



Touches sensibles (CapSense®) sur platine PSOC5 - STI2D-SIN

publié le 16/02/2013 - mis à jour le 23/04/2014

Descriptif :

Dans cet article sont exposés le principe de fonctionnement de la technologie CapSense ainsi qu'une exploitation possible en STI2D-spécialité SIN.

Sommaire :

- 1. Principe de fonctionnement
- 2. Exploitation en STI2D spécialité SIN : Projet PSOC CREATOR
- 3. Carte CAPSENSE



Analyse et utilisation de la technologie des touches capacitive (Cypress CapSense®) en Sciences et Technologie de l'Industrie et du Développement durable (STI2D), spécialité Systèmes d'Information et Numériques (SIN).



L'utilisation de la technologie CAPSENSE en spécialité STI2D-SIN est un point d'entrée intéressant puisque cette technologie, aujourd'hui florissante, équipe une multitude d'objets de la vie courante (PAD d'ordinateur portable, touches sensibles sur les téléviseurs, smartphones...) et participe également à un positionnement marqué dans le cadre du développement durable.

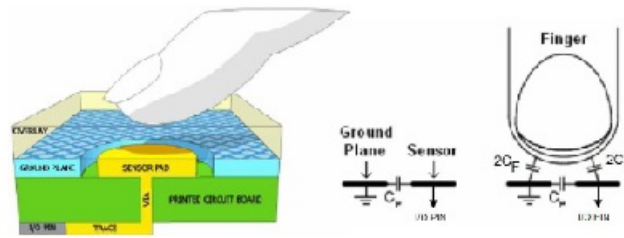
"Cypress is the world leader in Capacitive Sensing technologies. Our broad range of solutions provide robust noise immunity, enable quick time to market and system scalability and have replaced more than 3.5 Billion mechanical buttons over the past several years. The CapSense® portfolio ranges from simple buttons and sliders to more sophisticated solutions integrating other system components to reduce total

BOM cost" (Source : www.cypress.com : *BOM : Bill of Material)

● 1. Principe de fonctionnement

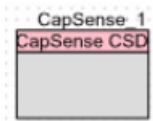
Lorsqu'un objet conducteur est placé à proximité des deux électrodes (Ground Plane et Sensor Pad), la capacité totale C_x ($C_x = C_P$ initialement) change et augmente pour devenir au maximum $C_x = C_P + C_F$. La technologie CapSense CSD (CSD pour CapSense® Sigma Delta) permet de détecter le changement de la

capacité C_x . L'électronique du macro-composant CapSense permet d'informer le programme (fonction "C" dédiée) lorsque la touche est sollicitée.



- Chaque touche déclarée dans le module CAPSENSE devra bénéficier d'un temps d'analyse (SCAN) permettant d'identifier l'appui .

● 2. Exploitation en STI2D spécialité SIN : Projet PSOC CREATOR

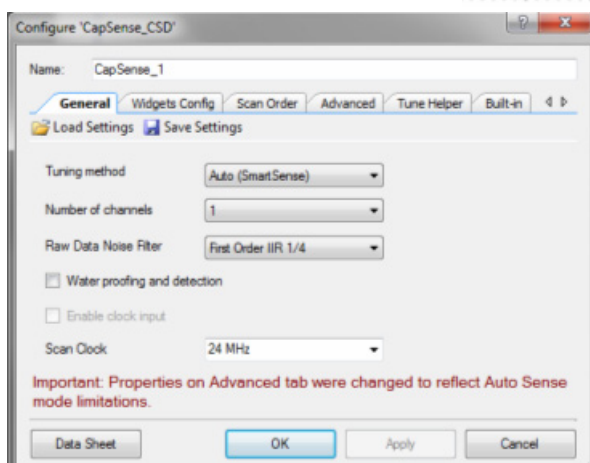
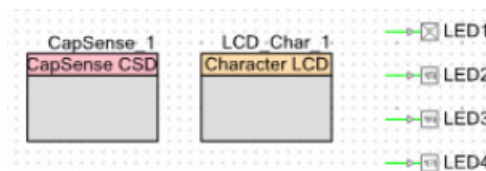


L'idée est ici avant tout de profiter de la simplicité de mise en oeuvre offerte par l'interface de management du composant CAPSENSE dans PSOC CREATOR. Ce type de technologie pourra être intégrée dans des projets présents ou à venir.

On propose ici une exploitation des capteurs CAPSENSE permettant de gérer un "slider linéaire" ainsi que 6 boutons permettant, par exemple, la commande d'un baladeur media.

