11068 VSP

Virtual System Prototyping for 16-bit Devices using MPLAB® IDE and Proteus VSM

Class Objectives

When you finish this class you will:

- Understand the benefits and advantages of Virtual Simulation and Prototyping
- Be able to use Virtual Demo Boards within MPLAB® IDE
- Know how to rapidly test proof of concept ideas prior to full system development
- Use Virtual System Prototyping as part of the product development process
- Use standard Microchip application notes libraries and tools to aid rapid solution development

Agenda

- Setting the Scene
- Virtual System Prototyping
- Virtual Development Boards
- Lab1 – Explorer-16 VDB
- Lab2 – Adding Speech
- Lab3 – Adding Mass Storage
- Lab4 – Adding Ethernet

Setting the Scene

Setting the Scene
Market Information

- **Currently we manufacture a range of high quality monitoring equipment for**
  - Temperature
  - Pressure
  - Flow….etc.

- **Market information shows opportunities for monitoring equipment with**
  - Local audio feedback
    - **Audible spoken warnings and sensor readings**
  - Data recording features
    - **Usage and monitoring data**
  - Live monitoring and update capabilities
    - **Remote diagnostics, monitoring and updates**

Setting the Scene
The Scenario

You know the situation. You are working away quite happily when the boss calls you in. Hey Bob, we have a great new product idea, drop everything. We need to prove the concept asap.

Great!, you say, what's the deadline?

Oh, if you can get me the preliminary proof of concept for let's say next Thursday.

No problem, you say.

Now you have got 10 days to look at the concept, determine if it will work and create some proof of concept demonstrations.

Of course, you take it all in your stride. You have limited budget, little time to obtain development boards and no time to create any hardware but you do have a secret weapon to call on….. …..

Setting the Scene
What tools are available

- **Design will be based around the Microchip 16-bit Family of MCUs**
  - PIC24F chosen as starting point
    - **DV164005 – MPLAB® ICD2**
    - **DM240001 - Explorer-16 Demo Board**
    - **AC164125 - Speech Playback PICtail™ Plus**
    - **AC164122 - PICtail Plus for SD/MMC to SPI Interface**
    - **AC164123 - Ethernet PICtail Plus**

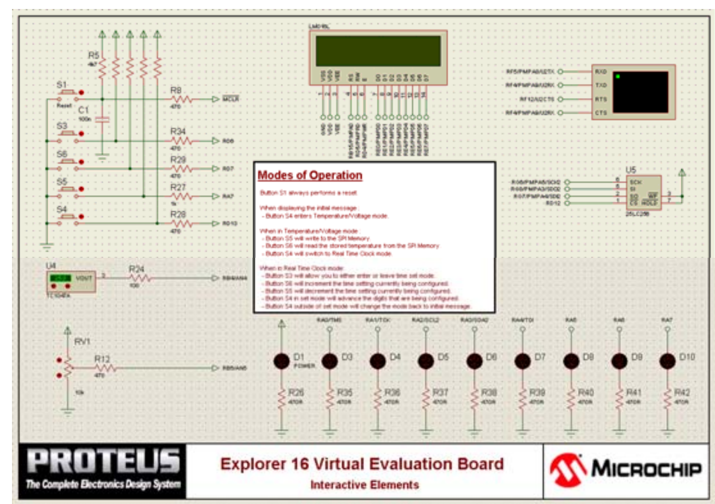
+

+

+

+

OR

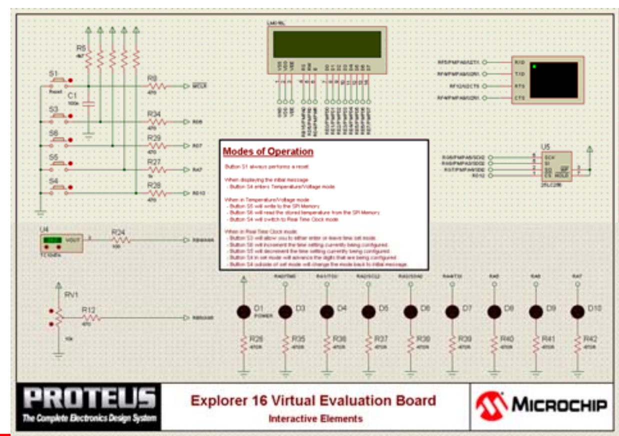


University of Microchip U of M MASTERs 2007

Setting the Scene
Our Tool Choice

- **Immediate Start Required so use easily available software tools**
  – Virtual Simulation and Prototyping with MPLAB® IDE and Proteus VSM
  – Code development using MPLAB C30 C Compiler
  – Application Notes and Libraries from Microchip

- **Proteus provides virtual equivalents of**
  – PIC24F devices in 64 to 100 pins
  – Development tool functions similar to MPLAB ICD2 / MPLAB SIM
  – DM240001 - Explorer-16 Demo Board
  – AC164125 - Speech Playback PICtail™ Plus
  – AC164122 - PICtail Plus for SD/MMC to SPI Interface
  – AC164123 - Ethernet PICtail Plus



Agenda

- Setting the Scene
- **Virtual System Prototyping**
- Virtual Development Boards
- Lab1 – Explorer-16 VDB
- Lab2 – Adding Speech
- Lab3 – Adding Mass Storage
- Lab4 – Adding Ethernet

Virtual System Prototyping - Overview

Agenda

- **What is Virtual System Prototyping?**
  - The Design Lifecycle
  - What is Proteus VSM?
  - Interactive Debug with MPLAB® IDE

UNIVERSITY OF MICROCHIP U OF M MASTERs 2007

Virtual System Prototyping

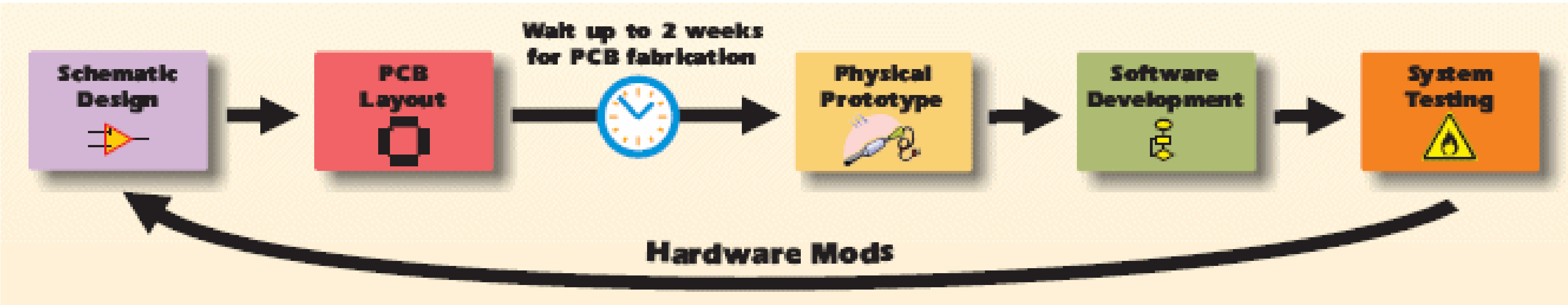
What is Virtual System Prototyping?

- **Development of a system using**
  - Software biased approach
  - Simulation and test of ideas and principles.
  -
- **Allows rapid development and modification**
  - Prior to a commitment to hardware
- **Simultaneous development of hardware and software**
  - On the actual design
  - Prior to availability of physical hardware
- **Provides early indication of**
  - Design flaws
  - Software/Hardware conflicts

Virtual System Prototyping

Classic Design Lifecycle

The critical design path in a typical project.

