



## Pourquoi FOX<sup>1</sup> est-il différent?

établi le 11.11.2008 / RFO  
mise à jour le 16.09.2011 / CA

### Choix des composants

Le microordinateur FOX est un ordinateur intégré. Inséré dans un appareil, il devra effectuer longtemps son service sans poser de problèmes. Les composants constituant un ordinateur intégré FOX sont des composants industriels ayant fait leurs preuves et livrables à long terme. Grâce à sa structure simple et robuste, le produit peut être implanté sur un circuit imprimé à 4 couches, rendant possible et intéressante soit l'adaptation de l'affichage, soit la réalisation d'une version selon les besoins particuliers des clients, pour des quantités à partir de 100 pièces.

### Concept

A la différence d'un PC ou d'un PC industriel, le FOX découle d'un concept qui n'a pas besoin de s'aligner sur les cycles d'innovation imposés par le marché de consommation. Il faut être indépendant de la tyrannie des logiciels toujours plus gourmands en puissance, exigeant un matériel de plus en plus performant. Les bus de données de 16 bits procurent un compromis optimal entre les performances et les coûts.

### Développement

Priorité au client! Il doit être en mesure de programmer lui-même des modifications sans diplôme universitaire en informatique. Les outils de programmation nécessaires pour le développement, eigerStudio et eigerGraphicsuite, sont téléchargeables gratuitement depuis le site [www.eigergraphics.com](http://www.eigergraphics.com).

L'interface graphique peut être réalisée avec Photoshop, Paint, Word, Excel ou Powerpoint.

Il est facile d'insérer des photos dans l'application.

### Interfaces vers l'application

Les ordinateurs intégrés FOX disposent d'interfaces série vers l'application. Un circuit de commande pilote l'application. Il contient toutes les E/S et les alimentations nécessaires pour la commande des capteurs et des actionneurs de la machine. Les applications très simples peuvent

---

<sup>1</sup> „FOX“ c'est le FOX embedded computer de l'eigerPanel





aussi être pilotées directement par le FOX. Lorsqu'un automate programmable est déjà présent, FOX peut communiquer avec lui au moyen d'une interface série.

## Tâches du processeur d'affichage

Le processeur d'affichage gère l'interface homme machine (HMI) et établit le lien entre la commande en temps réel et l'opérateur. Grâce à FOX, l'opérateur peut utiliser l'écran tactile en souplesse, sans attente gênante. Le processeur d'affichage exécute les tâches suivantes:

1. Afficher le contenu d'une image: il peut s'agir d'une photo ou d'un contenu d'écran calculé (rendu).
2. Rafraîchir l'affichage. L'affichage est rafraîchi 60 fois par seconde pour créer une image exempte de scintillement.
3. Interroger et traiter les données de l'écran tactile pour l'interaction avec l'opérateur
4. Communiquer avec l'application via les interfaces série.

## Calcul de la bande passante pour les affichages VGA et WVGA

L'affichage VGA compte 640 pixels en largeur et 480 pixels en hauteur, soit un total de 307'200 pixels. Sachant qu'il faut 16 bits (2 octets) pour afficher un pixel, cela représente 614'400 octets par image. En rafraichissant l'affichage 60 fois par seconde, il en résulte un débit de données de 36'864'000 octets par seconde ou 36.9 MB/s nets. En réalité on travaille avec une horloge pixel de 25 MHz, parce que le timing d'affichage contient aussi des portions invisibles (fenêtres horizontales/ verticales avant et arrière)

L'affichage WVGA compte 800 pixels horizontaux, et le débit net de données correspondant s'élève à 46'080'000 octets/s ou 46.1 MB/s. L'horloge pixel pour l'affichage WVGA tourne à 33 MHz.

Le timing du rafraichissement vidéo s'effectue de façon absolument précise pour éviter tout scintillement de l'affichage!

