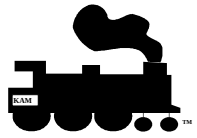
Jacobsen Declaration Exhibit AG

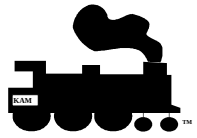
Computer Interface Application Programming

Matt Katzer
KAM Industries
Portland, Or.



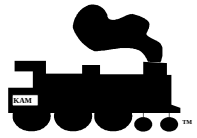
Agenda

- **NMRA software application model**
- **Philosophy / development hardware**
- **Command control software available**
- **Third party products**
  - Object oriented, user extensible (new model)
  - monolithic programs
- **Writing an application (VB, Java, C/C++)**
  - Using propose NMRA API (Train Tools® interface) in VB
  - Using NMRA TP 9-2.4 (roll your own) in C
- **Questions/Answers**



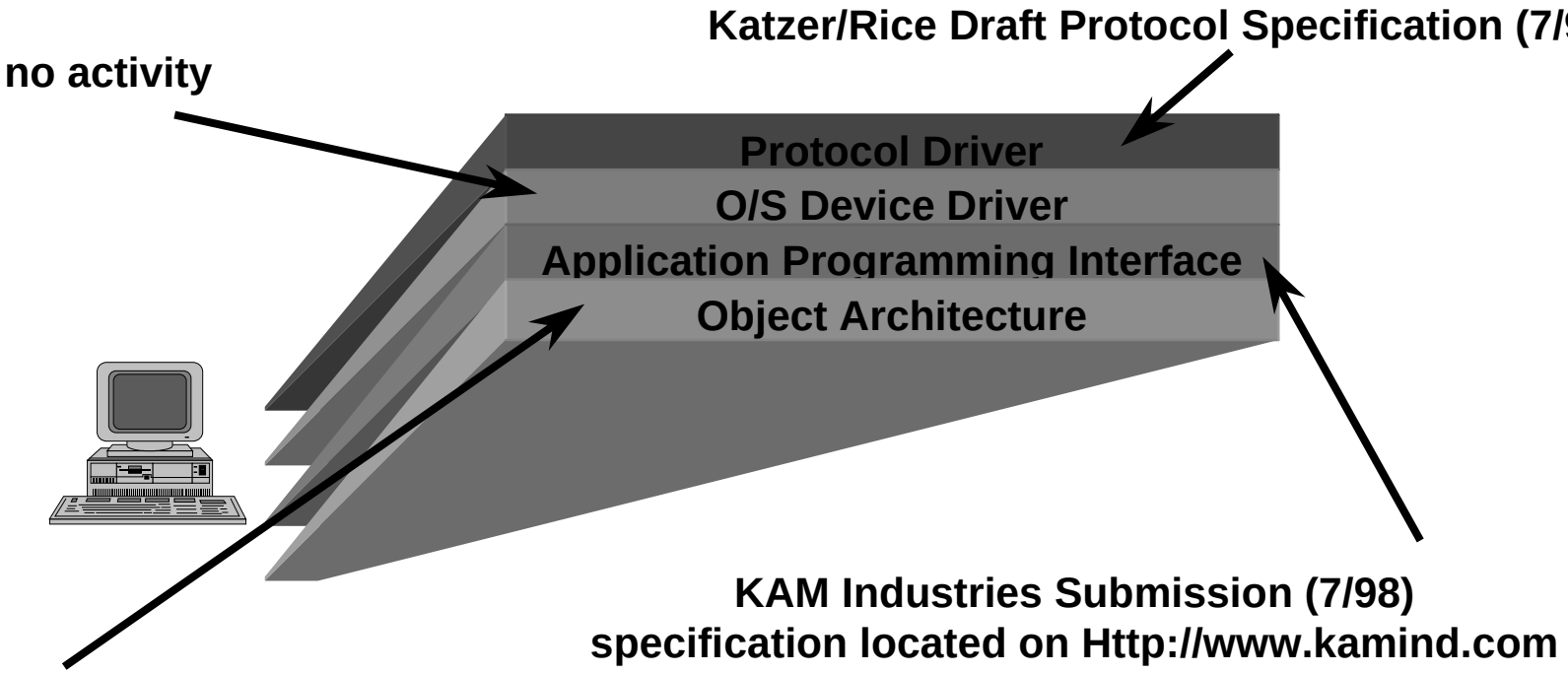
Why are you here

- **Clinic will provide a status update to the NMRA software application model**
- **Clinic will focus on writing programs to control your railroad….**
  - we will talk about PC's
  - programming languages
  - example programs
- **Clinic subject are focus on software direction from the NMRA DCC working group**
- **What are your expectations?**



Status of NMRA Application S/W Architecture Model

- **There are four parts to the NMRA DCC software architecture model**



Status of NMRA Application S/W Architecture Model (cont.)

