Tiny InterNet Interface (TINI)

Didier DONSEZ

Université Joseph Fourier

IMA –IMAG/LSR/ADELE

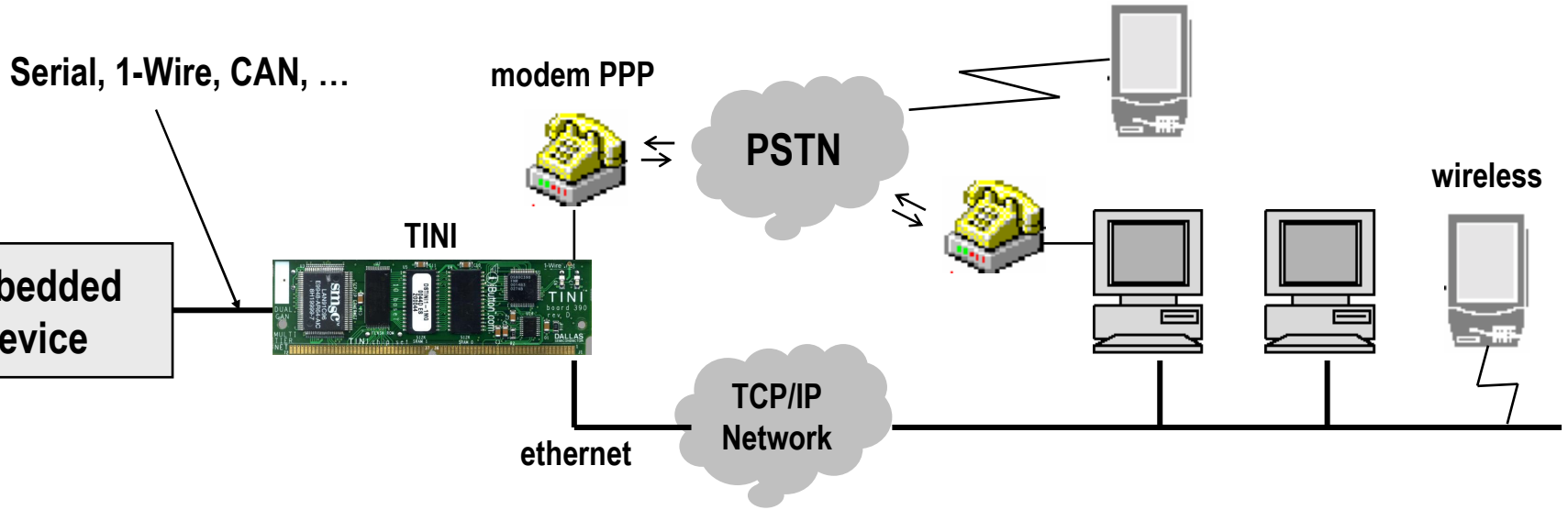
Didier.Donsez@imag.fr, Didier.Donsez@ieee.org

Sommaire

- Matériel
- Runtime
- Programmation Java
- Divers

Tiny InterNet Interface (TINI)

- Motivation : Coupleur Réseaux
  - entre Ethernet et des réseaux IP
  - et des périphériques sur des réseaux non IP
    - Capteur température, actionneur circuit électrique (CAN, 1Wire, …)
    - Enregistreur ECG, terminal GPS, … (RS232)
- Applications : *Web based management*
  - Mesure et Contrôle à partir de l'Internet (fil/sans fil)



Tiny InterNet Interface (TINI)

- Coupleur réseaux
- Caractéristiques
  - Faible coût (quelques dizaines d'euros)
  - Encombrement (format SIMM)
  - Fiabilité (mémoire auto-alimentée)
- Environnement logiciel
  - OS multi-processus et multi-thread
  - slush, Langage de commande Unix *like* et *light*
  - JVM basé JDK1.1 et Environnement de programmation Java

Applications

- Contrôle Industriel
  - Équipement industriel (automates, senseurs, activateurs, …)
- Contrôle et Supervision d'instruments par le Web
  - fil / sans fil
- Conversion de Protocoles
  - Passerelle entre des protocoles IP et des protocoles patrimoniaux ou embarqués (CAN, 1Wire, …)

Matériel

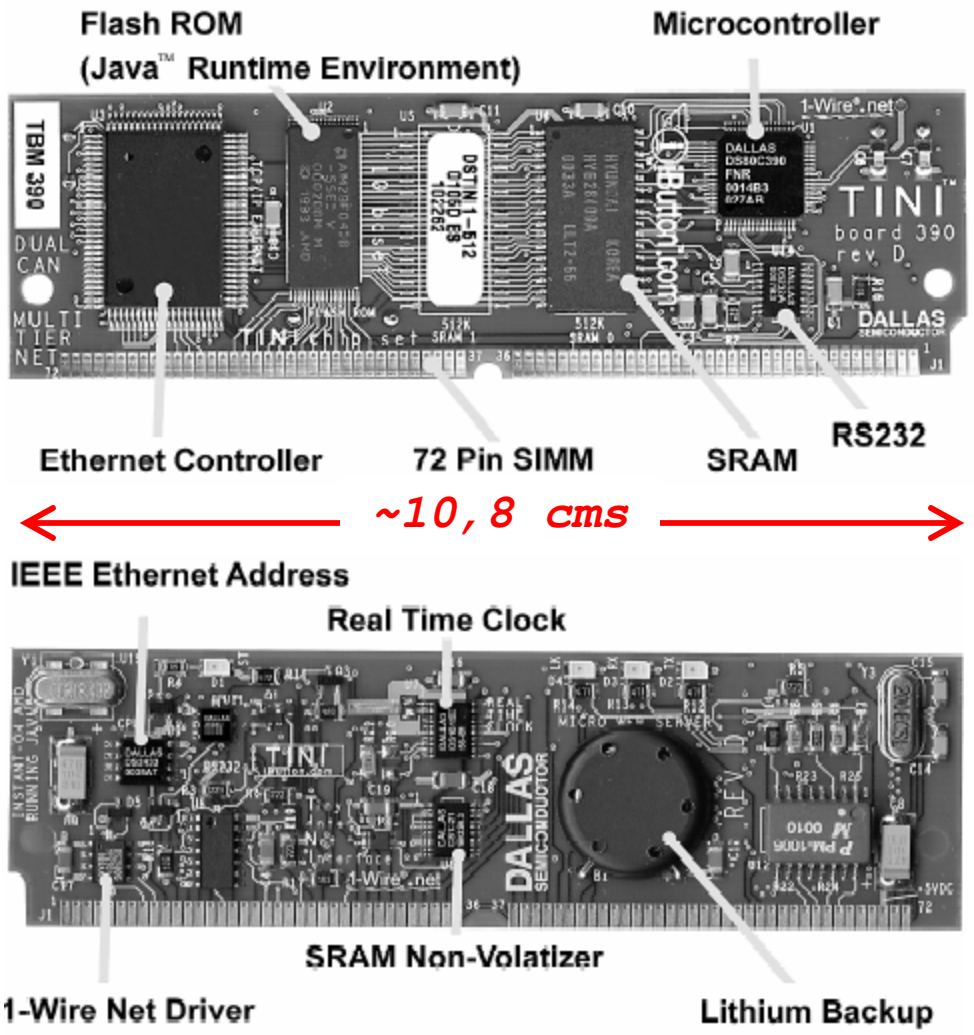
- DS TINI TMB390
  - Processeur embarqué DS80C390 40 Mhz
  - 512 KB ou 1024KB de NV SRAM
    - RAM Non Volatile sauvée par pile
  - Carte format SIMM
- DS TINIm400
  - Processeur embarqué DS80C400 75 Mhz
  - 1MB NV SRAM, 1MB Flash
  - Carte format SIMM
- Cartes Socket
  - formats E10, E20, STEP (systronix), Taylec …