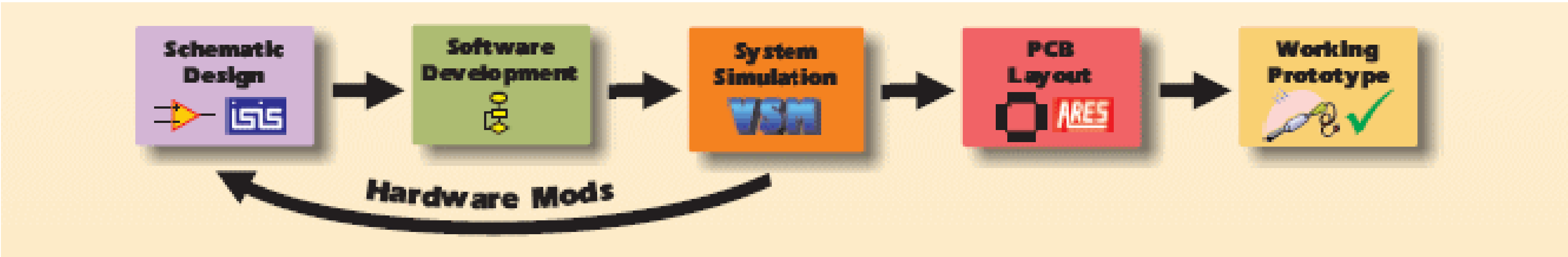


- Testing of the system cannot begin until a physical prototype is available
- Experimentation with code design is difficult without the system hardware
- Software/Hardware conflicts may lie undiscovered until late in the design cycle
- Changes to system hardware are time consuming, particularly if a new prototype is required

Virtual System Prototype

Design Lifecycle with Virtual Simulation

The critical design path with virtual simulation.

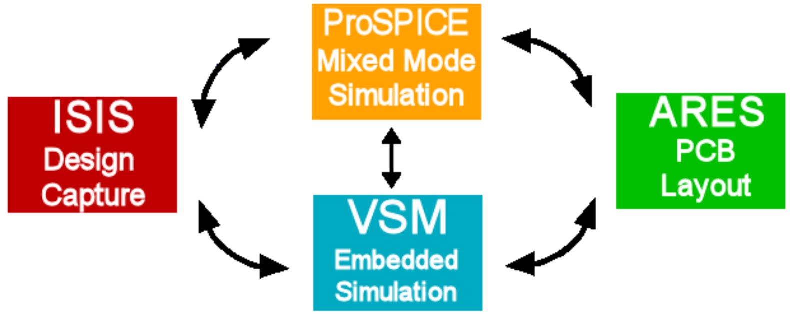


- The system is available for testing as soon as the schematic has been drawn
- Early simulation of software and its interaction with the entire system available prior to prototyping
- Changes to hardware design can be made as easily as changes to software design
- Potential issues can be evaluated and handled earlier in the design cycle
- Software development can continue in parallel on an actual target design

Virtual System Prototyping

What tools do we need?

- **Proteus VSM**
  - Schematic Capture
  - Mixed Mode Simulation
    - **SPICE Simulation**
      - Digital and Analog
    - **Virtual Simulation for Microcontrollers**
      - Cycle accurate models of PIC® microcontrollers and peripherals

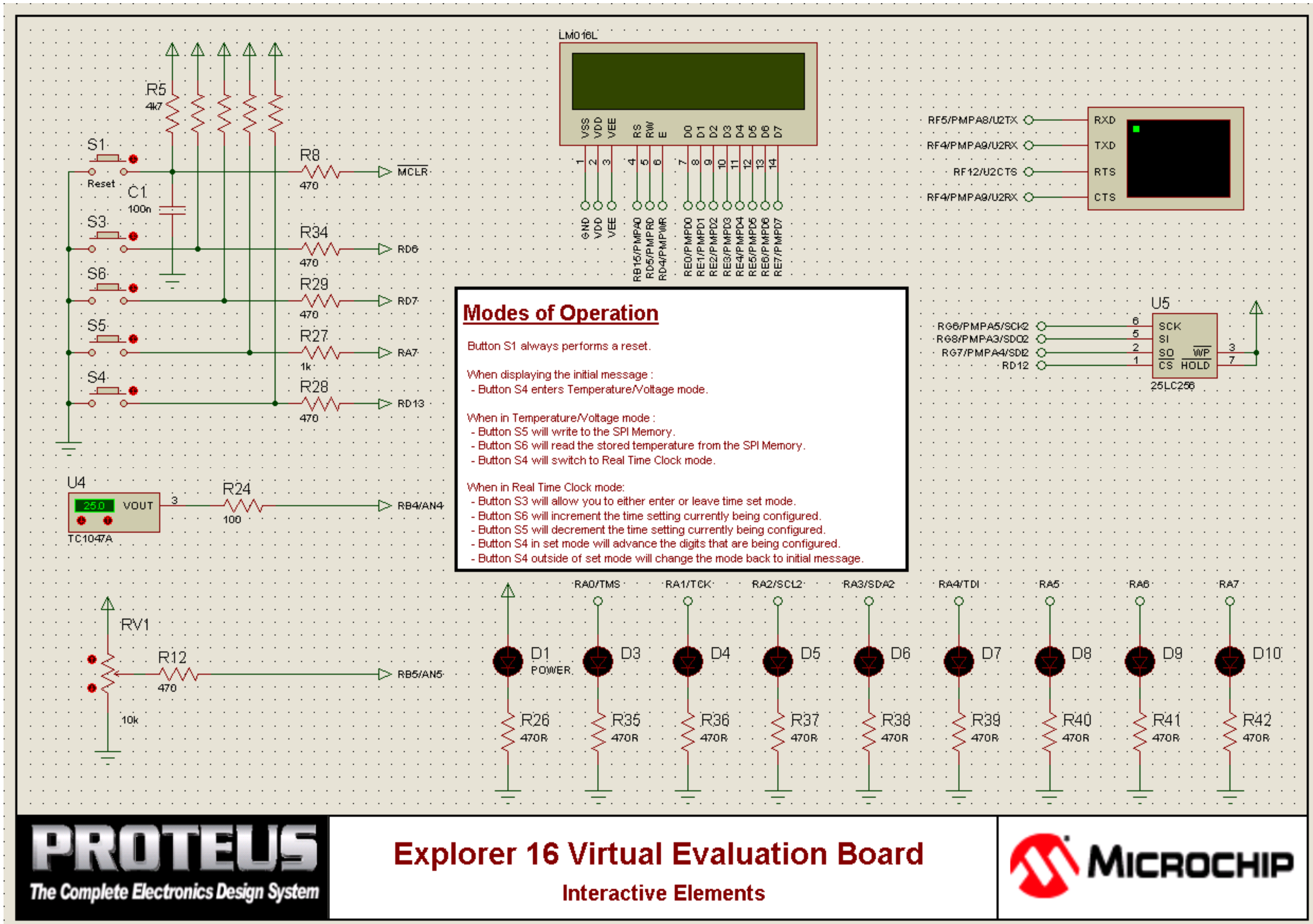


- **MPLAB® IDE**
  - Code Development
  - Debugging and Simulation
    - **Proteus VSM Plug-In**

