ASA10651D | REV: B | |
| | CASE NUMBER: 1622-02 | 27 SEP 2006 | |
| | STANDARD: NON-JEDEC | | |

PACKAGE DIMENSIONS

NOTES:

1. ALL DIMENSIONS ARE IN MILLIMETERS.
2. INTERPRET DIMENSIONS AND TOLERANCES PER ASME Y14.5M-1994.
3. THIS DIMENSION APPLIES TO METALLIZED TERMINAL AND IS MEASURED BETWEEN 0.25MM AND 0.30MM FROM TERMINAL TIP.
4. COPLANARITY APPLIES TO THE EXPOSED HEAT SLUG, TERMINALS AND CORNER PADS.
5. RADIUS ON TERMINAL IS OPTIONAL.
6. MINIMUM METAL GAP SHOULD BE 0.2MM EXCEPT GAP BETWEEN CORNER PADS AND THE EXPOSED HEAT SLUG.

| | | | |
|---|---------------------------|----------------------------|--|
| © FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED. | MECHANICAL OUTLINE | PRINT VERSION NOT TO SCALE | |
| TITLE: QUAD FLAT NON-LEADED PACKAGE (QFN) FOR SENSORS 16 TERMINAL, 1.0 PITCH (6 X 6 X 1.45) | DOCUMENT NO: 98ASA10651D | REV: B | |
| | CASE NUMBER: 1622-02 | 27 SEP 2006 | |
| | STANDARD: NON-JEDEC | | |

PAGE 3 OF 3

**CASE 1622-02
 ISSUE B
 16-LEAD QFN**

How to Reach Us:

Home Page:

www.freescale.com

E-mail:

support@freescale.com

USA/Europe or Locations Not Listed:

Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

Europe, Middle East, and Africa:

Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:

Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:

Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:

Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

RoHS-compliant and/or Pb-free versions of Freescale products have the functionality and electrical characteristics of their non-RoHS-compliant and/or non-Pb-free counterparts. For further information, see <http://www.freescale.com> or contact your Freescale sales representative.

For information on Freescale's Environmental Products program, go to <http://www.freescale.com/epp>.

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals", must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.

© Freescale Semiconductor, Inc., 2006. All rights reserved.



A Low Cost 8-bit Microcontroller BDM tool for Freescale's HCS08 and RS08 Microcontrollers from SofTec Microsystems

Introduction

Freescale Semiconductor is the #1 supplier of microcontrollers and yet their tool set and distribution networks were unattainable to the hobbyist market until 2006 as their tool solutions, both hardware and software, were price inhibitive. Since breaking off from Motorola they have placed a strong focus on the mass market and with the recent addition of an e-commerce site, free samples, free of charge compilers and debuggers and low cost hardware tools Freescale's microcontroller families are now accessible to all. The latest tool, the SpYder, manufactured by SofTec Microsystems with a resale price of €30(?) is another strong step into this marketplace.

The roadmap shown in figure 1 shows how Motorola/Freescale's cores have evolved. The HC05 and HC11 8-bit cores were introduced in the 1980's and were widely used by all kinds of developers. In the late 1990's the HC08 (8-bit) and HC12 (16-bit) cores were introduced but were never widely adopted by the mass market. In very early 2000's the HCS12 16-bit core followed by the HCS08 8-bit core were introduced with the key new feature on the Background Debug Module (BDM).

The 8-bit HCS08, and introduced last year, the RS08 microcontrollers contain a single-wire background debug interface, supporting in-circuit programming of on-chip nonvolatile memory and sophisticated non-intrusive debug capabilities. It is this module which enables the development of these low cost, easy to use tools. The BDM connection will also be present on the 32-bit microcontroller 68K/ColdFire™ V1 core products which will be available later this year.

In 2005 freegeeks.net provided the HCS12 microcontroller community with an open source tool named TBDML, and with 1454 downloads in 12 months it was hailed a great success. Now the equivalent tool for their 8-bit, BDM enabled microcontrollers is available in two forms.

- You can chose the OSBDM for the HCS08's which you can find details of on the freescale forums (<http://www.freescale.net/forums>). This self build tool has a BOM of <\$10.
- Or you can buy a ready made SpYder which supports MC9S08QG, MC9S08QD and MC9RS08KA 8-bit microcontrollers to date, and as more microcontrollers are announced this list will grow.