DS80Cxxx *(Dallas Semi-Conductor)*

- DS80C390
  - µC 40Mhz
  - Instruction set : 8051 + special functions
  - 16/32-bit math coprocessor
  - 3 16-bit timer/counters
  - 4 Ports 8bit I/O
  - 2 Controller bus CAN
  - Address space
    - Max 4MB Data + Program
    - 3 modes : 16b contiguous, 22b contiguous, 22b 8bit-paged
- DS80C400
  - µp 75 Mhz
  - PMM (Power Management Mode) : fréquence ralenti par 256

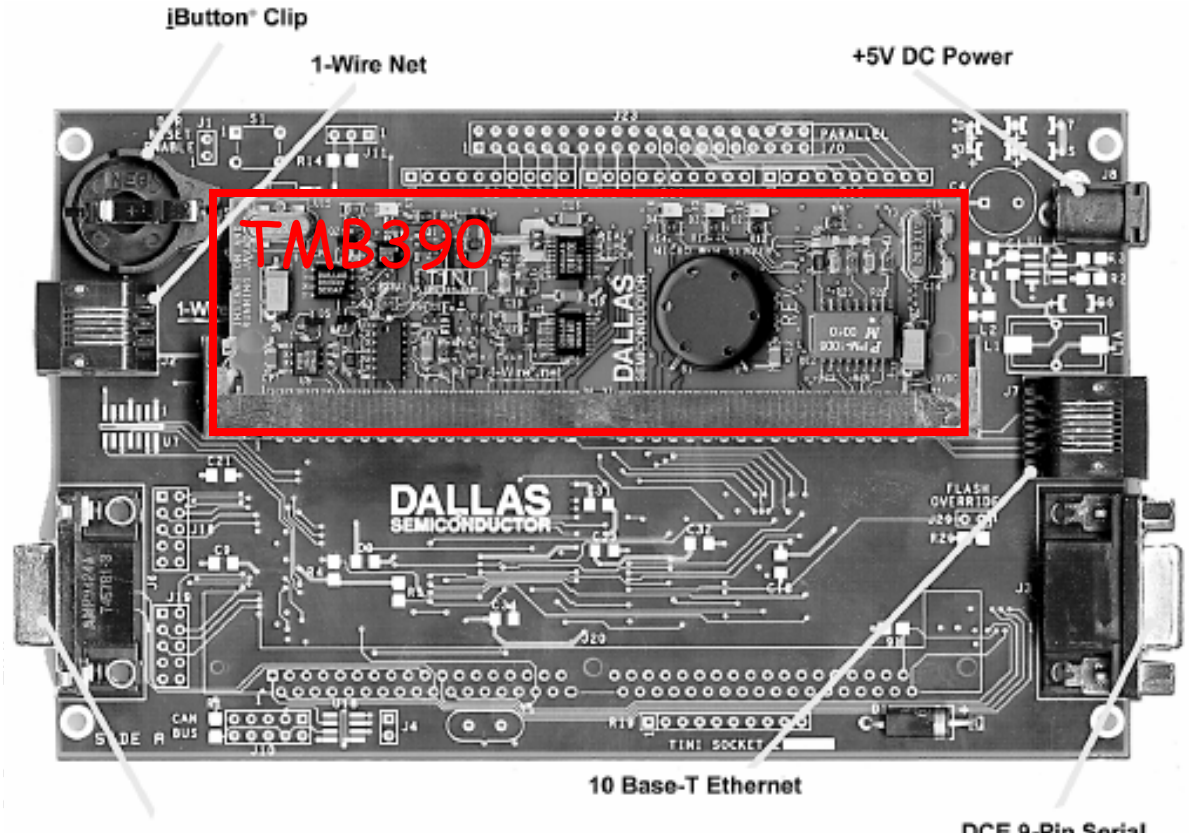
Carte modèle TMB390

- Format SIMM (depourvu de connecteur)