Virtual System Prototyping

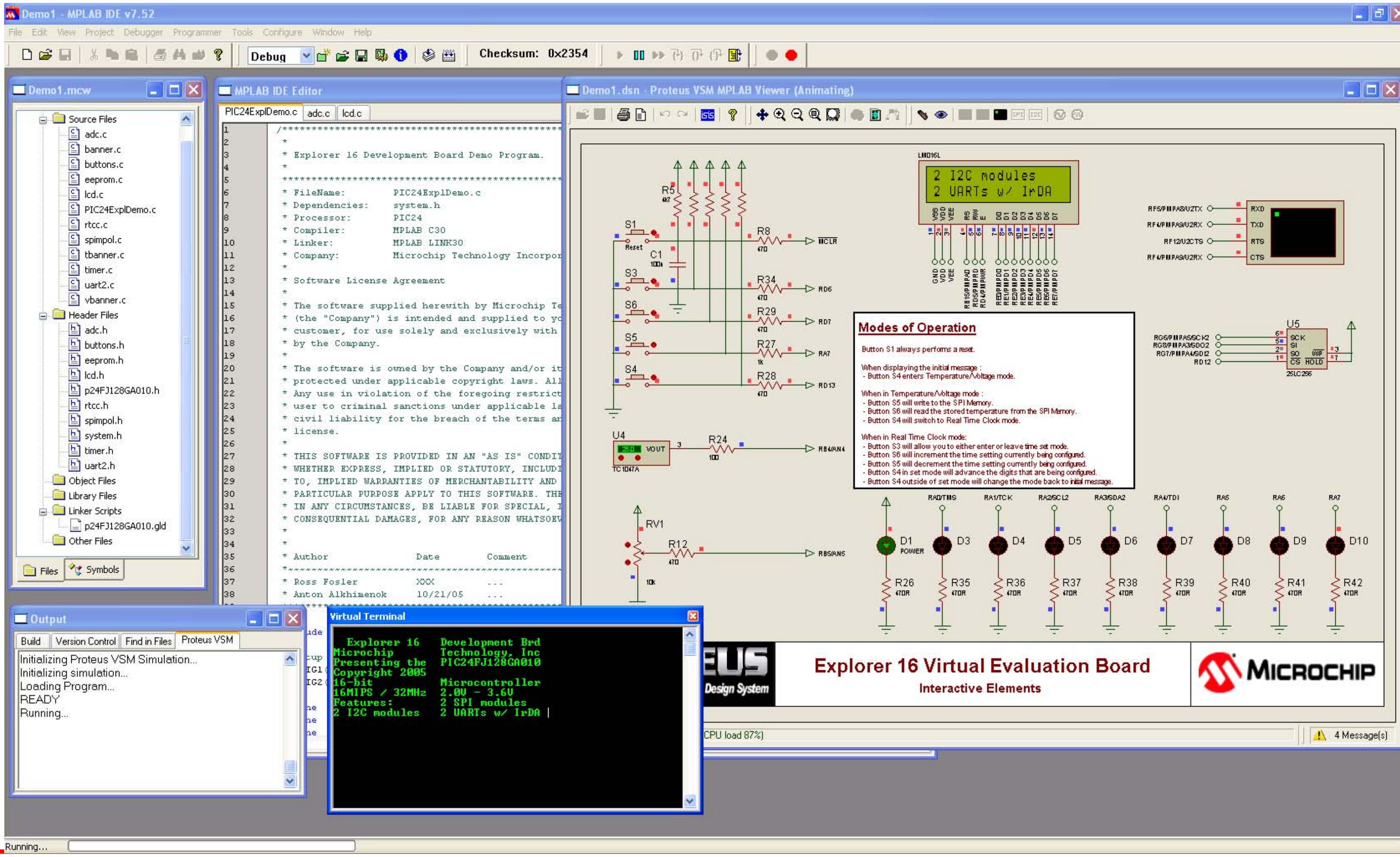
Proteus VSM – System Creation

- **System is generated as standard schematic**



Virtual System Prototyping
MPLAB® IDE Integration – System Simulation

- **System is integrated with MPLAB IDE via Proteus VSM Plug In**



Virtual System Prototyping

Virtual System Prototyping - Benefits

- **All of the factors discussed should result in**
  - Reduced design cycle time
  - Reduced effort
  - Reduced cost
  - Reduced time to market

Agenda

- Setting the Scene
- Virtual System Prototyping
- **Virtual Development Boards**
- Lab1 – Explorer-16 VDB
- Lab2 – Adding Speech
- Lab3 – Adding Mass Storage
- Lab4 – Adding Ethernet

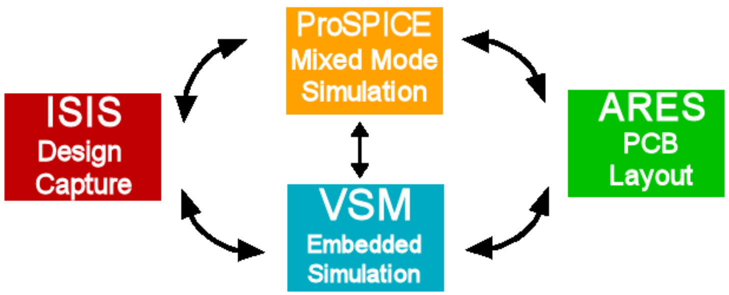
Virtual Development Boards

Agenda

- **Virtual Development Boards**
  - What is a Virtual Development Board (VDB)?
  - What tools are available to aid design and simulation?
  - Design/Simulation Trade-Offs

Virtual Development Boards
What is a Virtual Development Board?

- **A schematic representation of a system design**
  - Allows both
    - **Standard Design and Forward Annotation for PCB manufacture**
    - **Interactive Simulation**



  - How
    - **Components are standard schematic symbols with simulation model properties**
    - **Components can be used for simulation only**
  - Model Properties
    - **Models are faithful representations of real components**

Virtual Development Boards

What tools are available to aid design and simulation?

- **MCU**
  - Allows full simulation of
    - **Device Management Functions**
      - WDT
      - Timers
      - Sleep
      - Etc.
    - **ADC**
    - **Comms Peripherals**
      - UART/USART (incl EUSART)
      - SPI and I^2C™
    - **Plus many other features**

UNIVERSITY OF MICROCHIP UOFM MASTERS 2007

Virtual Development Boards

What tools are available to aid design and simulation?

- **SPICE**
  - Classic SPICE Analog simulation
    - **Based on Berkeley SPICE v3f5**
    - **Fully integrated with VSM microcontroller simulation engine**
- **Interactive Components**
  - Buttons, LED's, Pot's and active components
  - LCD's
    - **Segment, Matrix and Graphical**

Virtual Development Boards

What tools are available to aid design and simulation?

- **Suite of Virtual Instruments**
  - Oscilloscope
  - Logic Analyzer
  - Serial Terminal
  - Protocol Analyzers
  - DVM
  - Function Generators
  - Graphs
  - ….etc

- **Component and System Level Debugging**
  - Breakpoints, Watch Windows etc
  - Visual Logic State Indicators
  - Real Time Current and Voltage Probes
  - Component Level Diagnostics
  - Trace and Log functions for MCU and Peripherals

Virtual Development Boards

What tools are available to aid design and simulation?