These two tools essentially do the same thing. They interface between your development environment (Windows PC based) and your target microcontroller as shown in Figure 4. The key aim of these tools is to provide a tool which is cheap and easy for hobbyists, students etc. to use.

The 2-G accelerometer from last months Elektor was enabled by Freescale's MC9S08QG8 MCU and the SpYder Discovery Kit. The SpYder Discovery Kit is a new USB-to-BDM development tool for Freescale's MC9S08QG, MC9S08QD and MC9RS08KA 8-bit microcontrollers. For those of you unfamiliar with BDM, it is Freescale's version of ICD, debugWIRE, JTAG etc. used on their recent 8- and 16-bit products.

The BKGD (BackGrounD) pin on these devices provides this single-wire background debug interface to the on chip debug modules. See the Development Tools chapter of any HCS08 or RS08 data sheet for more information about these debug modules and how to use them. While the interface is single wire, typically a 6-pin connector, a BDM port is used to interface with the target as shown in Figure 2.

The primary function of this pin is for bidirectional serial communication of active background mode commands and data transfer. During reset, this pin is used to select between starting in active background mode or by starting the user's application program. Additionally, this pin requests a timed sync response pulse, allowing a host development tool to determine the correct clock frequency for background debug serial communications. BDC commands are sent serially

from a host computer to the BKGD pin of the target HCS08 or RS08 MCU. All commands and data are sent MSB-first using a custom BDC communications protocol. With a single-wire background debug interface it is possible to use a relatively simple interface pod to translate commands from a host computer into commands for the BDC.

In the case of the SpYder Discovery Kit a low-speed universal serial bus (USB) interface is used. The tool takes the form of a USB Flash Memory Stick, and as the name suggests, *it's a bug eating, MCU spying tool for 8-legged microcontrollers.*

The HCS08 and RS08 SpYder tool is a very low cost tool designed and built by SofTec Microsystems with full collaboration from Freescale Semiconductor. Together with the CodeWarrior IDE, SpYder provides you with everything you need to write, compile, download, in-circuit emulate and debug user code. Full-speed program execution allows you to perform hardware and software testing in real time. The tool works up to bus speeds of 10MHz, supports the 3.3V operation range of the microcontrollers and has on board a socketed target microcontroller which can be replaced with other supported PDIP packaged parts available in small sample quantities FOC from <http://www.freescale.com>. To increase the flexibility of the tool it has a BDM connector for off board debugging of the supported products in other packages or if you need to develop along with other board components.

Last month you enjoyed testing your brakes with the 2-G accelerometer using the SpYder tool and to help you develop new projects using the same tool here is how the SpYder works...

How SpYder Works

SpYder relies on the HCS08 and RS08 Background Debug Controller (BDC). This enables a fast and easy way to program the on-chip FLASH and any other memories. It is the primary debug interface for development, allowing non-intrusive access to memory data and traditional debug features such as CPU register modify breakpoint, and single-instruction trace commands. If you have used the HC05 or other older Freescale cores, as I endeavored to, you will appreciate the advancement this module enables.

The SpYder tool uses a standard USB interface to communicate to the PC and uses the USB bus voltage to power the tool and microcontroller without the bulky wall adapters of old, making it truly portable. This USB power supply can also power the target board, provided you don't run too many motors on it (up to 100mA)! With the tools reliance on USB the SpYder tool's heart is Freescale's MC908JB16 MCU. The MCU has a USB (2.0 low speed) interface and operates from the 5V supplied by USB. As the tool supports the RS08 microcontrollers too, 12V is needed to program the RS08 devices. The MC908JB16 can recognize the target and via control of PTD0 can enable the 12V signal using the ST662 DC-DC converter chip.

Step 1 – SpYder Board Assembly

The essential parts of the SpYder tool are already populated for you, which will allow you to use it as a standalone tool with the socketed micro.

If you want to use another package type or use your own target board you can do so. By adding a 0 ohm resistor or a short circuit on the space next to the bed of nails labeled R2 you are connecting the power supply to the BDM socket so that you can use another target board. The net result is the SpYder Discover Kit turns in to a BDM pod, but don't worry...it still works as a standalone tool. If you don't have a BDM form cable already you can easily build one.

- 6 way IDC socket with strain relief - farnell order code 1097021 (2 of)
- Ribbon Cable with 1.27mm pitch - farnell order code 9187111

Remember to only have one micro connected at a time: either in the socket or on a target.