- **Protocol Level**
  - **hardware Products**
    - » North Coast Engineering, Wangrow Electronics
    - » Easy DCC
    - » ZTC systems
  - **Software drivers for command station hardware**
    - » WinLok, Engine Commander®, Railroad Company Tayden Design
  - **Generic draft protocol driver**
    - » Engine Commander®

Status of NMRA Application S/W Architecture Model (cont.)

- **Device Driver Level**
  - » no activity
- **Application Interface Level**
  - – **hardware Products**
    - » not applicable to hardware
  - – **Microsoft COM/DCOM implementation of API**
    - » Engine Commander®
    - » Generic type library available for linking with application written in Java, Visual Basic, C/C++
  - – **CORBA support**
    - » no activity

Status of NMRA Application S/W Architecture Model (cont.)

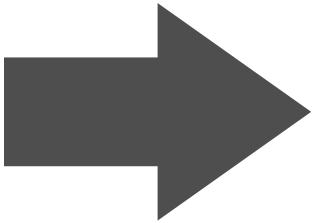
- **Object level**
  - **Rosa application model proposed (update on http://www.digi-toys.com)**
  - **hardware Products**
    - » not applicable to hardware
  - **Software products**
    - » Engine Commander® and Train Server® conforms in architecture model
  - **COM support**
    - » no activity
  - **CORBA support**
    - » no activity

Agenda

- **NMRA software application model**
- **Philosophy / development hardware**
- **Command control software available**
- **Third party products**
  - Object oriented, user extensible (new model)
  - monolithic programs
- **Writing an application (VB, Java, C/C++)**
  - Using propose NMRA API (Train Tools® interface) in VB
  - Using NMRA TP 9-2.4 (roll your own) in C
- **Questions/Answers**

Philosophy

- **Computer Controlled**
  - The computer controls the routes of the trains
  - The operator runs his/her layout from the computer
- **Computer Monitored**
  - The computer is a tool of the modeler
  - The computer is used to manage events
  - The computer does not control!
  - The computer programs decoders (because I can't remember the key sequence!)



I like to write software, but I want to run trains and use computers monitor the layout and to enhance the fun

Hardware Requirements

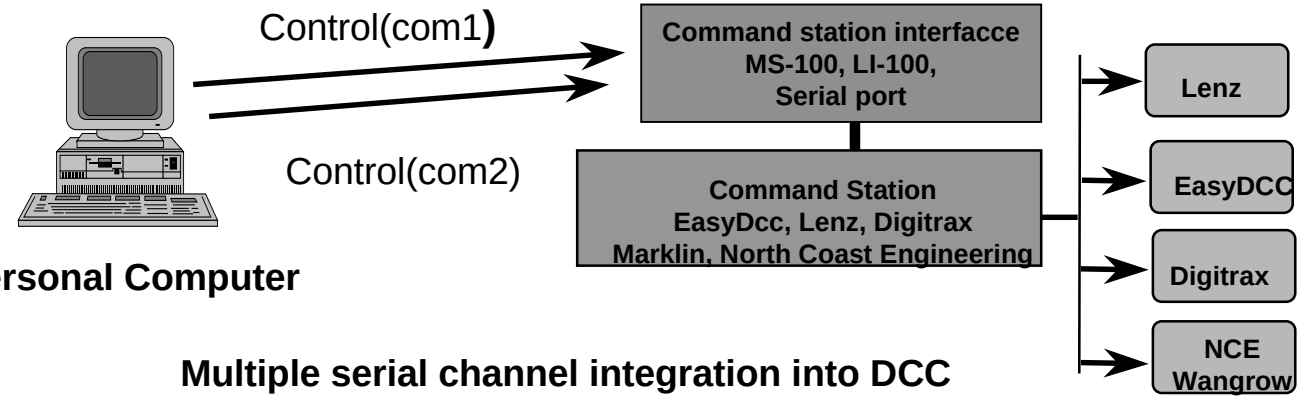
- **What type of PC hardware should you buy?**
  - **depends on what you are doing..**
    - » **Development:**
      - Big fast disk(4gbyte, 8-10 msec access time, 512K cache)
      - Iomega Jazz (1Gyte) drive for backup
      - lots of memory (64meg at 100 Mhz ~ $62)
      - Pentium II ; 300 Mhz, BX series motherboard
    - » **Operation:**
      - The PC must run Windows 95
      - 16 mbytes of memory
    - » **Command Station support**
      - One command station does not require new serial ports
      - Multiple command stations (like a programmer and and controller); you must purchase a smart serial card (MaxSpeed, Consensys etc, with Windows 95 drivers)
      - Don't waste money on a dumb serial card for COM3/COM4, these don't work.

Hardware Requirements(cont.)