- **Benefits**
  - Full System Simulation
    - **Peripherals function as would a real device**
  - ADC is fully modelled incl. Vrefs
    - **Can be simulated by external components**
  - Comms modules are fully simulated
    - **Terminal and Protocol Analyzers available**
  - External amplifiers and filters simulated in step with code
  - Debugging and interaction is similar to working with hardware

UNIVERSITY OF MICROCHIP U OF M MASTERs 2007

Virtual Development Boards

Design and Simulation Trade-Offs

- **Simulation Only**
  - No forward annotation to PCB expected
  - Components can be excluded from design
  - Only key components used
  - Simulation time reduced
  - Simple, small design
  - Easier to place virtual instruments etc

- **Fully Annotated Designs**
  - All Components required
    - **Some can be excluded from simulation and annotation**
  - Group interactive components on single sheet
    - **Allow space for instruments**

Agenda

- Setting the Scene
- Virtual System Prototyping
- Virtual Development Boards
- **Lab1 – Explorer-16 VDB**
- Lab2 – Adding Speech
- Lab3 – Adding Mass Storage
- Lab4 – Adding Ethernet

Hands-On Labs
Lab Structure

Virtual System Prototyping
Hands-On Labs: Structure

- **Intro and Explanation of**
  - Key Concepts
  - Design Considerations
    - **What decisions were made**
    - **Where to find more information**
  - Walk through Demo
    - **No need to follow along**
    - **Show and tell of key operations**
  - Lab using Lab Notes
    - **These will provide sufficient detail to follow along**
  - Additional Exercises
    - **These are OPTIONAL**
    - **More exercises are provided than there will be time to complete**
    - **Plenty to keep the high flyer's busy**

Virtual System Prototyping
Hands-On Labs: Tools

- **MPLAB® IDE**
  - Code Development
  - Debugging and Simulation
- **Proteus VSM**
  - Mixed Mode Simulation Tool
- **MPLAB C30 C Compiler**
- **Additonal tools**
  - Where needed to aid the system design additional tools will be introduced

Hands-On
Lab1 – Explorer-16 VDB

Agenda

- **Lab1 – Explorer-16 VDB**
  - What is implemented?
  - Types of simulation
    - **Interactive / Real Time**
    - **Batch Mode**
  - Virtual Instruments to aid debug
  - Using MPLAB® IDE and Proteus VSM
  - Lab1 Demo
  - Lab1 Practical

UNIVERSITY OF MICROCHIP U OF M MASTERs 2007

Lab1 – Explorer-16 VDB

What are we going to do?

- **Familiarise ourselves with the baseline toolset**
  - Use software tools
    - **MPLAB® IDE**
    - **MPLAB C30 C Compiler**
    - **Proteus VSM Simulator**
  - Open the Virtual Explorer-16 Demo Board
    - **Run the standard 'out of the box' hardware demo**
    - **Interact with the demo**
    - **Add a virtual instrument**
    - **Add a breakpoint to the source code**
- **How does this fit into our scenario**
  - Prior to adding new features we need a baseline system

Lab1 – Explorer-16 VDB
What is implemented?

- **Simulation Only Design**
  - Only functional or interactive components implemented
    - **No Power**
    - **No Connectors**
    - **No PIC18F4550 USB Device**
    - **No RS232 Device**
      - Replaced with Virtual Terminal

Lab1 – Explorer-16 VDB
What do we gain?

- **Simple, clean VDB**
  - Only essential components added
    - **Only components needed for simulation implemented**
    - **Quick and easy to implement proof of concept designs**
    - **Additional components can be added as needed**
  - Quick and easy to implement changes and modifications
    - **No Soldering**
    - **No Lifted Pads**
    - **No Wire Mods**
    - **No Component Changes**
    - **No Searching for Components**
    - **No Waiting for Orders to Arrive**
    - **No Waiting for PCB's**

Lab1 – Explorer-16 VDB
What are the limitations?

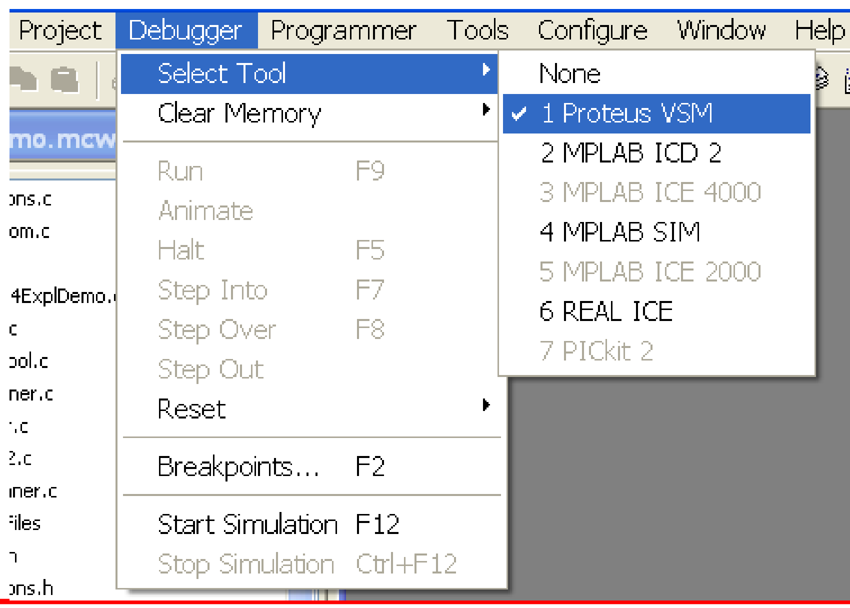
- **Simulation limited by**
  - Available models
    - **Often a nearest best fit can be used**
    - **Models can be generated / written**
      - SDK available (under NDA)
  - Real world issues
    - **Simulation of ESD/EMC and other effects are unimplemented**
    - **Models are pure functional equivalents – No Errata are modelled**

Lab1 – Explorer-16 VDB
Let's take a tour

- **Open Lab1**
  - Open Project in MPLAB® IDE
    - **Navigate to C:\Masters\11068\Lab1\Lab1a**
    - **Select Lab1a.mcp**
  - Follow along for this section
  - Will advise when you go solo

Lab1 – Explorer-16 VDB
Simulation Modes – Interactive Mode

- **How Do I Use Interactive Mode?**
  - Select Proteus VSM as Debugger
    - **Click 'Debugger' Menu**
      - Select Proteus VSM as Debugger tool
    - **Viewer Window Opens**
    - **If needed open required schematic design inside viewer**
      - Use the Open Design icon inside viewer window
      - Browse to and open relevant schematic file