Step 2 – Debugger Installation

CodeWarrior™ Development Studio for Freescale HC(S)08/RS08 v5.1 is available FOC in its Special Edition incarnation, a copy of which is on the SpYder CD. A detail of what features Special Edition has over the Standard and Professional editions is available at www.freescale.com/codewarrior. Without a license key, the product will run in a 1 KB code-size in limited demonstration mode. To break the 1 KB limit, you have two options:

- contact Freescale to request an unlimited period, free license key to increase the code size limit to 16 KB.
- contact Freescale to request a 30-day limited, free license key to run the compiler without limitations.

Step 3 - Drivers

Once the board is physically the way you want it to be and CodeWarrior is installed you must make the SpYder communicate with your debugger. The next step is to install the SpYder drivers available from the accompanying CD. Instructions on how to do this are available on the inside cover. When the tool is connected to the PC for the first time, Windows recognizes a new USB device; the “**Windows New Hardware Wizard**” dialog box will be opened and asked for the required driver. To complete the installation choose the “**Install Automatically**” option. Simple!

Step 4 – Ready to Start Discovering

It is important to understand that SpYder uses the target microcontroller to execute the in-circuit execution, not an emulator, so microcontroller’s peripherals e.g. Timers, A/D converters, Serial Communication Modules etc. are not reconstructed by software or an external device.

To create a new project with CodeWarrior for any of the devices supported by SpYder the first place to start is with CodeWarrior. From the main menu, select “**File > New Project...**” or from the startup box click “**Create New Project**”. A dialog box will appear prompting you to select your target device from a pull down menu and select “**SofTec HCS08**” as the connection type. Then you set the code type, project name and location. If you have no files to add and don’t want to use the device initialization tools (not recommended if you’re new to CodeWarrior :-0) then you press “**Finish**”. CodeWarrior will create the project, set up the framework for you code and include the header file which has all the register definitions. Once you have some code to debug you **make** the file and then **compile**. There are a few ways to do this by either using the shortcut buttons on the window, using the drop down menu “**Project**” or by pressing F5 then F7.

The first time you enter a debugging session an “**MCU Configuration**” dialog box will open, prompting you to select the debugging hardware connection to your PC. Make sure that the connection type “**USBSPYDER08**” is selected. At this point CodeWarrior has erased and reprogrammed the memory and trimmed the oscillator if feature available on the selected MCU.

Now you have everything you need to start debugging your code, you just need to know how to do it. CodeWarrior enables a variety of ways to analyze program flow via breakpoints, watchpoints and a trace buffer. All these features are implemented by taking advantage of the target microcontroller’s debug peripheral. Your window will consist of:

- a source window with you code displayed
- an assembly window where you can see what the compiler has created of your source code
- a registers window where the CPU registers are visible
- a memory window where you can watch any location or force bytes to another value
- and other windows with data, procedures and commands.

The format of the data and the refresh rates of the data can be changed by right clicking on the window and changing the mode or format. You can save the changes by going to “**File > Save Configuration**”. Take a few minutes to find and play with the Start/Continue, Single Step, Step

Over, Step Out, Assembly Step, Halt and Reset Target buttons. AN3335 - Introduction to HCS08 Background Debug Mode and AN2616 - Getting Started with HCS08 and CodeWarrior Using C are a good place to start to find you way around the debugging environment.

The bed of nails next to the MCU can be connected to a scope to allow you to monitor the pins outwith the debugger in real time.

Modify the Car Accelerometer

The Car Accelerometer gadget from last months Elektor can easily be modified to keep the maximum G reading for each axis displayed. The Mode button can be reprogrammed for clearing the display.

Start by opening last month's project in the CodeWarrior environment (5.1). The source file, main.c is where the application code is located.