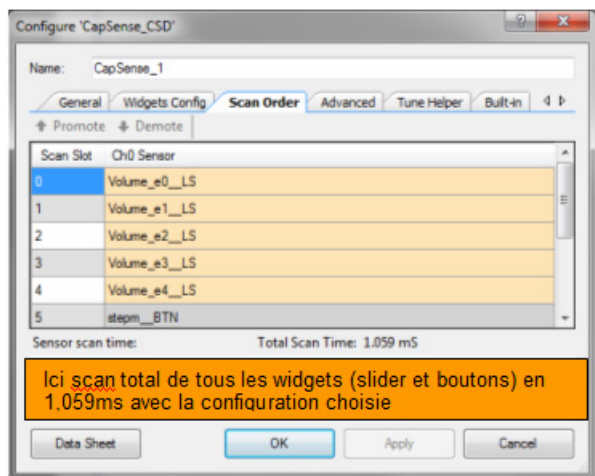
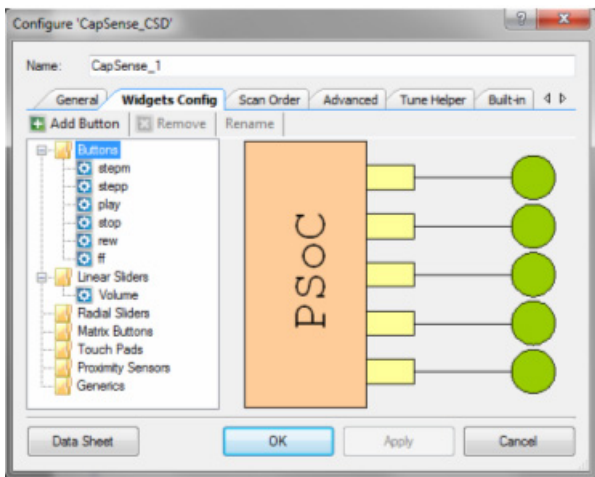
Le Kit CY8CKIT-001A ou B offre ce type de captage avec deux boutons seulement. Une carte supplémentaire a été ici réalisée afin de contextualiser les fonctionnalités offertes par la technologie CAPSENSE (slider linéaire 5 positions et 6 boutons).

- Configuration du top design sous PSOC CREATOR et configuration du CAPSENSE

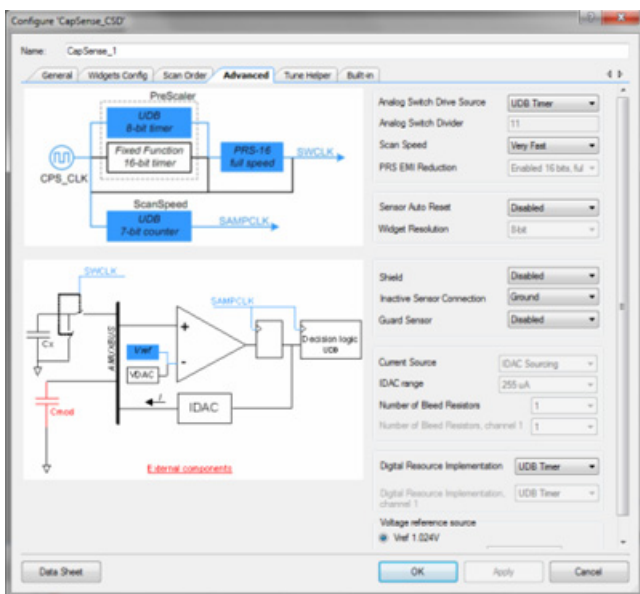


Ici la configuration générale du module CAPSENSE. Ce module pourra posséder plusieurs "widgets" (boutons, sliders...). Une fréquence de travail de 24MHz permettra une plus grande rapidité de SCAN des différents Widgets.

Ici la configuration des différents widgets.

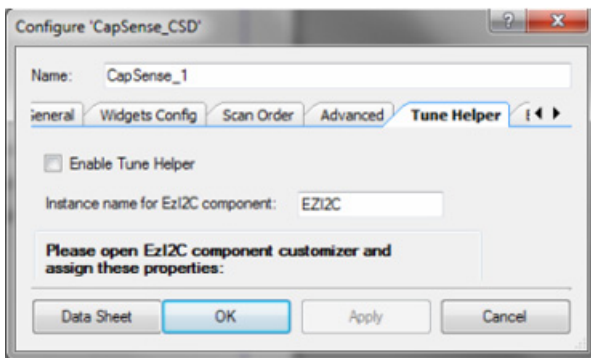


Ici la configuration de l'ordre de scan des différents widgets.