Lab1 – Explorer-16 VDB

Let's take a tour

Character based LCD

Serial Terminal

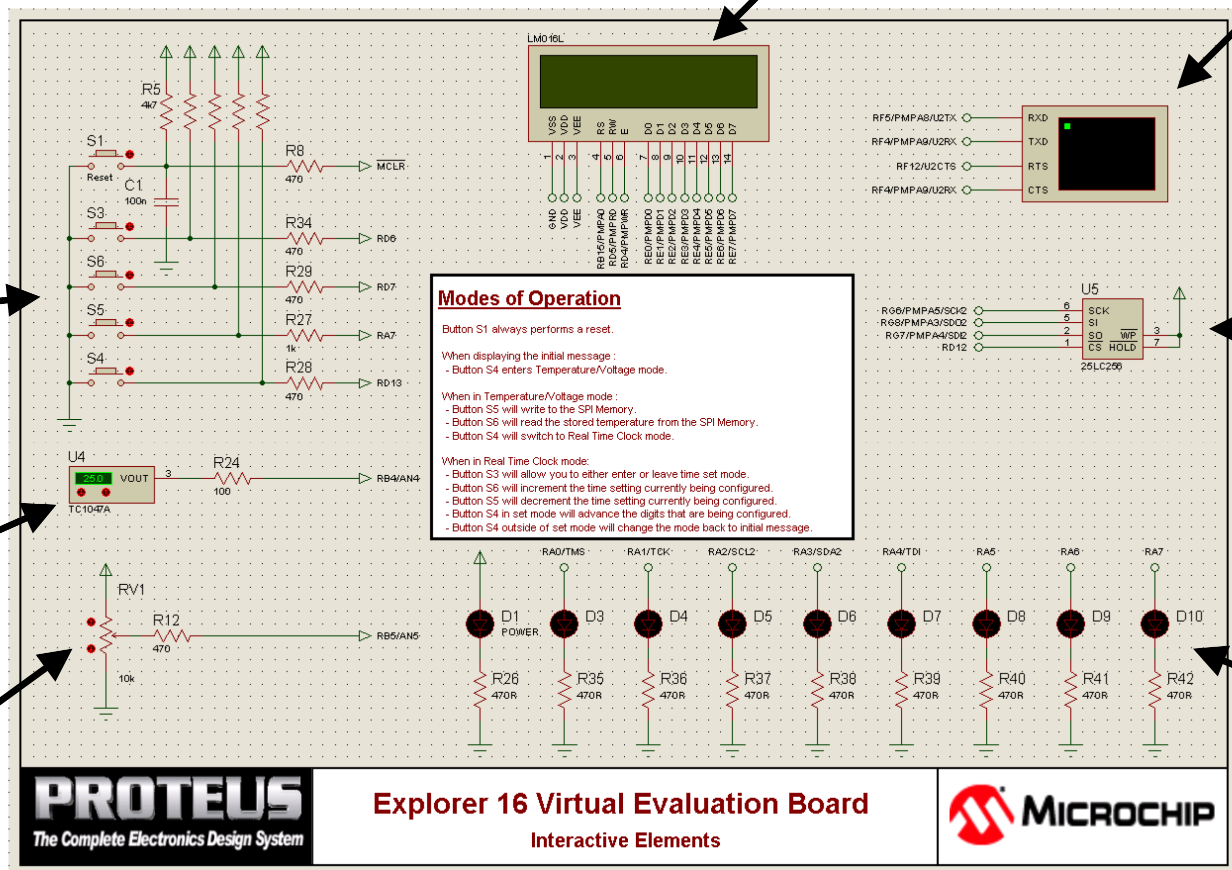
Interactive Buttons

Data EEprom

Interactive Temperature Sensor

Interactive Potentiometer

LED's



Lab1 – Explorer-16 VDB
Using Proteus VSM and MPLAB® IDE

- **Overview of Package Operation**
  - Simulation modes
    - **Interactive Mode**
      - Code level debug
      - Virtual Instruments active
      - Interactive Components operational
      - Similar to hardware
        - Uses same MPLAB IDE debugging functions
        - Allows interaction with design via Proteus VSM Viewer
    - **Batch Mode**
      - Mixed mode simulation
        - Allows SPICE and source code to be synchronised
        - Use with Analysis Graphs
        - Used in Lab2

Lab1 – Explorer-16 VDB
Interactive Components

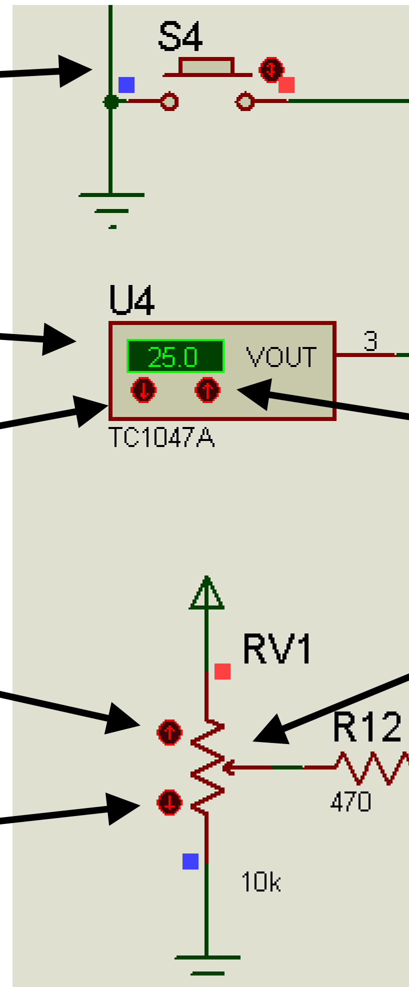
Interactive Buttons

Place Cursor over Button and Left Click

Interactive Temperature Sensor

Decrease Temperature

Increase Temperature



Potentiometer

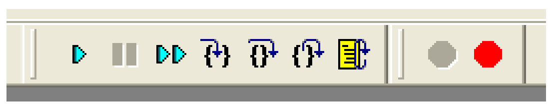
Increase

Decrease

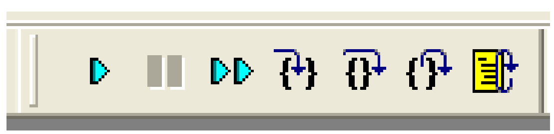
Lab1 – Explorer-16 VDB
Simulation Modes – Interactive Mode

- **Using Interactive Mode**
  - Click the Green Button
    - **Connects MPLAB® IDE to Proteus VSM Simulator**

    - **Turns to RED when MPLAB IDE is connected to Proteus VSM**
      - Enables interactive buttons
      - Opens Virtual Instrument windows



  - Use Normal MPLAB IDE functions to debug
    - **MPLAB IDE operates as per MPLAB SIM or MPLAB ICD2/Real ICE**



Lab1 – Explorer-16 VDB
Lab1 - Demonstration

Follow the Lab Notes

- **Lab1a**
  - Connect MPLAB® IDE to Proteus VSM
    - **Run Simulation**
      - *may request a code out of date re-build
    - **Use interactive buttons to scroll through menus on LCD**
    - **Modify Potentiometer position and observe changes to voltage on LCD**
    - **Modify Temperature using buttons on interactive temperature sensor**

Lab1 – Explorer-16 VDB
Lab1 – Additional Excercise

Follow the Lab Notes

- **Lab1b (optional)**
  - Stop and disconnect Proteus VSM Simulator
    - **Hint : Press Red Button**
  - Add a DVM to each of
    - **Net R12-2/RB5**
    - **Net R24-2/RB4**
  - Add Breakpoints in ADC.c to capture ADC conversions
    - **Compare DVM readings to measured and calculated values on LCD**
      - Allows check of Vout range of temperature sensor to reported temperature
        - Confirms calculations and sensor operations
        - Allows range checking in code etc

Virtual System Prototyping
Hands-On Labs : Timing

BREAK

Back in 10mins for
Lab2 – Adding Speech to your design

UNIVERSITY OF MICROCHIP UOFM MASTERS 2007

Agenda

- Setting the Scene
- Virtual System Prototyping
- Virtual Development Boards
- Lab1 – Explorer-16 VDB
- **Lab2 – Adding Speech**
- Lab3 – Adding Mass Storage
- Lab4 – Adding Ethernet

Hands-On
Lab2 – Adding Speech to Your Design

Agenda

- **Lab2 – Adding Speech/Audio to Your System**
  - System choices for Speech Playback
    - **Quality**
    - **Memory**
    - **Algorithms**
  - Playback hardware
  - Message creation

Lab2 – Adding Speech
Market Requirements

- **Market requirement #1**
  - Provide Speech/Audio playback capability
    - **Audible Warnings or Readouts**
    - **Set Point or Alarm messages**
    - **May require**
      - Multiple languages to suit market needs

UNIVERSITY OF MICROCHIP U OF M MASTERs 2007

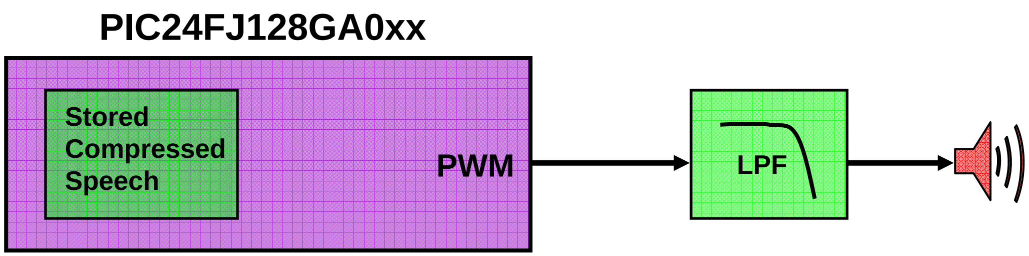
Lab2 – Adding Speech
What are we going to do?