In function main case ModBut is the Mode button. To use this button as a display and max level reset the following code can be used;

```
case (ModBut) :    // Mode Button pushed?
    /* Clear Display */
    LEDMag[PosYLED] = LEDOFF;
    LEDMag[PosXLED] = LEDOFF;
    LEDMag[NegYLED] = LEDOFF;
    LEDMag[NegXLED] = LEDOFF;
    /* Clear Maximum G's */
    LEDMax[PosYLED] = LEDOFF;
    LEDMax[PosXLED] = LEDOFF;
    LEDMax[NegYLED] = LEDOFF;
    LEDMax[NegXLED] = LEDOFF;
    break;
```

The second change is to the Byte2LED function. The following code replaces the previous version;

```
void Byte2LED(char Val, char Dir, char SenMode){

    /* Positive Green */
    if ((Val < PosA[SenMode])&&(Val >= PosG[SenMode])) {
    //if (PosA[SenMode] > Val >= PosG[SenMode]) {
        LEDMag[NegYLED + Dir] = (LEDOFF | LEDMax[NegYLED + Dir]);
        if(LEDMag[PosYLED + Dir] > LEDMax[PosYLED + Dir]) {
            LEDMax[PosYLED + Dir] = LEDG;
        }
        LEDMag[PosYLED + Dir] = (LEDG | LEDMax[PosYLED + Dir]);
    }

    /* Positive Amber */
    if ((Val < PosR[SenMode])&&(Val >= PosA[SenMode])) {
    //if (PosR[SenMode] > Val >= PosA[SenMode]) {
        LEDMag[NegYLED + Dir] = (LEDOFF | LEDMax[NegYLED + Dir]);
```

```

if (LEDMag[PosYLED + Dir] > LEDMax[PosYLED + Dir]) {
    LEDMax[PosYLED + Dir] = LEDA;
}
LEDMag[PosYLED + Dir] = (LEDGA | LEDMax[PosYLED + Dir]);
}

/* Positive Red */
if (Val >= PosR[SenMode]) {
    LEDMag[NegYLED + Dir] = (LEDOFF | LEDMax[NegYLED + Dir]);
    LEDMag[PosYLED + Dir] = LEDGAR;
    LEDMax[PosYLED + Dir] = LEDR;
}

/* Negative Green */
if ((Val < NegG[SenMode]) && (Val >= NegA[SenMode])) {
//if (NegG[SenMode] > Val >= NegA[SenMode]) {
    LEDMag[PosYLED + Dir] = (LEDOFF | LEDMax[PosYLED + Dir]);
    if (LEDMag[NegYLED + Dir] > LEDMax[NegYLED + Dir]) {
        LEDMax[NegYLED + Dir] = LEDG;
    }
    LEDMag[NegYLED + Dir] = (LEDG | LEDMax[NegYLED + Dir]);
}

/* Negative Amber */
if ((Val < NegA[SenMode]) && (Val >= NegR[SenMode])) {
//if (NegA[SenMode] > Val >= NegR[SenMode]) {
    LEDMag[PosYLED + Dir] = (LEDOFF | LEDMax[PosYLED + Dir]);
    if (LEDMag[NegYLED + Dir] > LEDMax[NegYLED + Dir]) {
        LEDMax[NegYLED + Dir] = LEDA;
    }
    LEDMag[NegYLED + Dir] = (LEDGA | LEDMax[NegYLED + Dir]);
}

/* Negative Red */
if (NegR[SenMode] >= Val) {
    LEDMag[PosYLED + Dir] = (LEDOFF | LEDMax[PosYLED + Dir]);
    LEDMag[NegYLED + Dir] = LEDGAR;
    LEDMax[NegYLED + Dir] = LEDR;
}

/* Neutral */
if ((Val < PosG[SenMode]) && (Val > NegG[SenMode])) {
//if (PosG[SenMode] > Val > NegG[SenMode]) {
    LEDMag[PosYLED + Dir] = (LEDOFF | LEDMax[PosYLED + Dir]);
    LEDMag[NegYLED + Dir] = (LEDOFF | LEDMax[NegYLED + Dir]);
}
}

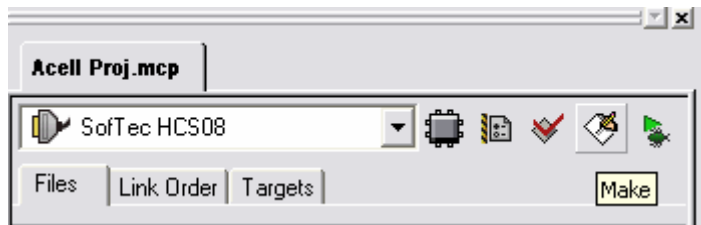
```

The code uses the passed parameter Val to scan the range of the G spectrum and finds out whether it is neutral, positive (G, A or R) or negative (G, A or R). Once it has found the correct range. It clears the opposite axis but leaves the maximum LED displayed. It then decides whether Val is a new maximum and updates is appropriate (always if Red). The routine then displays the magnitude with the maximum on that axis.