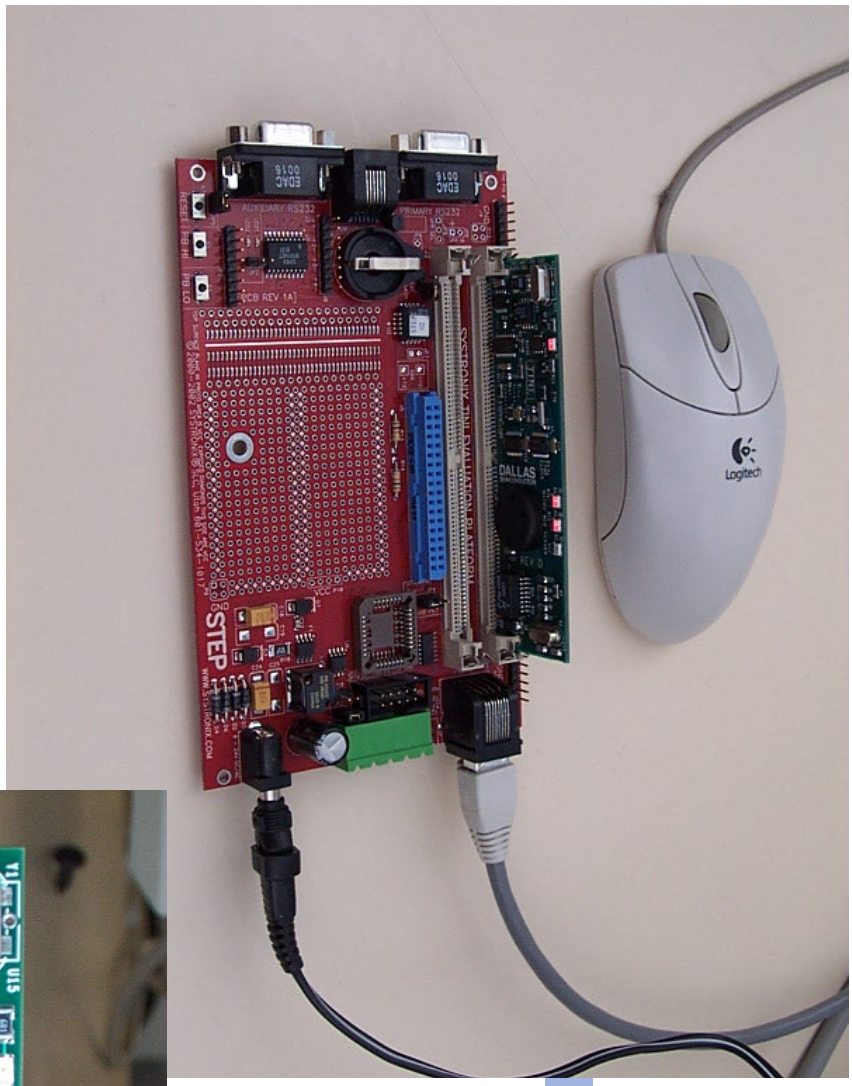


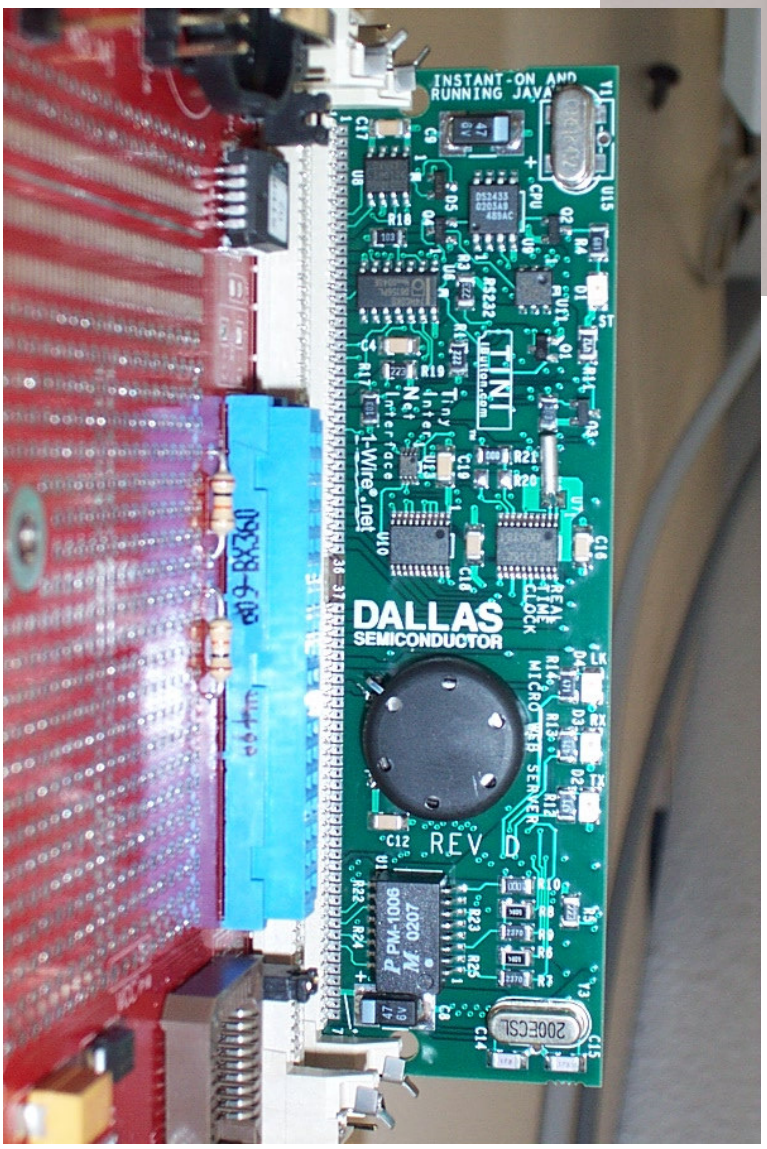
Carte Socket

- Apporte les connecteurs à la TMB390
  - Permet de tester avant intégration de la TMB sur des cartes spécifiques (custom)
- Produits
  - E10, E20, STEP, Taylec TutorIO board, ADON, NetMaster (Elsist) …