- **Add speech playback to our system**
  - Use a readily available demo to create a speaking thermometer
- **Provides a platform to evaluate**
  - Low cost PWM DAC generated speech output
- **Introduces method to**
  - Create and manipulate samples
  - Store in memory
  - Recall from memory relative to real events
- **Introduces additional software tools**
  - Goldwave : wave editor
  - Winspeech : compression tool
  - MPFS : filing system tool

Lab2 – Adding Speech

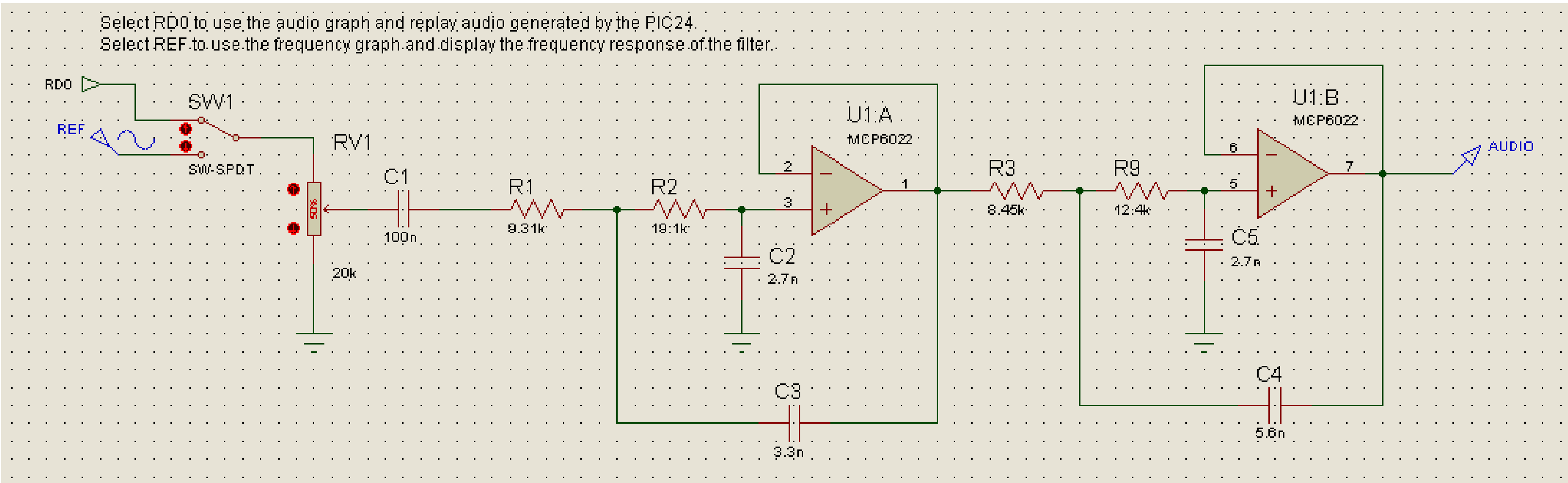
System Choices for our Playback Only System

- **MCU**
  - PIC24FJ128GA010
    - **Ample Internal Memory for Application and Data**
  - PWM / Output Compare Module
    - **Allows 16kHz at 16bits to be implemented easily**
- **Output Filter**
  - At least 4th order Low Pass
    - **Cut-off Freq c. 4kHz**
    - **Low Cost components**
- **Memory**
  - Internal Program Memory
  - MPFS Filing System implemented
- **Algorithm**
  - IMA ADPCM Chosen
    - **Easy to implement**
    - **Low MIPS requirement, No DSP so can be implemented in an MCU**
    - **Adequate Compression ratio at 4:1**



Lab2 – Adding Speech
Message Output

- **A Digital to Analog Converter (DAC) is created using**
  - PWM Output from Output Compare Module
    - **Frequency and Resolution are limited**
  - Analog Filter
    - **At least 4th Order Filter recommended**



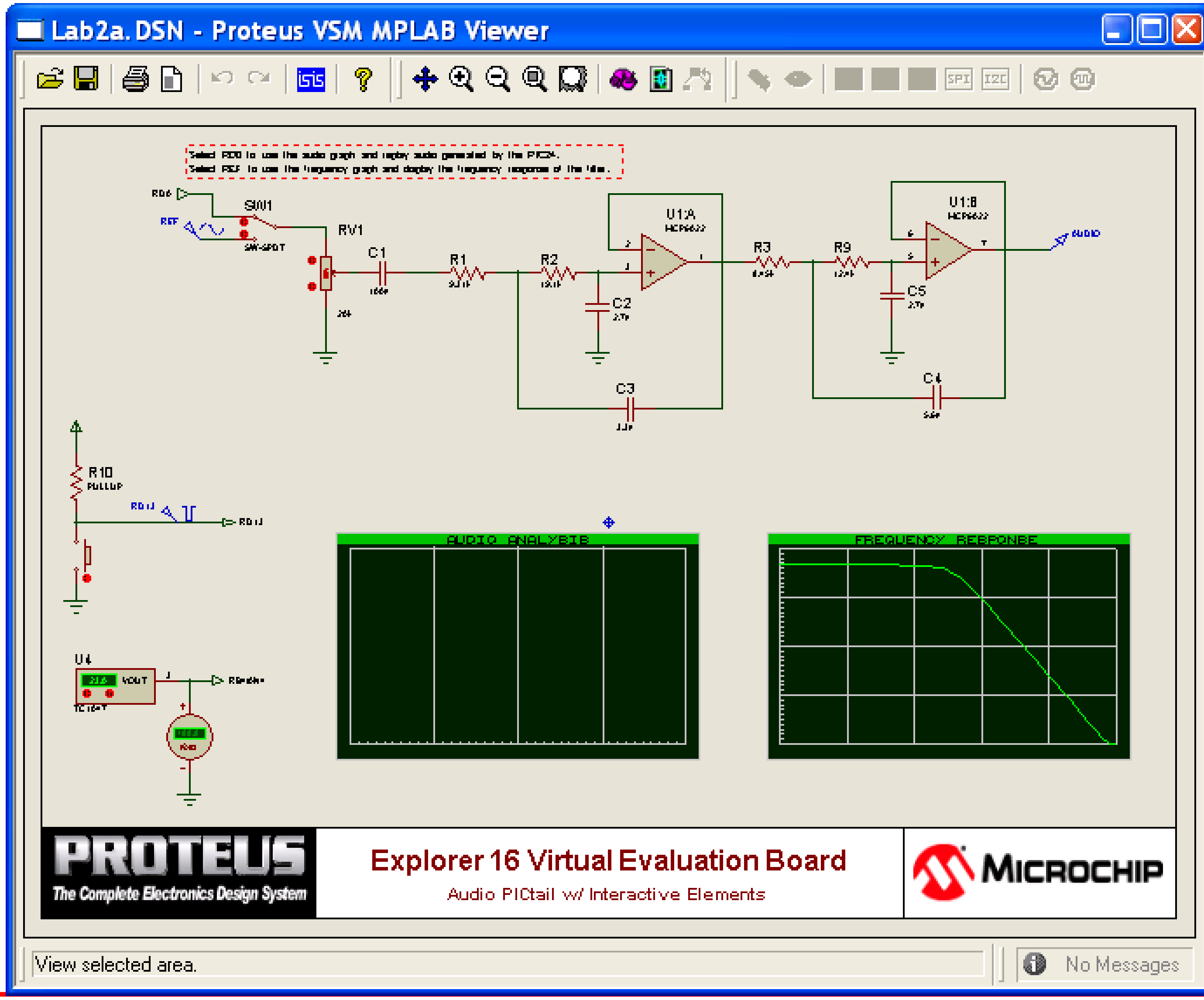
Lab2 – Adding Speech

Lab2 – Open Lab

- **Open Lab2.mcp**
  - Open Project in MPLAB® IDE
    - **Navigate to C:\Masters\11068\Lab2\Lab2a**
    - **Select Lab2a.mcp**
  - Perform batch mode operation to generate speech output