- **What you should not buy**
  - 286, 386 or 486 PC's (ROT: must run Win 95/98)
  - systems that contain less then 8 Mbytes
  - systems with drives less then 512 mbytes
  - system must have a CDROM; All new windows 95/98 software will not install from floppy. it is no longer profitable to make diskettes.
  - Apples; what you have in software is all you will have
  - Macs: unless connected to a network(client/server model), most like advance software will not be available (it is cheaper for mfg's to ship a PC then it is to develop for a Mac.
- **You must have internet access!**
  - Driver updates are located on the web
  - Software products will require web registration for update files.

Railroad Requirements

- **Must have NMRA DCC compatible engines**
  - Pick a DCC supplier based on current required for your locomotive
  - By 2000, all locomotives in a price range above $100 will most likely have a decoder integrated into the unit
- **Command station equipment**
  - Expect a hybrid; plan for multiple command stations on layout
  - Model expected; one for programming the other for control



Multiple serial channel integration into DCC

Agenda

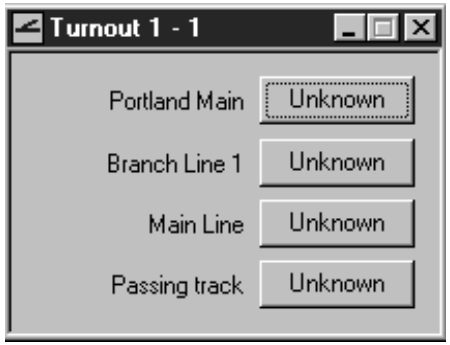
- **NMRA software application model**
- **Philosophy / development hardware**
- **Command control software available**
- **Third party products**
  - Object oriented, user extensible (new model)
  - monolithic programs
- **Writing an application (VB, Java, C/C++)**
  - Using propose NMRA API (Train Tools® interface) in VB
  - Using NMRA Draft Protocol Specification (roll your own) in C
- **Questions/Answers**

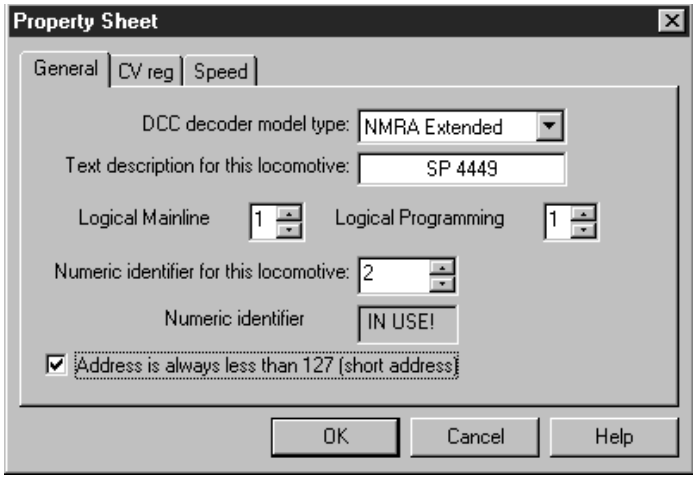
Types of DCC Software

- **Two styles of program in market today**
- **Monolithic style**(Self contained)
  - MS-DOS examples
    - » Marklin shareware apps
    - » John Kabat Quick basic packet driver
    - » Digipert/Digiplus II
  - Microsoft Windows 95/98 and Windows NT
    - » WinLok, Tayden design, Real Railroad
- **Modular style** (object structure, user extensible)
  - Microsoft Windows 95/98 and Windows NT
    - » Engine Commander®
    - » and soon others as well

Engine Commander®

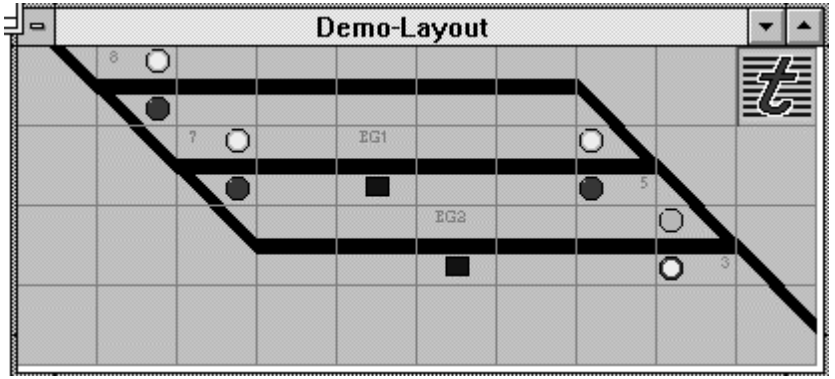
- **Built on a modular philosophy**
  - implements NMRA TP/RP whenever possible
  - Users can write their own programs(COM MS application)
  - simulation interface include for development
- **Loco, Switch and Sensor Feedback**
  - Asynchronous feedback support for state changes
  - Full decoder control and programming