Ici la configuration Advanced. Noter la présence de la fonction PRS (pour Pseudo Random Sequence) qui permet de créer une horloge limitant la pollution électromagnétique.

Ici dé-validation de la fenêtre d'aide Tune Helper.



- Configuration des broches de la carte CAPSENSE

Alias	Name	Pin	Lock
Cmod_CH0	\CapSense_1:CmodCH0\	P15[5]	☒
#_BTN	\CapSense_1:PortCH0\[10]	P3[5]	☒
rew__BTN	\CapSense_1:PortCH0\[9]	P3[4]	☒
stop__BTN	\CapSense_1:PortCH0\[8]	P3[3]	☒
play__BTN	\CapSense_1:PortCH0\[7]	P3[2]	☒
stepp__BTN	\CapSense_1:PortCH0\[6]	P3[1]	☒
stepm__BTN	\CapSense_1:PortCH0\[5]	P3[0]	☒
Volume_e4__LS	\CapSense_1:PortCH0\[4]	P5[4]	☒
Volume_e3__LS	\CapSense_1:PortCH0\[3]	P5[3]	☒
Volume_e2__LS	\CapSense_1:PortCH0\[2]	P5[2]	☒
Volume_e1__LS	\CapSense_1:PortCH0\[1]	P5[1]	☒
Volume_e0__LS	\CapSense_1:PortCH0\[0]	P5[0]	☒
	\LCD_Char_1:LCDPort\[6:0]	P2[6:0]	☒
	LED1	P1[4]	☒
	LED2	P1[5]	☒
	LED3	P1[6]	☒
	LED4	P1[7]	☒

- Programme main.c

```

1 | #include <device.h>
2 |
3 | #define LED_ON 1 /* pour allumage */
4 | #define LED_OFF 0 /* pour extinction */
5 |
6 | void main()
7 | {
8 |     uint8 stepp,stepp,play,stop,rew,ff,volume;
9 |     /* Initialisation LCD */
10 |    LCD_Char_1_Start();
11 |
12 |    CYGlobalIntEnable; /* Autorisation globale des interruptions */
13 |
14 |    /* Initialisation capsense */
15 |    CapSense_1_Start(); //démarrage du composant capsense
16 |    CapSense_1_InitializeAllBaselines();
17 |    CapSense_1_ScanEnabledWidgets();
18 |
19 |    for(;;)
20 |    {
21 |        while(CapSense_1_IsBusy())//attente de disponibilité (fin du scan)
22 |        {
23 |            CapSense_1_UpdateEnabledBaselines();//gestion capsense
24 |            CapSense_1_ScanEnabledWidgets();//gestion capsense, scan de l'ensemble des widgets
25 |
26 |            /*détection des états des touches et slider */
27 |            stepp = CapSense_1_CheckIsWidgetActive(CapSense_1_STEPP_BTN);//detection bouton step_p
28 |            stepm = CapSense_1_CheckIsWidgetActive(CapSense_1_STEPM_BTN);//detection bouton step_m
29 |            play = CapSense_1_CheckIsWidgetActive(CapSense_1_PLAY_BTN);//detection bouton play
30 |            stop = CapSense_1_CheckIsWidgetActive(CapSense_1_STOP_BTN);//detection bouton stop
31 |            rew = CapSense_1_CheckIsWidgetActive(CapSense_1_REW_BTN);//detection bouton rew
32 |            ff = CapSense_1_CheckIsWidgetActive(CapSense_1_FF_BTN);//detection bouton ff
33 |            volume = (uint8)CapSense_1_GetCentroidPos(0);//detection position slider linéaire
34 |            //si volume=255 => pas de présence
35 |            //la valeur effective de volume est comprise entre 0 et 100