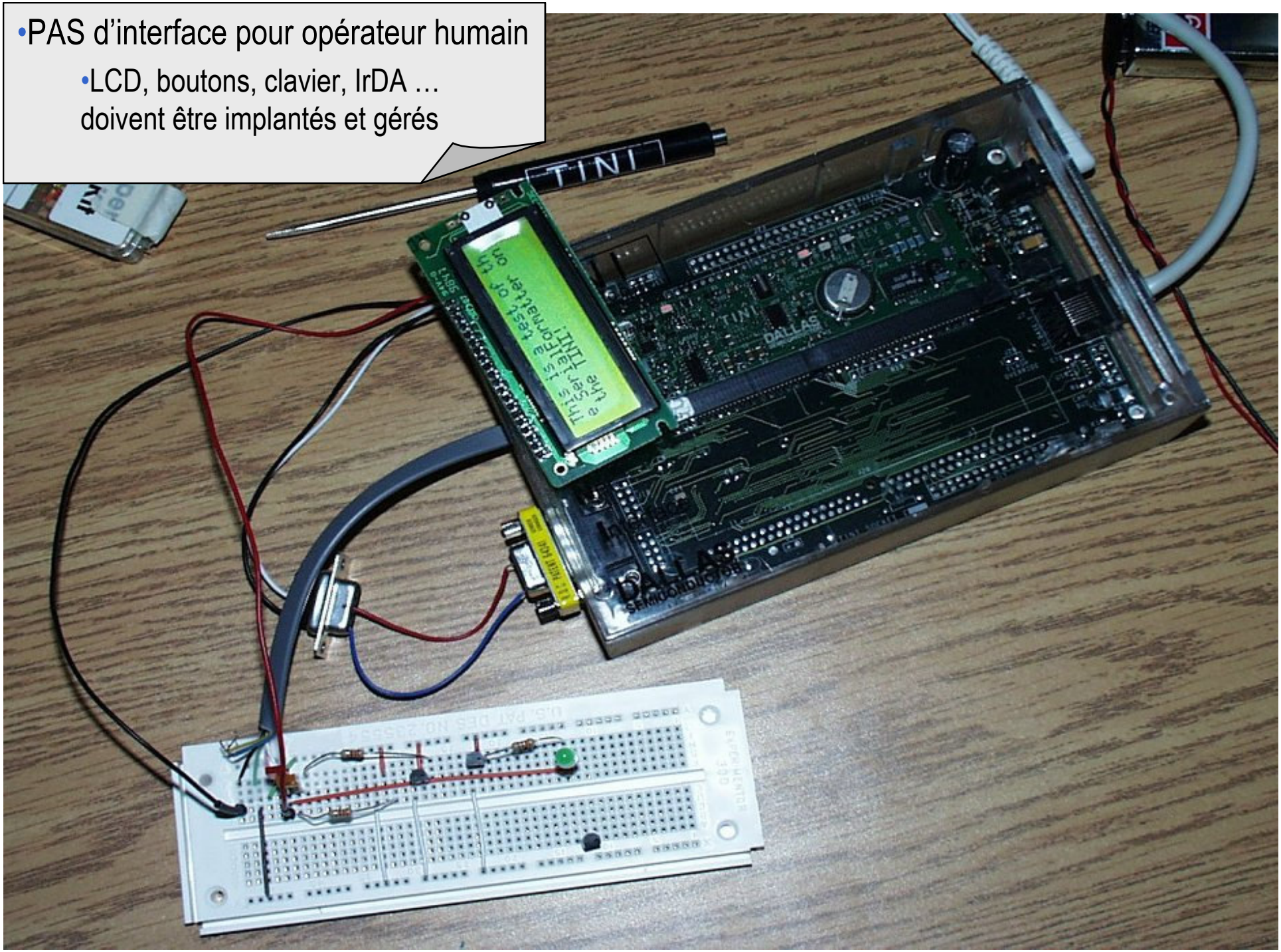


- PAS d'interface pour opérateur humain
  - LCD, boutons, clavier … doivent être implantés et gérés


STEP
DALLAS
Logitech

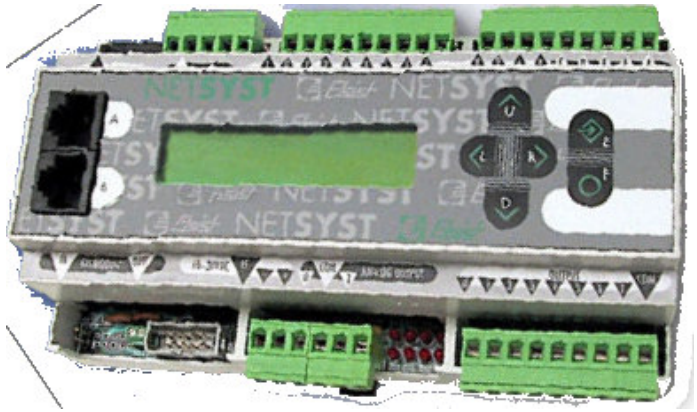

INSTANT-ON AND
RUNNING JAVA
TINI
DALLAS
SEMICONDUCTOR
REV D

- PAS d'interface pour opérateur humain
  - LCD, boutons, clavier, IrDA …
    doivent être implantés et gérés

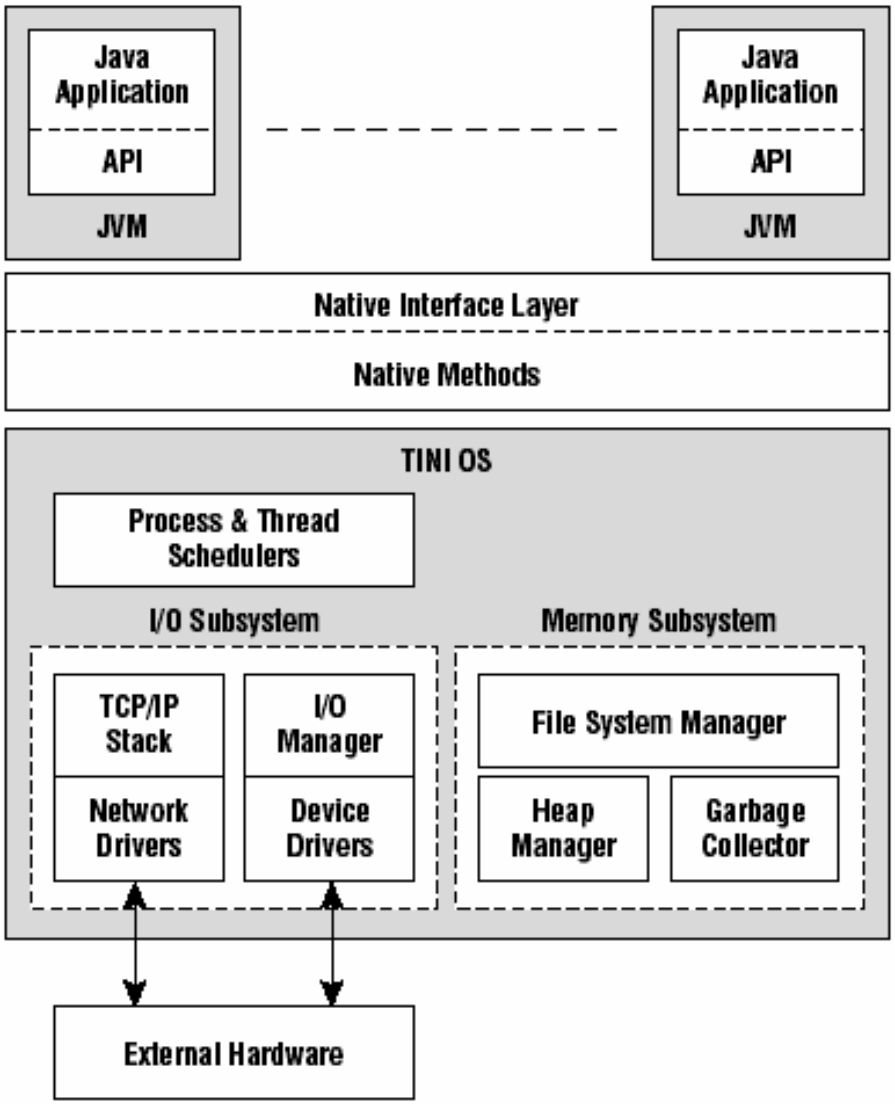


Ou conditionnée

- Exemple : NetMaster (Elsist)