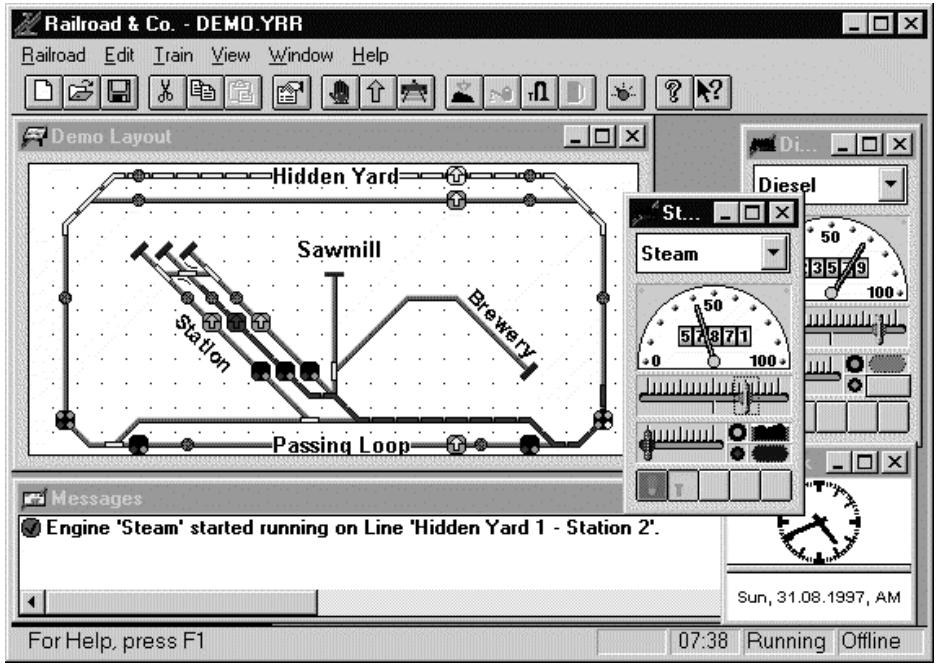
WinLok

- **Supports DCC command stations**
  - draws from the German railways operation
  - supports visual layout display
  - multiple user throttles
  - integrated acceleration curves
- **European design/tradeoffs**



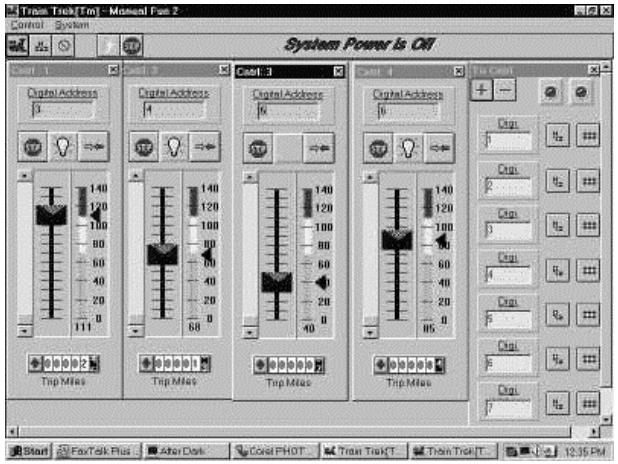
Real Railroad

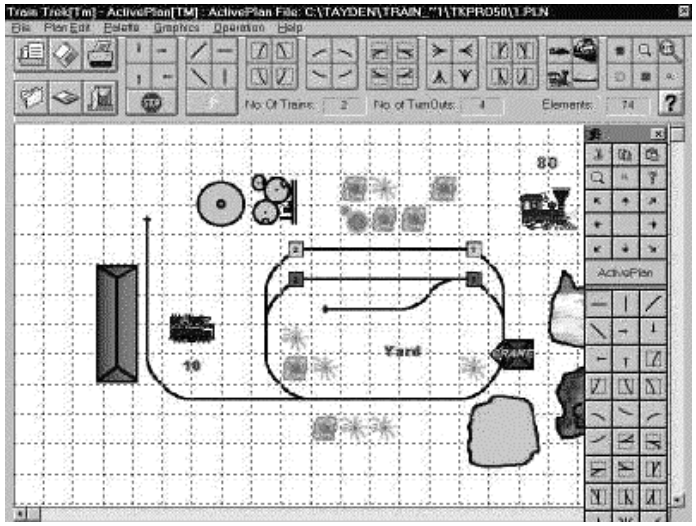
- **Windows 3.1 implmentation**
- **Supports most popular DCC systems**
- http://www.he.net/~freiwald/pages/railco.htm