Lab2 – Adding Speech

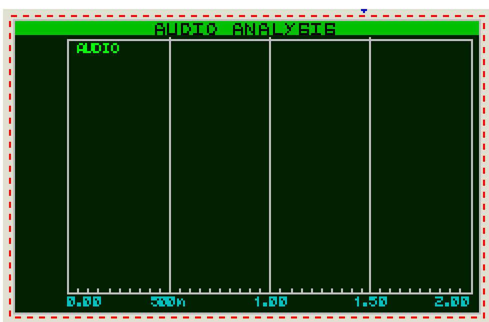
Lab2 – What we should see is…



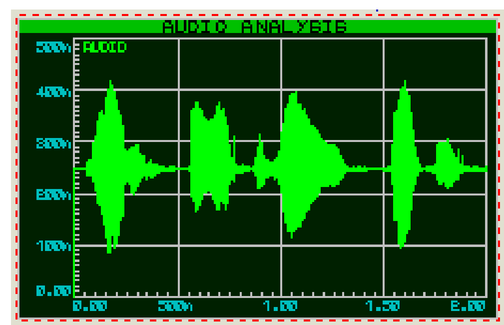
Lab2 – Adding Speech
Simulation Modes – Batch Mode

- **Using Batch Mode**
  - No Connection to Proteus VSM Simulator is made
    - **Button should stay GREEN**

- **Place mouse pointer over Audio Analysis Graph**
  - A Red Dashed Line Box will appear around graph
  - Press Space Bar OR Right-Click and Select Simulate Graph



Pre-Simulation



Post Simulation

UNIVERSITY OF MICROCHIP U OF M MASTERS 2007

Lab2 – Adding Speech
How Speech Output is Constructed

- **Phrase constructed from a series of individual words**
  - In this case for 23°C, we use
    - **Twenty**
    - **Three**
    - **Degrees**
    - **Celsius**
  - Complete vocabulary stored in minimal memory
- **Vocabulary consists of individual words**
  - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,30,40,50,60,70,80,90,100,Degrees,Celcius, Fahrenheit

Lab2 – Adding Speech

Lab2 - Demonstration

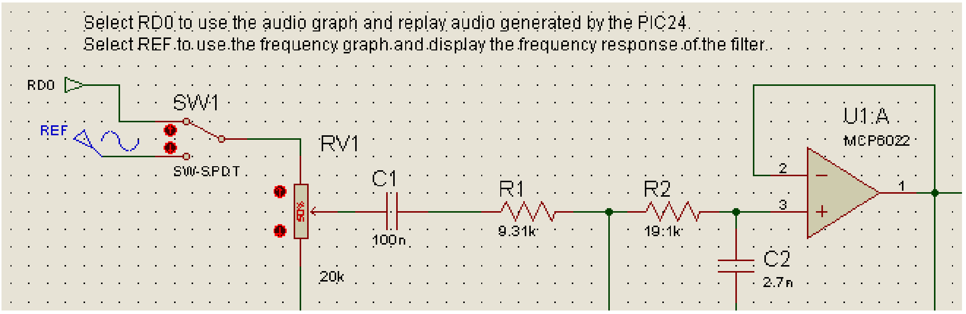
Follow the Lab Notes

- **Lab2a**
  - Run simulation
    - ***may request a code out of date re-build**
  - DO NOT CONNECT to VSM
    - **Batch mode simulation required**
  - Move Cursor over Analysis Graph
    - **Press Space Bar**
  - Allow simulation to complete
    - **Message should play**
    - **Use Ctrl-Space to re-play message**
  - Modify temperature and re-run

Lab2 – Adding Speech
Lab2 - Additional Excercise

Follow the Lab Notes

- **Lab2b (optional)**
  - Observe the differences between filter types
    - **Run the Audio Analysis as for Lab2a**
  - Change switch position to determine filter response
    - **Switches a 1kHz Sine wave into filter**
  - Select the Frequency Response Graph
    - **Press <Space> to run simulation**
  - Select the Fourier Analysis Graph
    - **Press <Space> to run simulation**
  - Observe the effects and modify the filter characteristics
    - **Change values of passives to modify filter**



  - Select, in turn, Frequency Response and Fourier Analysis Graphs
    - **Press <Space> to run simulation and observe changes**
  - Change switch position to PWM DAC signal
    - **Simulate message as Lab2a**

Lab2 – Adding Speech

Lab2 - Additional Excercise

Follow the Lab Notes

- **Lab2c (optional)**
  - Open Lab2c
    - **Code setup for full temperature range playback**
  - Record 'Zero', 'Minus', 'Two Hundred', 'AND' Messages
    - **Use PC and Microphone/Speakers**
    - **Manipulate using Goldwave**
  - Compress using Winspeech
  - Add messages to image file using MPFS v2
  - Add revised MPFSImgASM.s to your project
    - **Build Project**
  - Set required temperature for message to play back
  - Run simulation
  - *comprehensive details in Lab notes and/or appendix

Virtual System Prototyping
Hands-On Labs : Timing

BREAK

Back in 10mins for
Lab3 – Adding Mass Storage to your design

University of Microchip UofM Masters 2007

Agenda

- Setting the Scene
- Virtual System Prototyping
- Virtual Development Boards
- Lab1 – Explorer-16 VDB
- Lab2 – Adding Speech
- **Lab3 – Adding Mass Storage**
- Lab4 – Adding Ethernet

Hands-On
Lab3 – Adding Mass Storage

Agenda

- **Lab3 – Adding Mass Storage**
  - What are the options?
  - Filing Systems
  - Using a Virtual Mass Storage Device

Lab3 – Adding Mass Storage
Market Requirements

- **Market requirement #2**
  - Provide PC compatible Mass Storage capability
    - **Storage of usage and monitoring data**
    - **External memory for speech playback**
      - Aid storage for multi-lingual systems

Lab3 – Adding Mass Storage

What are we going to do?

- **Discuss design factors relative to adding mass storage to a system**
- **Create a functional system utilising**
  - External flash memory mass storage card
  - Microchip FAT16 Library
- **Use virtual equivalents of**
  - DM240001 - Explorer-16 Demo Board
  - AC164122 - PICtail™ Plus for SD/MMC to SPI Interface
- **Understand how to use additional tools to**
  - Add files to a virtual mass storage device image file

Lab3 – Adding Mass Storage

What Filing System Options Are Available

	Direct Access	MPFS	FAT16	Proteus Model	Licence Required
Internal					
Program Flash	✓	✓		✓	
Serial					
Data EE	✓	✓		✓	
Serial Flash	✓	Future		✓	
SD/MMC	✓		✓	✓	✓
Parallel					
Parallel Flash	✓				
Compact Flash	✓		✓	✓	✓
ATA HDD	✓		✓	✓	

Lab3 – Adding Mass Storage
Filing Systems - FAT16

- **For our purposes a SD/MMC or Compact Flash card can be used**
  - Microchip FAT16 Library implemented
    - **Best option for technology chosen**
    - **Easy interaction with PC**
    - **Use of MPFS or raw data would require a more complex target interface**
    - **Overhead of FAT16 library is acceptable**
- **Creation of Multi Language Talking Thermometer**
  - Similar to Lab2a
  - Increased memory capacity allows for increased functionality
  - Uses FAT16 filing system
  - Concept lends itself as basis for many other designs