Remember to add the new LED Bitmap definitions and button 3 (Mode button) max level to the header file and LEDMax bitmap to the global variables in main.c.

You can now save the project under a different name if you wish by “**File > Save A Copy As...**” As soon as the project is made it is saved so the previous project version will be lost. You can also save the individual source files by “**File > Save As...**” when in the source code window.

Once the code has been modified you must compile/make the project. If no errors messages are created you can now debug the project using the button to the right (green arrow with insect).



Ensure that USBSpYder08 is selected as the Hardware Model and the MC9S08QG8 as the Device. CodeWarrior will now erase and reprogram the device with the new project.

If you go to the Byte2LED function and place a breakpoint at the first if statement by right clicking CodeWarrior will halt the program when it reaches this point. In the Data window you will be able to see the value of Val and by single stepping through the function you can see the program flow in to the correct range. You can add more breakpoints and remove them by right clicking around other parts of the program.

Summary

The SpYder Discovery Kit is a surprisingly flexible tool that does meet all the needs of low budget developments. It supports Freescale's 8-pin S08 devices with the ability to extend pin count using the off chip target option. CodeWarrior is a very powerful tool and a bit daunting at first, but once you been shown where all the necessary buttons are you can get started very quickly and you can learn about other capabilities as you get used to the tool and with the help if the documents and training freescale provide online.

Happy Developing!