Tayden Design

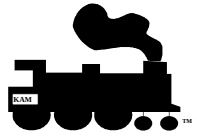
- **Train trek version 5.0**
  - **A lot of nice DCC integration**
  - **Different model of locomotive control**
  - **http://www.tayden.com**





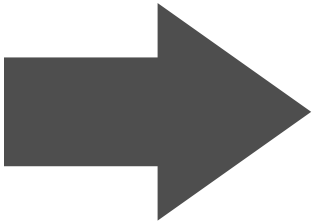
Agenda

- **NMRA software application model**
- **Philosophy / development hardware**
- **Command control software available**
- **Third party products**
  - **Object oriented, user extensible** (new model)
  - **monolithic programs**
- **Writing an application (VB, Java, C/C++)**
  - **Using propose NMRA API (Train Tools® interface) in VB**
  - **Using NMRA TP 9-2.4 (roll your own) in C**
- **Questions/Answers**



Should I write my own programs?

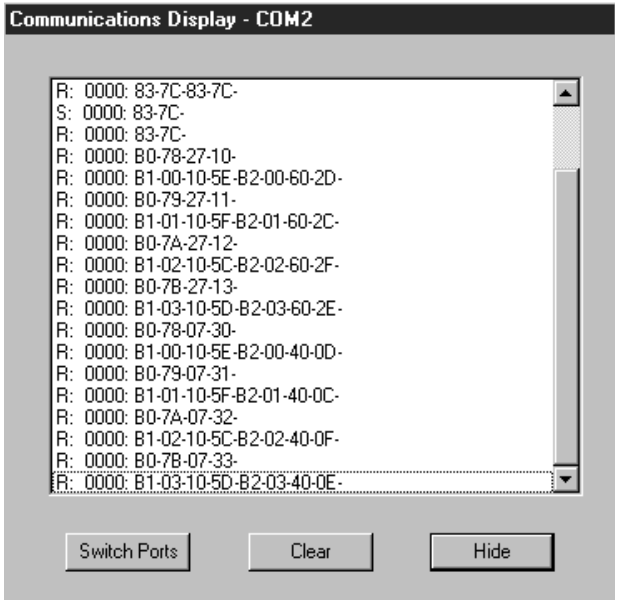
- **Are you ready to**
  - **read the protocol specification to the controller**
  - **Decider on which level do you want to write to**
  - **Pick your application language**
  - **and answer the following question?**
    do you want spend many hours away from you layout
    - ... have fun programming?
- **What Language do you use?**
  - **novice: Java or visual basic**
  - **experienced: C or Pascal**
  - **advanced: C++ under windows**



Understand where you want to put your energy to maximize your fun!

What is the best way to begin?

- **First understand the protocol and interface**
- **Second follow these rules**
  - **keep it simple…**
  - **design the architecture…**
  - **build the infrastructure…**
  - **Use a program with a debug display**



- **Best way to begin..**
  - **buy the correct PC and tools**
  - **if you are a novice used Visual Basic 5 or Java**
  - **if you are advance user, use Visual C++**

Remember, Rome was not built in a day!

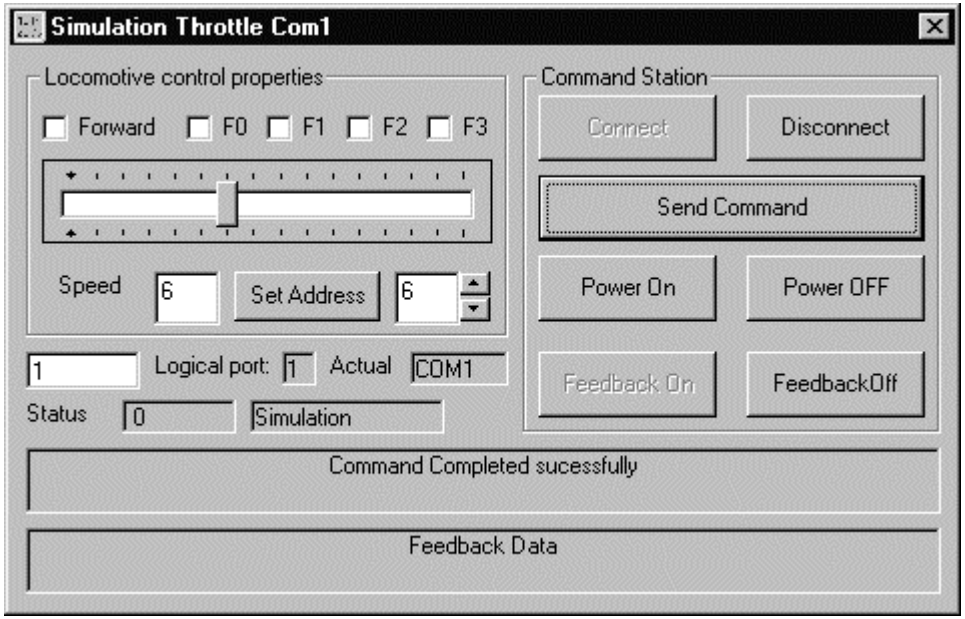
What is the best way to begin?