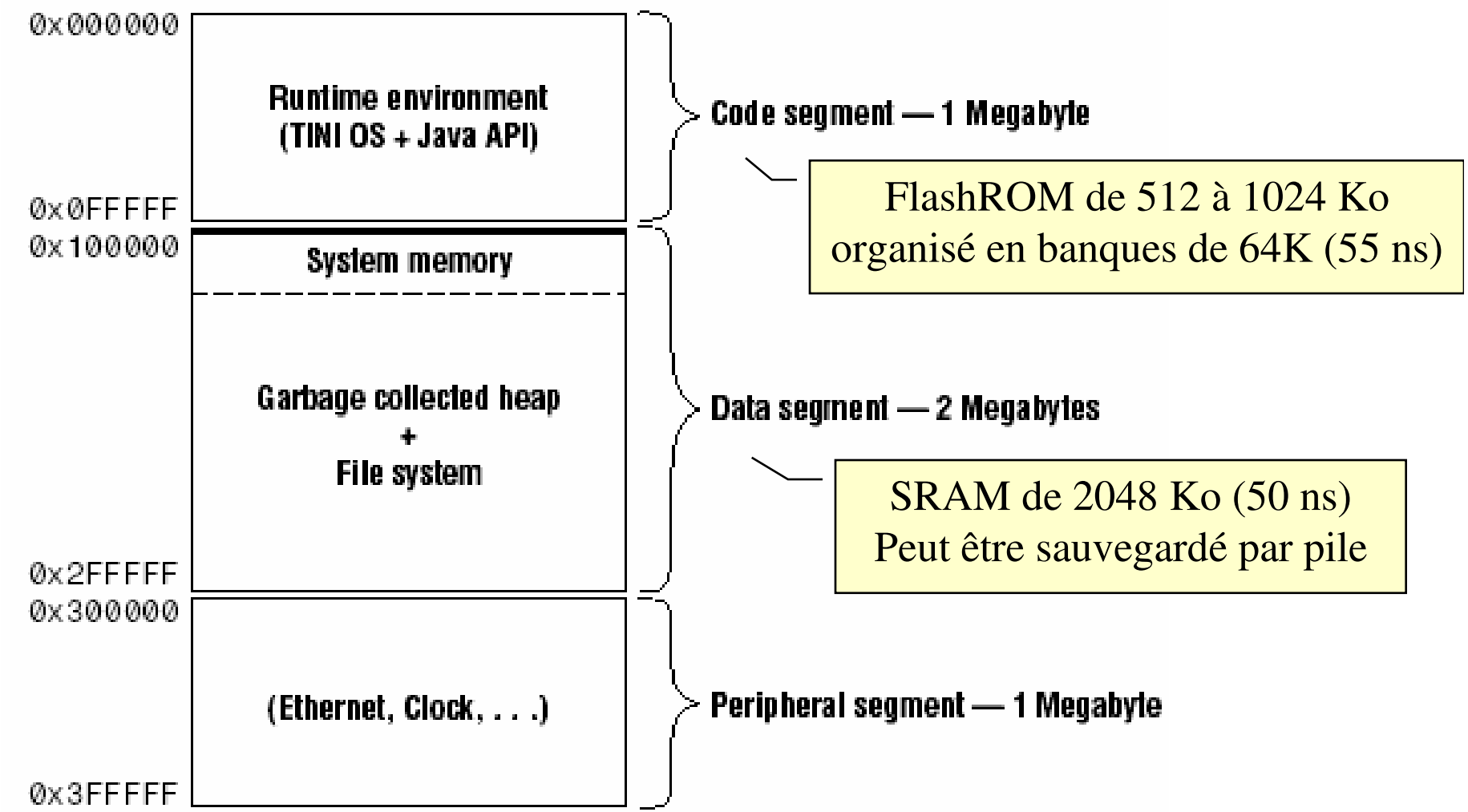
Architecture



Gestion du processus et des threads

- 2 types
  - Processus (représente une application)
  - Threads
- 2 ordonnateurs (scheluders)
  - Ordonnaceur Thread
    - Tourniquet (round robin)
    - toutes les 2 ms
    - préemptif et collaboratif `java.lang.Thread.yield()`
  - Processus
    - Tourniquet (round robin)
    - toutes les 8 ms
    - Préemptif
- Synchronisation
  - Pas d'IPC pour les processus
    - Remarque : Il est possible d'utiliser les sockets sur l'adresse local (loopback) ou des fichiers
  - Primitives de synchronisation Java pour les threads

Organisation de l'espace d'adressage



Gestion de la mémoire (RAM)

- Allocation partagée de la RAM
  - pour le tas (heap) des process Java et système
  - pour les fichiers
- Glaneur de cellules (Garbage collector)
  - Déclenchement
    - Explicitement par un process Java java.lang.System.gc()
    - Par le new lorsque la mémoire totale libre descend en dessous de 64Ko
    - A la terminaison d'un process Java
  - Portée
    - Seulement pour les objets du tas du process qui le déclenche
    - Algorithme mark-and-sweep donc bloquant pour le process (non temps réel)
- Remarque
  - La RAM peut être auto-alimentée par un circuit spéciale et une pile lithium pouvant durer jusqu'à 10 ans
  - Implique un GC au reboot en crash

Système de fichiers

- Hiérarchie de répertoires et de fichiers
  - Permission (rwx)
  - Owner/Group
  - Pas de lien
- Allocation de bloc
  - en bloc de 512 octets de RAM auto-alimentée
  - Temps d'accès constant (vitesse RAM)
  - non contigus en mémoire
    - Pas d'inconvénients sur le temps d'accès lors de la défragmentation
- Remarque
  - L'API permet le montage de systèmes de fichiers externes (NFS, FTP) `com.dalsemi.fs.FileSystemDriver`
    - http://pdfserv.maxim-ic.com/arpdf/AppNotes/app709.pdf