## Différences de matériel

On trouve dans l'industrie des microordinateurs munis d'une UMA "Unified Memory Architecture" (par ex. les cartes ARM). Dans ces ordinateurs, la mémoire vidéo gardant le contenu de l'écran est stockée dans le même module de mémoire que le programme et les données. Il en résulte un





goulet d'étranglement lors de l'accès à la mémoire, et il faut augmenter la vitesse d'horloge pour obtenir les performances souhaitées.

FOX, au contraire, dispose d'une mémoire vidéo à deux niveaux, indépendante de la mémoire du processeur. Le microcontrôleur travaille sans être influencé par le rafraîchissement d'affichage. Le processeur graphique "EVE anna" accélère les opérations de copie et de dessin. Grâce à l'architecture parallèle comportant 3 bus de données, les données circulent constamment, sans cycles d'attente. Les SRAM rapides utilisées comme RAM vidéo ne sont affectées d'aucune latence.

## **Différences dans le (développement du) logiciel**

L'architecture logicielle du FOX est basée sur une machine virtuelle (eigerVM). La machine virtuelle traite le bytecode généré par le compilateur de l'eigerStudio. L'environnement de développement (EDI ou IDE en anglais pour Integrated Development Environment) eigerStudio vérifie la syntaxe du code source (de votre programme) et le traduit en un fichier EVI (compilation). L'eigerVM interprète ensuite ce bytecode. Le programme est chargé de la carte Compact Flash dans la mémoire principale puis exécuté. Plusieurs programmeurs peuvent écrire différentes pages (Views) qui sont ensuite assemblées en un projet global. (Il suffit de tous les copier sur la carte CF). L'intégration généralement pénible du logiciel disparaît totalement.

## **Mode de fonctionnement du logiciel illustré par une tâche simple**

Dans l'exemple suivant, il faut dessiner un rectangle rouge plein à la position (20/30), large de 200 pixels et haut de 50 pixels (voir Figure 1).

eigerStudio traduit les instructions du programme en un bytecode exécutable. La machine virtuelle eVM va chercher la commande de chargement des registres (flèche rouge) et charge les 5 registres de la machine virtuelle prévus pour cette tâche. L'instruction suivante (flèche violette, écriture violette) est interprétée comme "dessine un rectangle plein". Le rectangle apparaît sur l'écran.



Compilateur  
eigerStudio

Code source dans eigerScript

```
eI.Pos_Xl := 20      ; Position Xl en haut à gauche  
eI.Pos_Yl := 30      ; Position Yl en haut à gauche  
eI.Width := 200      ; Largeur  
eI.Height := 50      ; Hauteur  
eI.FillColor := red   ; Couleur de remplissage  
Draw.RectangleFilleD () ; Dessiner le rectangle
```

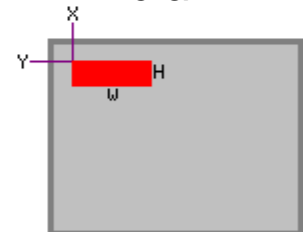
Byte code pour

```
210B      ; Charger les registres  
0005      ; 5 registres  
0010 0014 ; REG 0x0010 := 20  
0011 001E ; REG 0x0011 := 30  
0014 00C8 ; REG 0x0014 := 200  
0015 0032 ; REG 0x0015 := 50  
0019 001F ; REG 0x0019 := rot  
440B      ; Draw.RectangleFilleD ()
```