- **Follow these four steps**
  - **Acquire a 3rd party app for experimentation**
  - **Design your user interface (use GUI tool)**
    - » **Pick either NMRA API or NMRA protocol driver**
  - **Now implement small features**
  - **Add functionality as you desire**
- **Lets walk through these four steps...**

Remember, Rome was not built in a day!

Acquire a 3rd party Application or build it

- **How is this Visual Basic application built?**



- **Lets look at how you program it**

Visual Basic 5 Train Tools®

- **First step is to add the define the object**

```
' This first command adds the reference to the TrainTools Interface object
Dim EngCmd As New EngComIfc
'
' Engine Commander uses the term Ports, Devices and Controllers
'  Ports ->   These are logical ids where Decoders are assigned to.  Train Tools
'             Interface supports a limited number of logical ports.  You can
'             also think of ports as mapping to a command station type.  This
'             allows you to move decoders between command station without
'             loosing any information about the decoder
'
' Devices ->  These are communciations channels configured in your computer.
'             You may have a single  device (com1) or mutiple devices
'             (COM 1 - COM8, LPT1, Other).  You are required to map a port to
'             a device to access a command station.  Devices start from
'             ID 0 -> max id (FYI; devices do not necessarly have to be
'             serial channel.  Always check the name of the device before you use
'             it as well as the maximum number of devices supported.
'             The Command
'                 EngCmd.KamPortGetMaxPhysical(lMaxPhysical, lSerial, lParallel)
'             provides means that... lMaxPhysical = lSerial + lParallel + lOther
'
' Controler - These are command the command station like LENZ, Digitrax
'             Northcoast, EasyDCC, marklin...  It is recommend that
'             you check the command station ID before you use it.
'
' Errors    - All commands return an error status.  If the error value is
'             non zero, then the other return areguments are invalid.  In
'             general, non zero errors means command was not executed.  To
'             get the error message, you need to call KamMiscErrorMessage
'             adn supply the error number
'
' To Operate your layout you will need to perform a mapping between
```

Visual Basic 5 (cont.)

- **next,**
  - **Write the subroutine to control the loco**

```
'*******************************
'   Send Command
' Note:
'    Load the state of the decoder first, then send the command
'*******************************
Private Sub Command_Click()
    'Send the command from the interface to the command station, use the engineObject
    Dim iError, iSpeed As Integer
    If Not Connect.Enabled Then
        ' TrainTools interface is a caching interface.  This means that you need to set
        ' the CV's or other operations first; then execute the command.
        iSpeed = Speed.Text
        iError = EngCmd.DccEngSetFunction(lEngineObject, 0, F0.Value)
        iError = EngCmd.DccEngSetFunction(lEngineObject, 1, F1.Value)
        iError = EngCmd.DccEngSetFunction(lEngineObject, 2, F2.Value)
        iError = EngCmd.DccEngSetFunction(lEngineObject, 3, F3.Value)
        iError = EngCmd.DccEngSetSpeed(lEngineObject, iSpeed, Direction.Value)
        If iError = 0 Then iError = EngCmd.DccCmdCommand(lEngineObject)
        SetError (iError)
        End If

End Sub
```

Now Implement small features

- **Make the engine go!**
  - (run the program)
- **Now add additional functions**
  - Complete some of the button objects
  - Add additional commands
    - » (use the samples as a reference)
- **Now lets look at a C implementation of the protocol**

Remember, Rome was not built in a day!

C Example using NMRA Draft Protocol Specifcation

- case ENGINE:

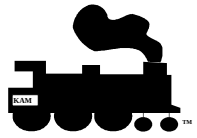
```
    {
     char szDCCData[10];    // DCC data structur

     // Build the command stream to be inserted...
    int iSize = 0;
    int iRetAddress, iDccIndx, iOffset, iCnt;
    long*  lplCookie;
    long   lDccCookie;

   // get the decoder info
   lplCookie = (long*)(lpSrcData + ENGADDR_COOKIE_LONG);
   lDccCookie = *lplCookie;
   iDccAddress = ConvertDccCookie(&lDccCookie);

  // Now build the speed control to be sent to the interface...
  SetNCESpeedMode(lpQueAdj, lDccCookie, iDccAddress, byCmd);        /Sets the slot Speed type 128 steps, 14 steps...
  iRetAddress = DccEngine(lpSrcData, szDCCData, &iSize);
  BuildDCCChecksum(szDCCData, iSize);
  iCnt = iSize + 1;
  iOffset = 1;
  iDccIndx = 0;

  // Now build the command packet for the Northcoast controllers
  lpDstBufData[0] = NCE_QUEUEDCC;
  while (iCnt > 0)
                {
                lpDstBufData[iOffset++] = NCE_ASCIISPACE;
                PutDataAsAsciiHexByte(lpDstBufData + iOffset, szDCCData[iDccIndx++]);
                iOffset = iOffset + 2;
                iCnt--;
                }
lpDstBufData[iOffset++] = NCE_SYSCOMMAND;
 iDataSize = iOffset;
 }
 break;
```

Still straight forward, except that you need to take more of an object view. This sample is a collection of software routines to implement the same engine functions in the previous slides.