Gestion des entrées/sorties

Under Construction
En Construction

- 2 types
  - Réseau
    - Ethernet
  - Non Réseau
    - 1Wire : bus séquentiel sur 2 fils
    - CAN : bus de terrain
    - Série (gestion PPP avec un modem)
    - Parallèle
- *Remarque : Permet l'ajout de*
  - *Extension de mémoire secondaire (FlashROM, …)*
  - *Port réseau (Bluetooth, …)*
    - *http://www.dhpc.informatics.bangor.ac.uk/reports/121/dhpc-121.pdf*

Arrêt et redémarrage

- 2 types d'arrêt
  - Perte de courant
  - Chien de garde
- Conséquence
  - Arrêt des processus (heap à désallouer)
- Reset
  - POR (Power-on-Reset)
  - Externe
- Bootstrap
  - Redémarrage normal après contrôles d'intégrité
  - Rechargement en FlashROM du Runtime (tini.tbin) et de l'application primaire (par exemple slush.tbin)

slush, le langage de commande

- Interprète de commandes *à la Unix*
  - Utile lors de la phase de développement
  - Multi-utilisateur, Multi-tache
  - Multi-session : par port (série) console et par telnet (serveur telnetd)
  - Contient un serveur ftp pour le chargement des fichiers dans le système de fichiers
  - Extensible : command addc
- Commandes
  - `append arp cat cd chmod chown clear copy cp date del df dir downserver echo ftp gc genlog help history hostname ipconfig java kill ls md mkdir move mv netstat nslookup passwd ping ps pwd rd reboot rm rmdir sendmail setenv source startserver stats stopserver su touch useradd userdel wall wd who whoami`
- Ligne de commande
  - Redirection des entrées-sorties standards
    - `java MyApp.tini > /res/out.txt`
  - Lancement en arrière-plan (background) &
    - `java MyApp.tini &`
- Remarque : slush est une application Java

slush, exemple de session (i)

```
----> TINI Boot <----
TINI OS 1.02
API Version 8009
Copyright (C) 1999 - 2001 Dallas Semiconductor Corporation
Hit any key to login.
After pressing a key, slush prompts the user for a login name.
Welcome to slush. (Version 1.02)
TINI login: root
TINI password:
TINI /> ls -l
total 2
drwxr-x 1 root admin 1 Jan 27 15:13 .
drwxr-x 1 root admin 3 Jan 27 15:14 etc
TINI>
TINI /> cd etc
TINI /etc> ls -l
total 5
drwxr-x 1 root admin 3 Jan 27 15:14 .
drwxr-x 1 root admin 1 Jan 27 15:13 ..
-rwxr-- 1 root admin 28 Jan 27 15:14 .tininet
-rwx--- 1 root admin 225 Jan 27 15:14 .startup
-rwxr-- 1 root admin 101 Jan 27 15:14 passwd
TINI /etc>
```

slush, exemple de session (ii)

```
TINI /> cd web
TINI /web> java simpleweb.tini 3128 /web/html &
TINI /web> ps
3 processes
1: Java GC (Owner root)
2: init (Owner root)
14: simpleweb.tini (Owner root)
TINI /web> kill 14
TINI /web> ps
2 processes
1: Java GC (Owner root)
2: init (Owner root)
TINI /web>
```

slush, configuration IP

```
TINI /> ipconfig -g 192.168.0.1 -p 192.168.0.2
Warning: This will disconnect any connected network users
and reset all network servers.
OK to proceed? (Y/N): y
[ Sun Jan 28 15:02:53 GMT 2001 ] Message from System: FTP server stopped.
[ Sun Jan 28 15:03:00 GMT 2001 ] Message from System: Telnet server stopped.
[ Sun Jan 28 15:03:00 GMT 2001 ] Message from System: Telnet server started.
[ Sun Jan 28 15:03:01 GMT 2001 ] Message from System: FTP server started.
TINI /> ipconfig
Hostname : TINI.
Current IP : 192.168.0.15
Default Gateway : 192.168.0.1
Subnet Mask : 255.255.255.0
Ethernet Address : 00:60:35:00:10:bb
Primary DNS : 192.168.0.2
Secondary DNS :
DNS Timeout : 0 (ms)
DHCP Server :
DHCP Enabled : false
Mailhost :
Restore From Flash: Not Committed
TINI /> ping www.ibutton.com
Got a reply from node www.ibutton.com/198.3.123.121
Sent 1 request(s), got 1 reply(s)
```

À l'initialisation, seule l'adresse MAC est renseignée
Cette adresse est globalement unique

Développement Java

- Java
  - Threads, ThreadGroup, types primitifs et String, Sérialisation (1.1)
- Limites et Différences
  - Pas de finalisation
  - Chargement de classe (ClassLoader) spécifique

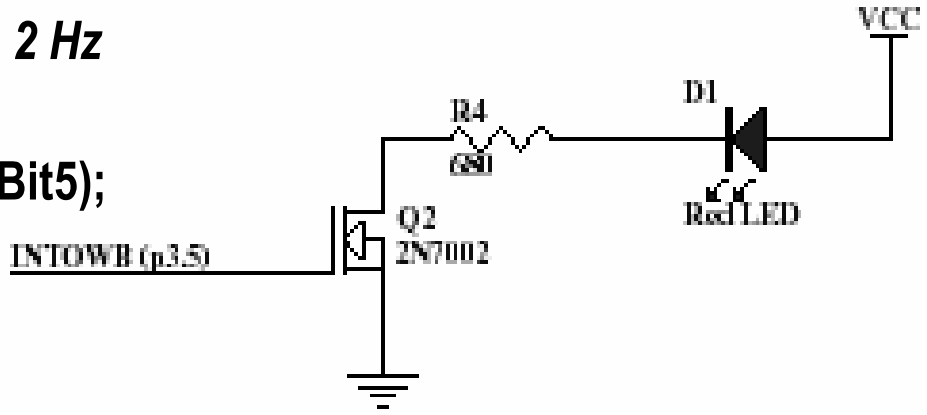
    car la résolution du constants pool sont effectués par le convertisseur lors du développement
  - Pas de vérification du bytecode
  - Pas de JIT
  - Méthodes natives
    - TNI est plus léger et flexible que JNI
    - Chargement dynamique java.lag.Runtime.loadLibrary(String libname)
  - Pas de Permissions
  - Pas de interface JVM pour le deboggage ou le profilage
- JRE
  - Core basé sur JDK1.1.8
    - java.lang, java.lang.reflect, java.io, java.net, java.util, java.io, javax.comm
    - Pas de classes orienté interface utilisateur (hormis HTTP)
  - TINI : com.dalsemi.*
- Remarque : la JVM a une empreinte mémoire de 40Ko