Inga

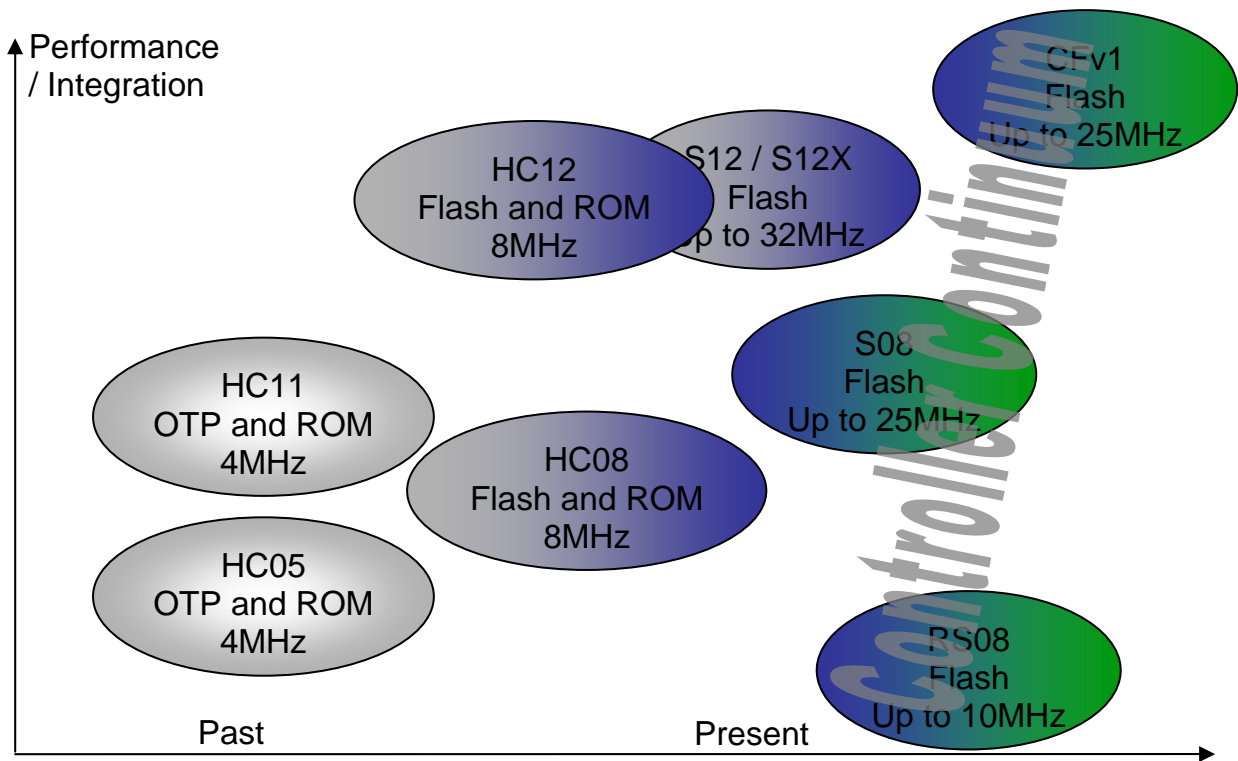


Figure 1 - Core Roadmap

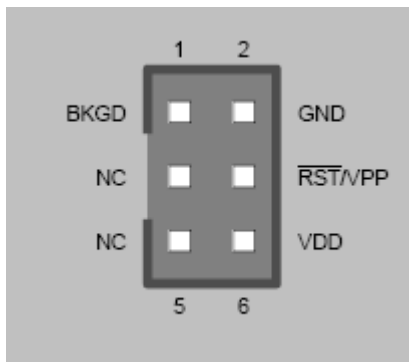


Figure 2 – BDM Connector