Sample DCC Packet Generated: Q 13 34 35

Where to from here?

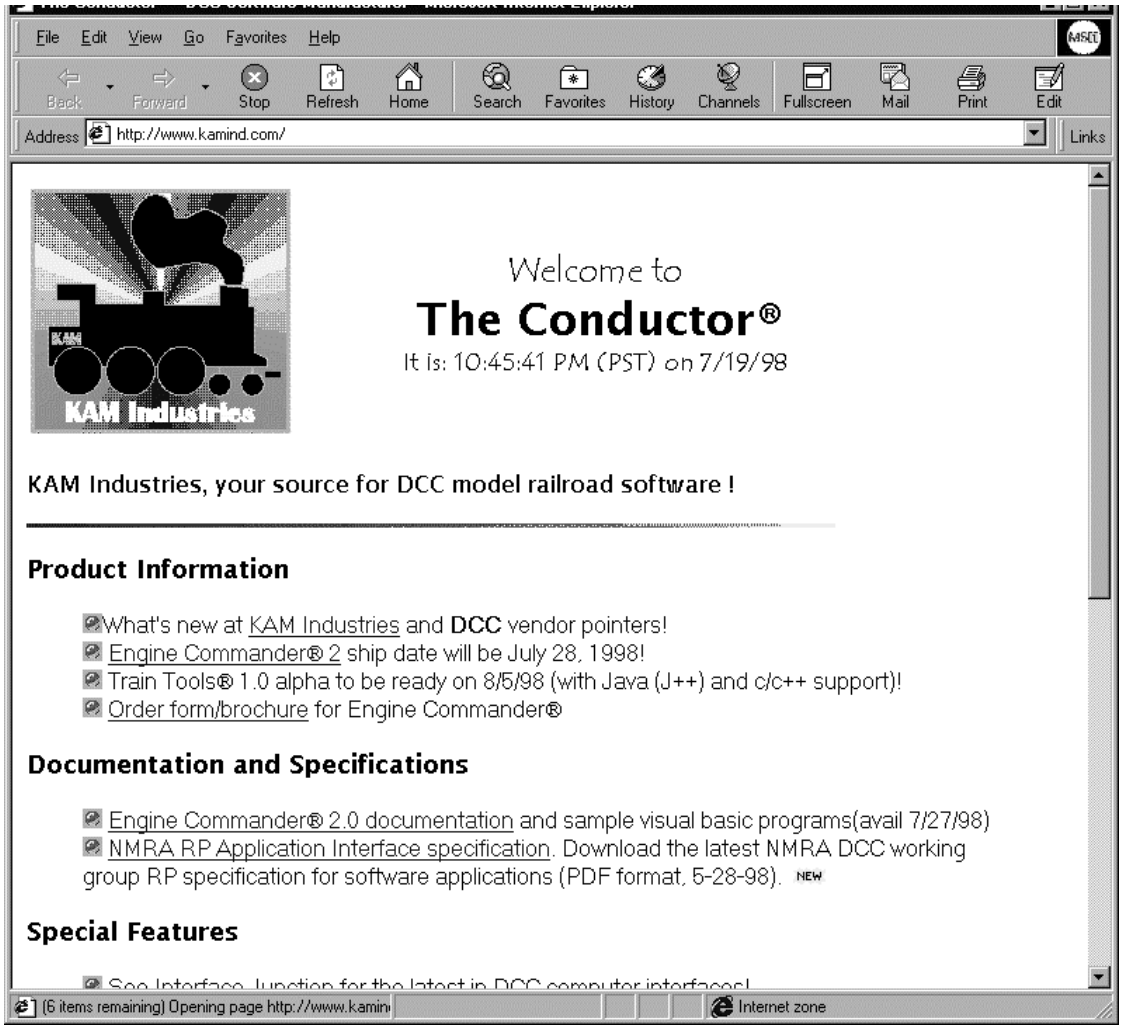
- You need to decide which direction you want to go
  - Download the API from the http://www.kamind.com
  - Download the serial protocol specifcation from KAM or one of the Command station vendors
  - Experiment with you design
  - Acquire a 3rd party app for experimentation
  - Design your user interface (use GUI tool)
  - Now implement small features
  - Add functionality as you desire