Développement Java

- Cycle de développement
  - Compilation des classes
    - javac –Xbootclasspath …
  - Convertisseur des classes en un fichier tini
    - Motivation : résolution des noms avant l'installation (gain temps/taille)
    - class TINIConvertor
    - Produit un fichier myapp.tini
  - Installation du myapp.tini
    - ftp 192.168.0.15
  - Exécution par la commande slush
    - telnet 192.168.0.15
    - java MyApp.tini parameters
  - Installation en application primaire à la place de slush pour la production (via le port console)

Exemple d'application Java
Contrôle d'un port

```
import com.dalsemi.system.BitPort;
class Blinky { // clignotement d'une LED à 2 Hz
  public static void main(String[] args) {
     BitPort bp = new BitPort(BitPort.Port3Bit5);
     for (;;) {
       // Turn on LED
       bp.clear();
       try { Thread.sleep(250); // Leave it on for 1/4 second
             } catch (InterruptedException ie) {}
       // Turn off LED
       bp.set();
       try { Thread.sleep(250); // Leave it on for 1/4 second
             } catch (InterruptedException ie) {}
     }
  }
}
```




```
TINI /> java Blinky.tini &
TINI /> ps
3 processes
1: Java GC (Owner root)
2: init (Owner root)
4: Blinky.tini (Owner root)
TINI /> kill 4
TINI /> ps
2 processes
1: Java GC (Owner root)
2: init (Owner root)
TINI />
```

Un serveur Web simple

http://tinadele.imag.fr:3128/index.html

```
TINI /> java simpleweb.tini 3128 /web/html &
TINI /> ps
3 processes
1: Java GC (Owner root)
2: init (Owner root)
5: simpleweb.tini (Owner root)
```

```
import com.dalsemi.tininet.http.HTTPServer;

class SimpleWebServer {
    public static void main(String[] args) {
        // Constuct an instance of HTTPServer that listens for requests port 80
        HTTPServer httpd = new HTTPServer(args[0]);
        httpd.setHTTPRoot(args[1]);
        httpd.setIndexPage("index.html");
        // Specify a name for the log file and turn on logging
        httpd.setLogFilename("log/web.log");
        httpd.setLogging(true);
        // Spin around forever servicing inbound requests
        for (;;) {    try {
            httpd.serviceRequests(); // Wait for a new request
          } catch (com.dalsemi.tininet.http.HTTPServerException e) {
                    System.out.println(e.getMessage()); }
        }
    } }
```

Un script Web simple

- Méthode POST (tini 1.10)
  - Invocation de la méthode handlePost() sur un objet de la classe dont le nom (eg monscript) est passé dans l'URL
  - Puis retour du fichier monscript.html qui contient la sortie de monscript

```
public class MonScript implements com.dalsemi.tininet.http.PostScript {
    public void handlePost(Vector data) {
      PrintStream out=null;
      try{
        String classname = "MonScript"; // this.getClass().getName();
        String filename = "/testweb/html/"+classname+".html"; // no httpServer.getHTTPRoot()
        out = new PrintStream(new FileOutputStream(filename));
      } catch (java.io.IOException ex){ return; }
      out.println("<html><body><h1>com.dalsemi.tininet.http.PostScript test</h1><hr><pre>");
      for (Enumeration e = data.elements() ; e.hasMoreElements() ;) {
                  PostElement pe=(PostElement)e.nextElement();
                  out.println(pe.field+"="+pe.value);
      }
      out.println("</pre><hr></body></html>
    }}
```

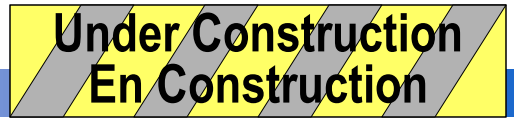
```
http://tinadele.imag.fr/MonScript
user=didier&password=toto+le%20+heros
```

Empreintes mémoire

- Capacité flash : 512Ko

- Environnement 448Ko
  - Bootstrap loader ~4Ko
  - Runtime
    - TINI OS
    - JVM 40Ko
    - Java Classes
- Application Java primaire
  - Jusqu'à 63Ko
  - Exemple : slush : 63Ko

Classes TINI



- Slush
- Réseaux IP
  - URL framework
  - DHCP, DNS, ICMP
  - Telnet, FTP, HTTP (très limité mais GET/POST)
- Modem PPP
- Entrées/Sorties Séries
- Entrées/Sorties Parallèles
- 1Wire

Temps

- Horloge temps réel (RTC: real-time clock)
  - Sauvée par pile (lithium)
  - resynchronisable avec un serveur de temps (RFC868)

Chien de garde (watchdog)

- Principe
  - Le système et l'application primaire peut partir dans le décor (runaway)
  - Causes:
    - Terminaison de Thread par une exception non manipulée, Threads interbloqués
    - Crash de l'OS, panne matérielle transitoire (décharge électrostatique …)
  - Le chien de garde force à rebooter TINI s'il n'est pas « nourri » par l'application primaire (vivante)

- API TINI
  - Méthodes setWatchdogTimeout(int timeout) et feedWatchdog() de com.dalsemi.system.TINIOS
  - commande wd de Slush

Sécurité

- Postulat
  - Toute machine connectée à un réseau est susceptible d'être attaquée (hacked)
- Risque
  - Interruption de service
    - reinstallation d'une carte embarquée et éloigné peut prendre des semaines
  - Corruption d'acquisition de mesure, …
- Conseils
  - Vérifiez que les serveurs soient protégés contre les attaques
  - Minimisez les services lancés
  - N'utilisez pas les logins/passwords (e.g. root/tini)
  - Chiffrez l'information qui transite par le réseau
    - telnet/ftp au dessus de SSH
    - https:// au lieu de http://

Des outils pour la sécurité

- Rien en standard
- Cryptographie
  - JCE light pour J2ME
    - Voir cours JCE
  - kSSL : SSL pour KVM
  - SSL for TINI
    - http://security.dstc.edu.au/projects/java/ssl_tini.html
  - iButton (DS1954) orienté Cryptographie
    - eOCF et iButton
- Remarque:
  - Vous pouvez utilisez des méthodes natives (TNI) pour les principales fonctions crypto (SHA1,MD5, DES, RSA, EC, …) en cas de problème de performance.