Registres eVM

REG	VAL
0010	0014
0011	001E
0012	????
0013	????
0014	00C8
0015	0032
0016	????
0017	????
0018	????
0019	001F

Écran



eVM Instruction Decoder

- 1) Charger cinq registres
- 2) Dessiner le rectangle

FOX\_HowItWorks\_01.bmp

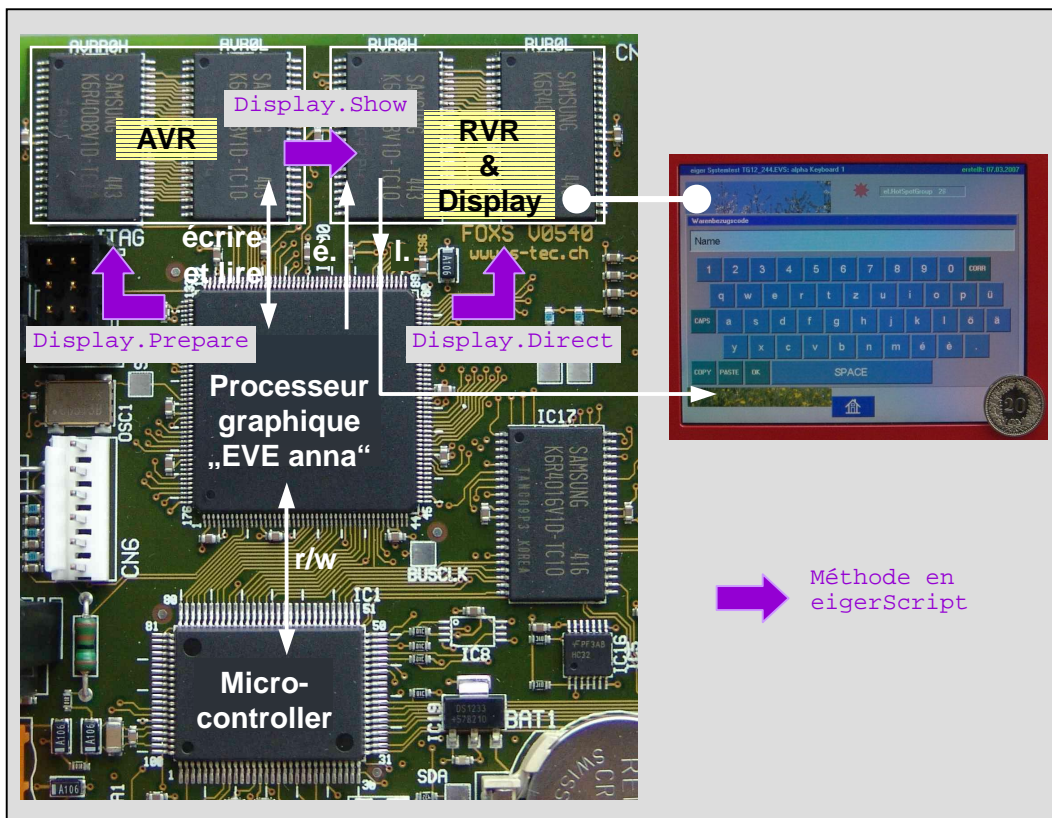
Figure 1: Mode de fonctionnement du logiciel illustré par une tâche simple, par ex. dessiner un rectangle rouge plein.

## Gestion de l'écran

L'écran du FOX fonctionne comme un tableau d'affichage. Ce qui a été dessiné en dernier apparaît "tout en haut" à l'écran. L'unité de mesure de programmation est le pixel. On compte les coordonnées X de gauche à droite et les coordonnées Y de haut en bas (voir Figure 1).

Il y a deux niveaux d'écran: le rafraîchissement de la RAM vidéo RVR induit le rafraîchissement d'écran (voir Figure 2). Les données dans cette RAM vidéo sont visibles à l'écran.

Le moteur vidéo EVE anna peut accéder à la RAM vidéo accessible AVR en lecture et en écriture. Les données de cette RAM vidéo ne sont pas visibles à l'écran. Le processeur graphique peut écrire simultanément dans les deux RAM vidéo et générer ainsi une copie des données. Lors de l'ouverture d'une fenêtre (par ex. clavier virtuel ou saisie du NIP), on peut écrire dans la RAM vidéo RVR indépendamment de AVR. Lors de la fermeture (l'effacement) de la fenêtre, le contenu inchangé de AVR est simplement copié à nouveau dans RVR, ce qui rétablit l'image initiale. eigerScript met à disposition à cet effet des méthodes performantes de la classe "display.XXX".