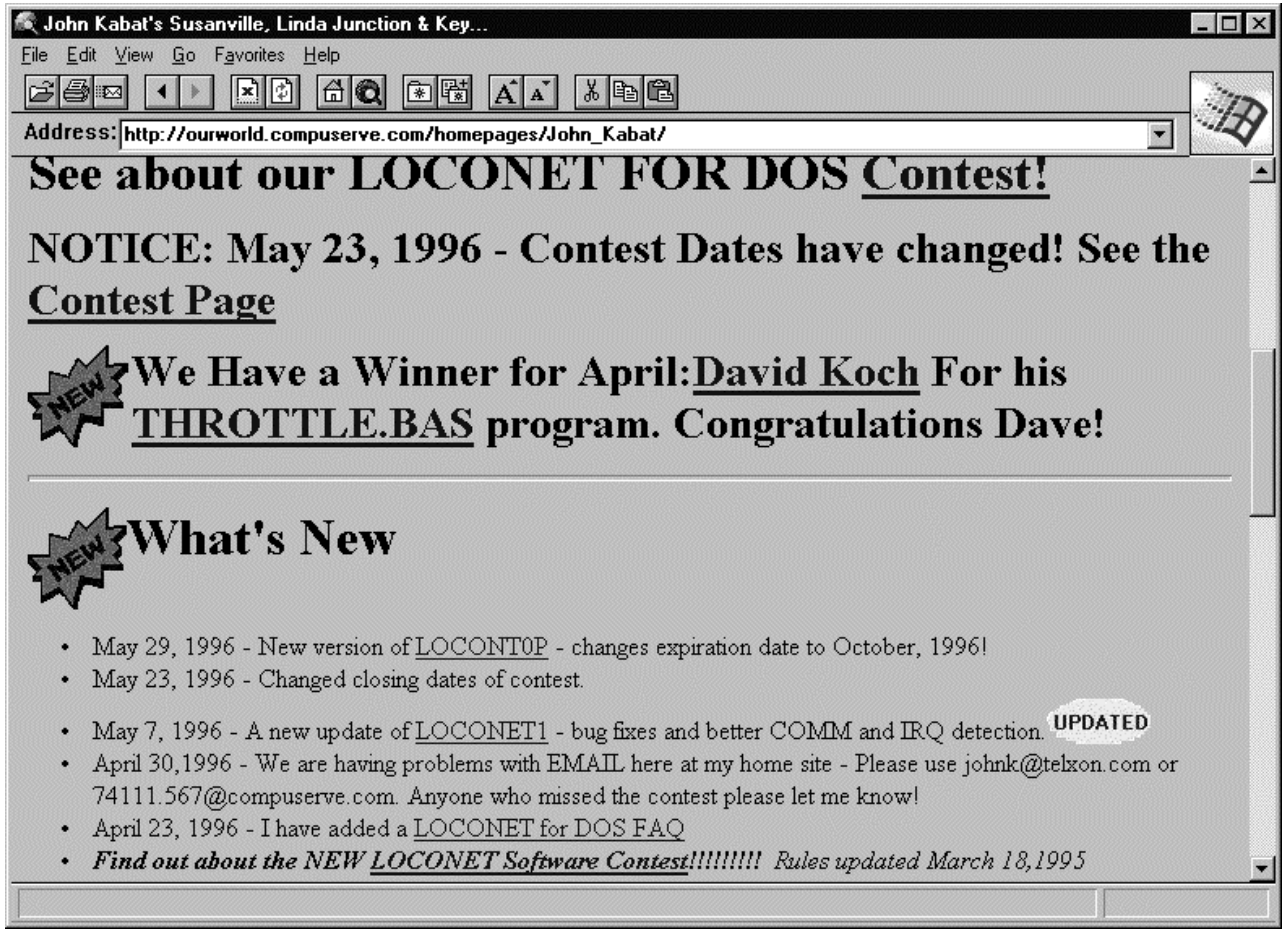
Questions ?

Matt Katzer
email: mkatzer@kam.rain.com
web: http://kam.rain.com
home: 503-291-1221

http://www.kamind.com



http://ourworld.compuserve.com/homepages/John_Kabat/



Other DCC web pages..

DCC Hardware

http://www.lenz.com

http://www.digitrax.com

http://www.wangrow.com

http://www.tttrains.com/tttrains/dccdiv.htm

DCC Software:

http://www.kamind.com

http:// ourworld.compuserve.com/homepages/John_Kabat/

http://www.modellbahn.com/www_links.html/

DCC information

http://www.tttrains.com/dcc/

http://www.mcs.net:80/~weyand/nmra/

http://www.mcs.net/~dsdawdy/NMRA/dcc.html

http://www.tttrains.com/tttrains/