```

```

34 |
35 |
36 | /* Visualisation sur leds de l'appui sur step_p, step_m, play, stop */
37 | if (stepm)
38 |     (LED1_Write(LED_ON));
39 | else
40 |     (LED1_Write(LED_OFF));
41 | if (stepp)
42 |     (LED2_Write(LED_ON));
43 | else
44 |     (LED2_Write(LED_OFF));
45 | //ici détermination du mode de lecture
46 | if (play)
47 |     (LED3_Write(LED_ON));
48 | else
49 |     (LED3_Write(LED_OFF));
50 |
51 | if (stop)
52 |     (LED4_Write(LED_ON));
53 | else
54 |     (LED4_Write(LED_OFF));
55 |
56 | /* Fin d'affichage sur led */
57 |
58 | //Affichage de la valeur du volume sur afficheur LCD
59 | if (volume<=100)
60 |     (LCD_Char_1_Position(0,0);LCD_Char_1_PrintString(""));LCD_Char_1_Position(0,0);//effacement
61 |     LCD_Char_1_PrintNumber(volume);
62 | }
63 |
64 | }
65 |
66 |
67 | /* {} END OF FILE */

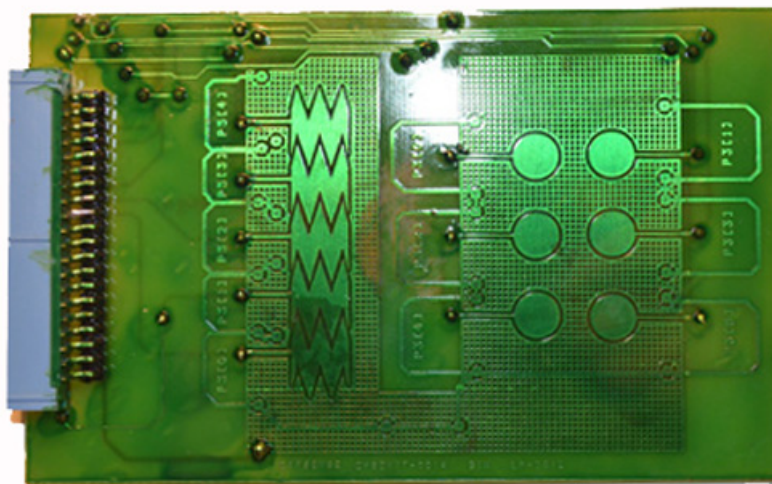
```

- Télécharger le projet complet

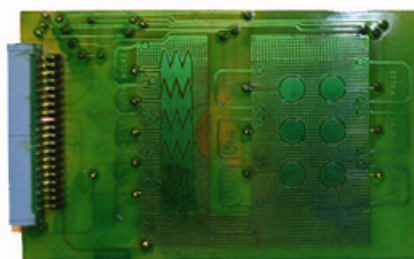
 **Projet CapSense® PSOC CREATOR** (Zip de 2 Mo)
Le Projet complet

● 3. Carte CAPSENSE

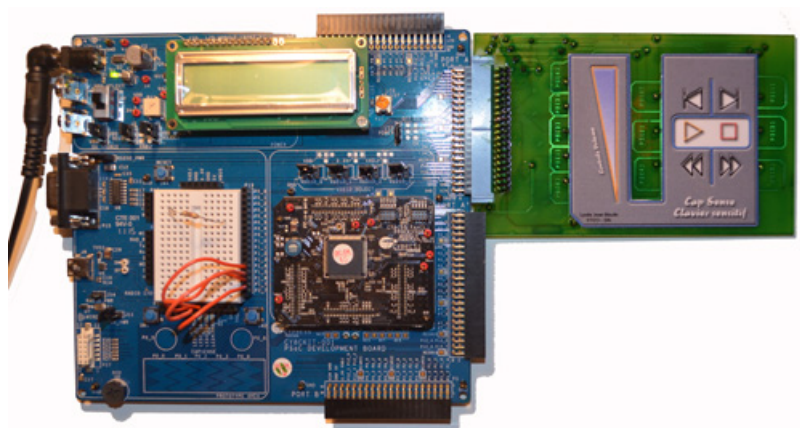
Cette carte est a connecter sur le portA du Kit CY8CKIT-001B. Voici une description de celle-ci ainsi que les documents de fabrication.



Circuit imprimé nu, on distingue les boutons à droite et le slider à gauche



Circuit imprimé avec son "habillage". Diélectrique formé de papier plastifié



Carte complète connectée au kit PSOC5

- Télécharger les documents de fabrication au format "ORCAD"

 **Maquette CapSense®** (Zip de 141.7 ko)
Les documents de fabrication au format "ORCAD".

Laurent Proust 2013



Académie
de Poitiers

Avertissement : ce document est la reprise au format pdf d'un article proposé sur l'espace pédagogique de l'académie de Poitiers.

Il ne peut en aucun cas être proposé au téléchargement ou à la consultation depuis un autre site.