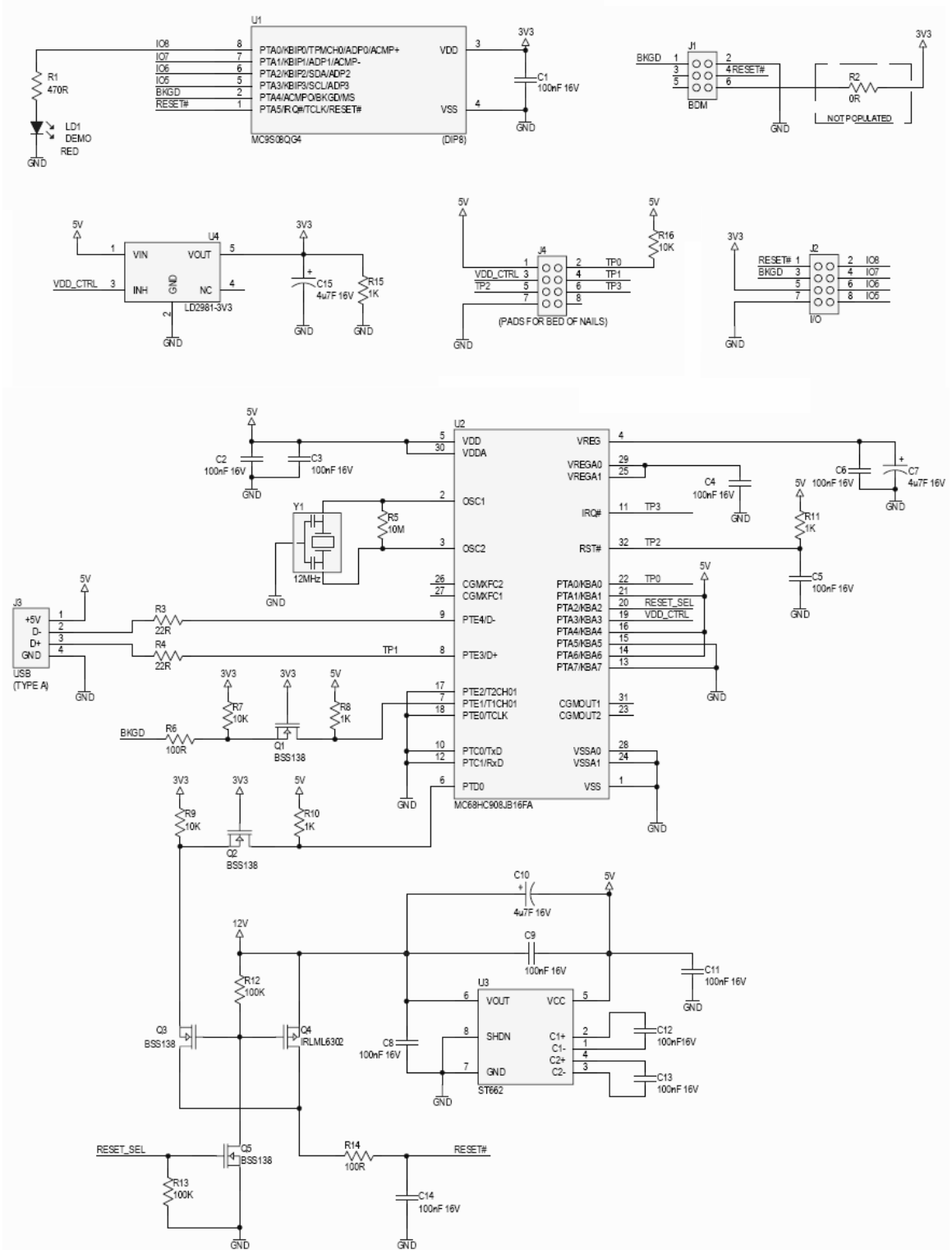


Figure 3 – SpYder Schematic

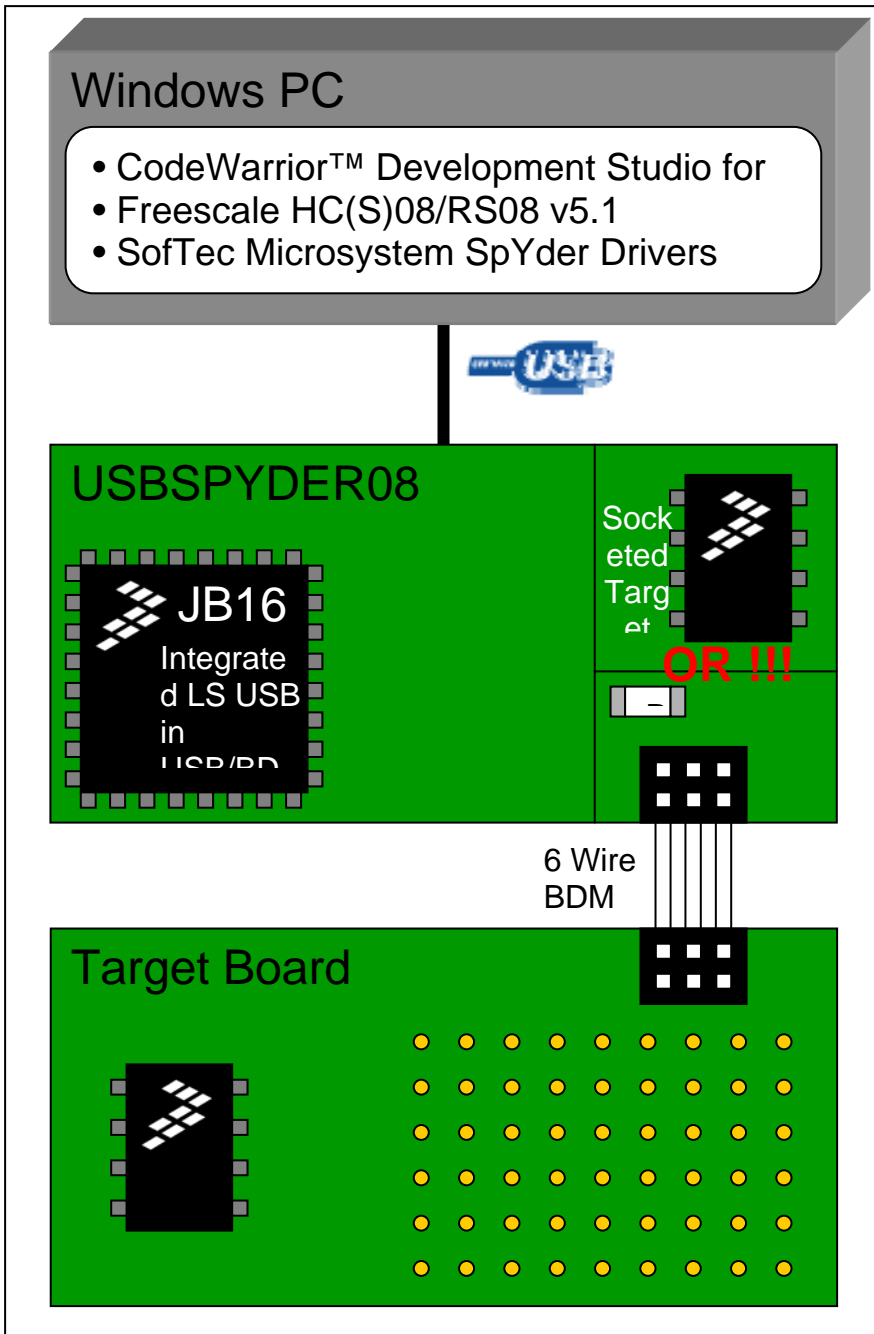


Figure 4 – Block Diagram