**Figure 2:** Part de l'ordinateur intégré FOXS illustrant le gestion de l'écran par le moteur vidéo „EVE anna“ et les deux RAM vidéo (RAM vidéo accessible et RAM vidéo de rafraîchissement). Les méthodes performantes que eigerScript met à disposition pour gérer l'écran sont de la classe "Display.XXX", par ex. Display.prepare().

## eigerScript: une programmation événementielle

Contrairement aux langages procéduraux, eigerScript implémente une programmation événementielle. En réponse aux événements dans le système (timer, hotspot etc), de petits sous-programmes (handler) sont exécutés. Le temps de réaction est extrêmement court: un changement de page est effectué en 20ms.

L'exemple de code suivant contient trois sous-programmes (sous-routines) qui sont exécutés lorsque l'utilisateur touche le hotspot correspondant sur l'écran tactile.

```
STRING [64] HSl_Text.$ = 'TopLine +1'
```

---

<pre>SUB   HSl_LE Fill.LabelParameter ( KMCO_Button_UP_Style ) Load.Pos_XlYl ( Btn_COLL, Btn_ROWl ) eI.Width := 120 Label.Text ( HSl_Text.\$ ) ENDSUB</pre>	<b>Button Leave Event</b>
---	---------------------------

---

<pre>SUB   HSl_DN Fill.LabelParameter ( KMCO_Button_DN_Style ) Load.Pos_XlYl ( Btn_COLL, Btn_ROWl ) eI.Width := 120 Label.Text ( HSl_Text.\$ ) ENDSUB</pre>	<b>Button Down Event</b>
---	--------------------------

---

<pre>SUB   HSl_UP CallSubroutine ( HSl_LE ) IF EF_TopLine.I &lt; EF_MaxLines.I THEN      EF_TopLine.I := EF_TopLine.I + 1     Load.Pos_XlYl ( EF_Pos_X, EF_Pos_Y )     CallSubroutine ( EditField_DrawLayout )  ENDIF ENDSUB</pre>	<b>Button Up Event</b>
--	------------------------

---

```
; Installer les hotspots -----
CallSubroutine ( HSl_LE )
HotSpot.Install ( NIL, HSl_LE, HSl_DN, HSl_UP )
```

FOX\_HowItWorks\_02.bmp



## Avantages et inconvénients de cette solution

### La solution FOX présente les avantages suivants:

- Conception simple et robuste, sans fréquences élevées d'horloge, d'où réduction des EMI (interférences électromagnétiques).
- Layout du circuit imprimé simple permettant l'adaptation économique à différents écrans.
- Traitement hautement parallèle, d'où vitesse élevée pour de basses fréquences d'horloge.
- Temps de réaction brefs, aussi pour les projets comportant plusieurs "views".
- Disponibilité des composants industriels à long terme.
- Le noyau graphique est intégré dans la puce programmable comme PI (propriété intellectuelle); pas d'annonce de fin de production de la solution graphique en vue.
- Faible consommation de courant, donc échauffement réduit.
- Aucune pièce mobile -> pas d'usure.
- Développement sur un PC avec des outils gratuits.
- Grâce à la machine virtuelle eigerVM, l'intégration du logiciel est superflue.
- La machine virtuelle peut être utilisée sur d'autres hardwares.
- Solution globale économique en raison de l'interaction optimale du logiciel et du matériel.
- Le logiciel peut être copié simplement (il suffit de copier la carte CF).
- Le logiciel peut être envoyé par mail comme fichier ZIP.
- Le logiciel peut être remplacé simplement (il suffit de changer de carte CF).
- Le bytecode est extrêmement compact (les petits programmes sont chargés rapidement).