Communications

- HTTP Server
  - TINIHttpServer, BrazilTINI, Servertec …
    - servlet 2.2, session tracking, authentification, logs régulièrement envoyé par mail…
    - http://www.smartsc.com/tini/index.html
    - http://www.servertec.com/products/tini_iws/tini_iws.html
- JMS
  - MindStream Software
    - (http://www.mindstrm.com/products/message)
  - Scalagent kJORAM ???
- RMI
  - Aucun pour l'instant ?
  - La sérialisation a été introduit dans la version 1.1
  - Une piste : Objectweb Jonathan

Communications

- XML et SOAP
  - kXML et kSOAP à tester ?
- P2P Protocols (JXTA) on TINI
  - http://tini.jxta.org/
    - *The TINI board gives you, essentially, a Java virtual machine on a stick. While that virtual machine has more features than other small Java devices that run J2ME, it does suffer from many of the same limitations, most notably limited memory.*
    - *One way to bring the power of JXTA peer-to-peer networking to the TINI is to use a proxy approach, wherein only a limited subset of JXTA function resides on the TINI, which in turn relies on an outside system to do the heavy-duty JXTA communications.*

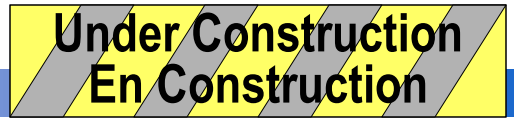
Déploiement

- OSGi
  - Aucun pour l'instant

Chargeurs de Classes

- Format .tini
  - Format « Just-In-Place » : peu ou pas de transformation pour être utilisé par la VM
  - .tini regroupe des .class convertis et résolus
    - Le convertisseur {.class,.tlib}→.tini effectue une édition de lien anticipée et une fermeture de référence de classes
- Chargement standard
  - Charge un .tini
- ClassLoader
  - Charge et résout un .class depuis le répertoire /tiniext
  - Exemple : XXFileSystemDriver

Format .tini



Méthodes et Bibliothèques natives

- Motivations
  - Accélérer certains algorithmes consommateur en CPU (eg. crypto)
  - Pilote matériel, …
- Outils
  - `macro.exe` : MacroAssembleur `.ext` → `.mpp`
  - `a390.exe` : Assembleur `.mpp` → `.tlib`
  - `TiniConvertor`: inclut STATIQUEMENT la .tlib dans le .tini
    - java TINIConvertor -n des.tlib -f Decrypt.class -d tini.db -o decrypt.tini
- Remarque
  - Un programme Java peut aussi charger DYNAMIQUEMENT une .tlib présente dans le FileSystem
- Limites
  - Le toolkit ne comporte pas de compilateur C
    - ∃ Cross-Compilateur C → 8051,DS390 (SDCC, http://sdcc.sourceforge.net/)
  - Pas de callback (i.e. méthode native appelant une méthode java)

Méthodes et Bibliothèques natives

- Limites
  - Le toolkit ne comporte pas de compilateur C
  - Pas de callback (i.e. méthode native appelant une méthode java)
- Cross Compilateurs tiers
  - Keil uVision2 suite tools : compilateur C
  - SDCC, http://sdcc.sourceforge.net/

Méthodes et Bibliothèques natives avec SDCC

```
// Hello.java
import com.dalsemi.comm.*; import com.dalsemi.system.*;
public class Hello {
 public static native int method1(int i,int j);
 static void main(String args[]) {
  System.out.println("Hello Started");
  try {
      System.loadLibrary("myn.tlib");
      System.out.println("Load Success");
      System.out.println("Native method1 returned "
                + method1(200,100));
  }  catch (Throwable t) { System.out.println(t);}}}
```

```
/* myn.c */
long Native_method1() _JavaNative{
    long l = NatLib_LoadInt(0);
    long k = NatLib_LoadInt(1);
    return l-k;
}
```

```
> javac -bootclasspath %TINILIB%\tiniclasses.jar Hello.java
> java -cp %TINILIB%\ tini.jar TINIConvertor
-f Hello.class -o Hello.tini -d
%TINILIB%\ tini.db
> sdcc -mTININative myn.c
Load Hello.tini & myn.tlib into the TINI board
```

```
TINI /> java Hello.tini
Hello Started
Load Success
Native method1 returned 100
```

Outil de Développement (i)

- SDK (Téléchargeable gratuitement)
  - Convertisseur
  - Macro/Assembleur
  - Constructeur de dependance
  - TTY pour le port série (JavaKit)
    - utilisation au chargement en Flash de l'OS + Application principale

Outil de Développement (ii) Compléments

- Taches ANT
  - http://tiniant.sourceforge.net/
  - complété des taches FTP, Telnet, … pour le déploiement
- Tests
  - JUNIT, …
- Ofuscateur (voir celui du J2MEWTK)
  - Réduit le ConstantPool des .class
- Optimisation de l'allocation mémoire
  - OptimizeIt, …
- TTY
  - http://www.turbobit.com/tini.htm, TiniTalk, TiniInstaller, …
- Editeur de fichiers
  - http://www.smartsc.com/tini/index.html

Conseil pour le Développement

- Environnement contraint
  - Voir les conseils du cours sur J2ME

Références

- Book
  - Don Loomis, « The TINI™ specification and developer's guide », June 2001, ISBN 0-201-72218-6, www.awl.com/cseng/
    - et téléchargeable librement sur le site TINI
    - http://www.maxim-ic.com/products/tini/pdfs/tinispec.pdf
  - Getting Started with TINI
    - http://www.maxim-ic.com/products/tini/pdfs/TINI_GUIDE.pdf
- Web site (www.ibutton.com/TINI)
  - http://www.ibutton.com/TINI/developers/community.html
  - http://www.smartsc.com/tini/index.html
    - Des informations et des applications pour TINI
  - http://www.jguru.com/faq/home.jsp?topic=TINI
  - http://www.apms.com.au/tini
- Forum
  - http://lists.dalsemi.com/maillists/tini/

Références

- Autres
  - Tutorials et Tips de Systronix
    - http://www.systronix.com/tutor/top.htm
  - Un cours en anglais
    - http://latitude.east.asu.edu/tini/

SNAP *Simple Network Application Platform*

http://www.imsys.se

- SNAP
  - network-ready, Java-powered plug & play reference platform.
- Architecture
  - µC
  - Format SIMM
- Applications
  - remote control, data processing and managing of everything from small sensors to advanced surveillance factory equipment.
- Runtime
  - OS ???
  - J2ME-CLDC certified by Sun Microsystems
- Platines
  - Celles de la TINI