### La solution FOX présente les inconvénient suivants

- Il faut plusieurs puces de RAM vidéo (coûteux)
- Puce graphique couteuse



## Ordinateur intégré FOX

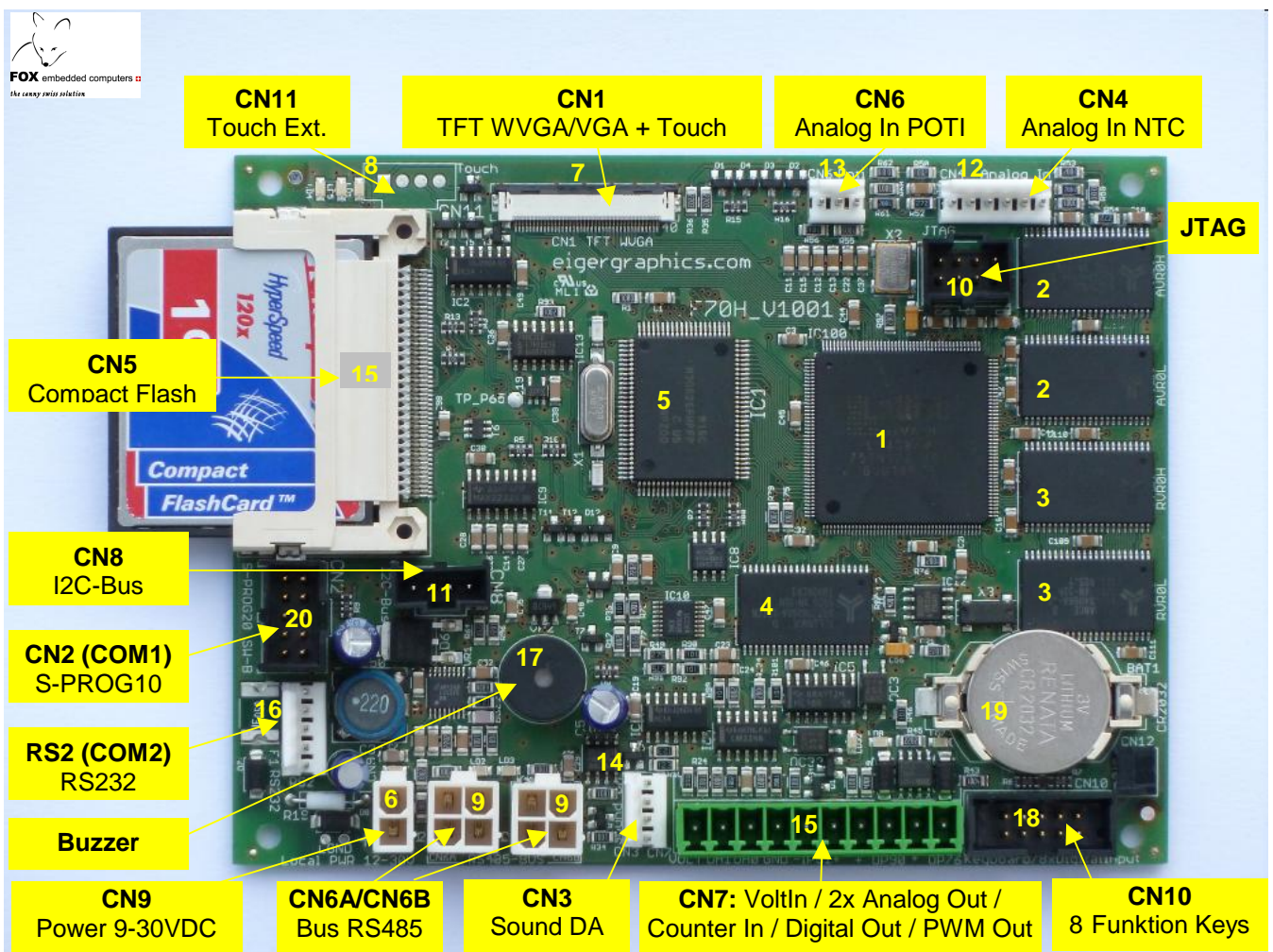


Figure 3: Micro-ordinateur F70H de l'eigerPanel70 avec ses interface.