Lab3 – Adding Mass Storage

Open Lab3

- **Open Lab3.mcp**
  - Open Project in MPLAB® IDE
    - **Navigate to C:\Masters\11068\Lab3\Lab3a**
    - **Select Lab3a.mcp**

Lab3 – Adding Mass Storage

Lab3 - Demonstration

Follow the Lab Notes

- **Lab3a**
  - Run simulation
    - ***may request a code out of date re-build**
  - DO NOT CONNECT to VSM
    - **Batch mode simulation required**
  - Move cursor over Analysis Graph
    - **Press Space Bar**
  - Allow simulation to complete
    - **Message should play**
    - **Use Ctrl-Space to re-play message**
  - Modify Temperature or Language Setting and re-run
  - Message is now being read from external mass storage rather than internal program memory

Lab3 – Adding Mass Storage

Lab3 - Additional Excercise

Follow the Lab Notes

- **Lab3b (optional)**
  - Record a Message and compress
    - **As per Lab2c**
  - Add file to Virtual Mass Storage device
    - **Use software utility to add files to virtual drive image file**
  - Move Cursor over Analysis Graph
    - **Press Space Bar**
  - Allow simulation to complete
    - **Message should play**
    - **Use Ctrl-Space to re-play message**

Virtual System Prototyping
Hands-On Labs : Timing

BREAK

Back in 10mins for
Lab4 – Adding Ethernet to your design

Agenda

- Setting the Scene
- Virtual System Prototyping
- Virtual Development Boards
- Lab1 – Explorer-16 VDB
- Lab2 – Adding Speech
- Lab3 – Adding Mass Storage
- **Lab4 – Adding Ethernet**

Hands-On
Lab4 – Adding Ethernet

Agenda

- **Lab4 – Adding Ethernet**
  - Ethernet and the Virtual ENC28J60
  - How it connects to the real world
  - Simulation considerations for Real Time operation
  - Connecting to the Network
  - Observing 'your' network transactions

Lab4 – Adding Ethernet
Market Requirements

- **Market requirement #3**
  - Ability to monitor/control remotely
    - **Enables live feedback/monitoring of data**
    - **Enables remote updates and changes**
    - **Enables remote diagnostics capability**

Lab4 – Adding Ethernet

What are we going to do ?

- **We will**
  - Discuss design factors relative to adding Ethernet to a system
  - Create a functional system utilising
    - **ENC28J60 Ethernet controller**
    - **Microchip TCP/IP Stack**
  - Use virtual equivalents of
    - **DM240001 - Explorer-16 Demo Board**
    - **AC164123 - Ethernet PICtail™ Plus**
  - Understand how to use additional tools to
    - **Capture network traffic**
    - **Setup and view network devices**

Lab4 – Adding Ethernet
Simulation Considerations

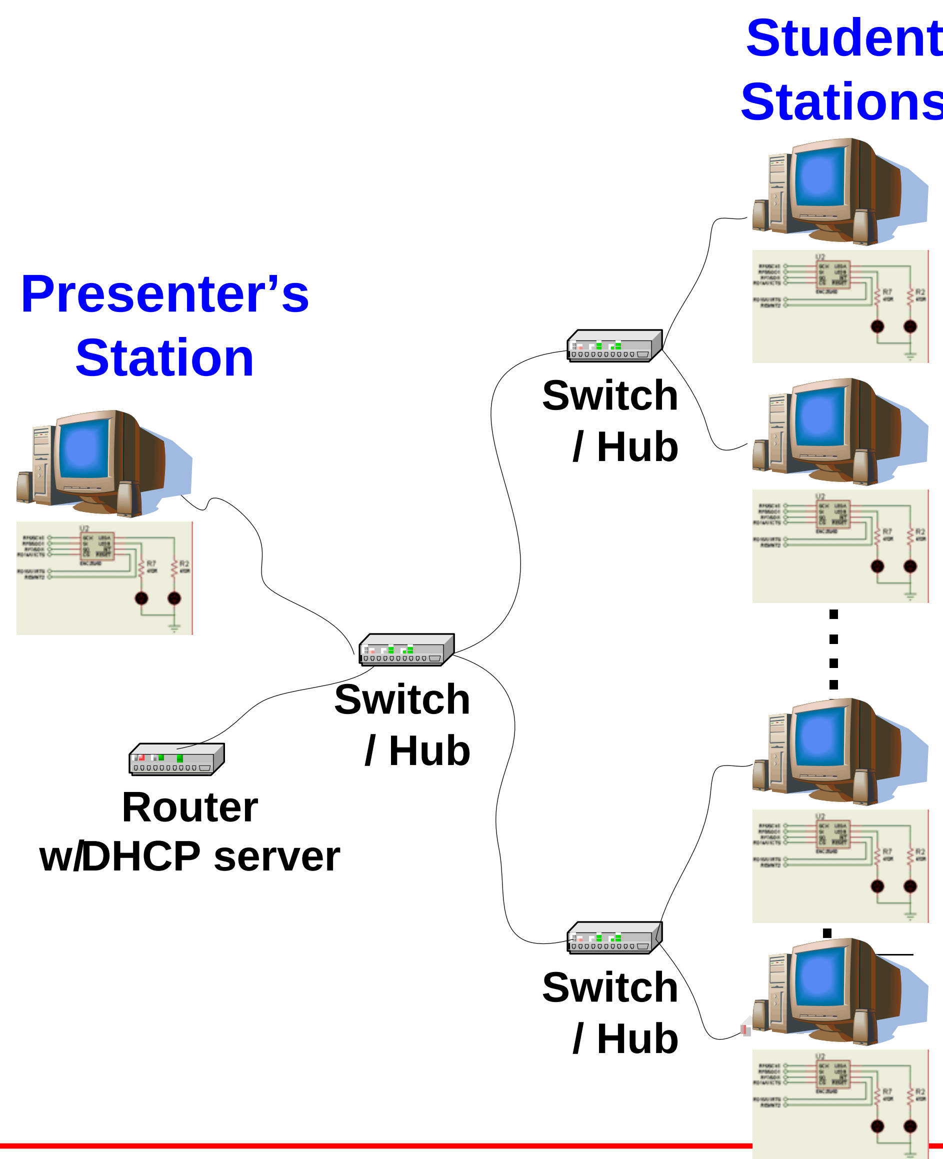
- **Simulation is CPU intensive**
  - Consider active devices
    - **PIC24FJ128GA010**
      - Running Application and TCP/IP Stack
    - **ENC28J60**
    - **LCD, Serial Terminal etc**
  - Load on CPU is heavy
    - **Where possible exclude unnecessary devices from simulation**
    - **A fast PC will help**
      - Minimise number of active applications and resource sinks

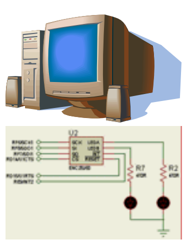
Lab4 – Adding Ethernet
Connecting to a Network

- **Simulation contained within PC**
  - Will need an active DHCP server
  - PC is supporting multiple IP addresses
    - **WinPcap hiding the virtual board from PC resource map**
    - **WinPcap enables PC to support traffic to browser, virtual demo board and packet sniffer**
  - Virtual ENC28J60 will request IP address via DHCP from server
    - **Once active other users can connect to your virtual board via network**
  - Ethernet Discoverer will list users as they connect

Lab4 – Adding Ethernet
Classroom Network

Use Dynamic IP Addresses for all virtual boards and computers (DHCP)



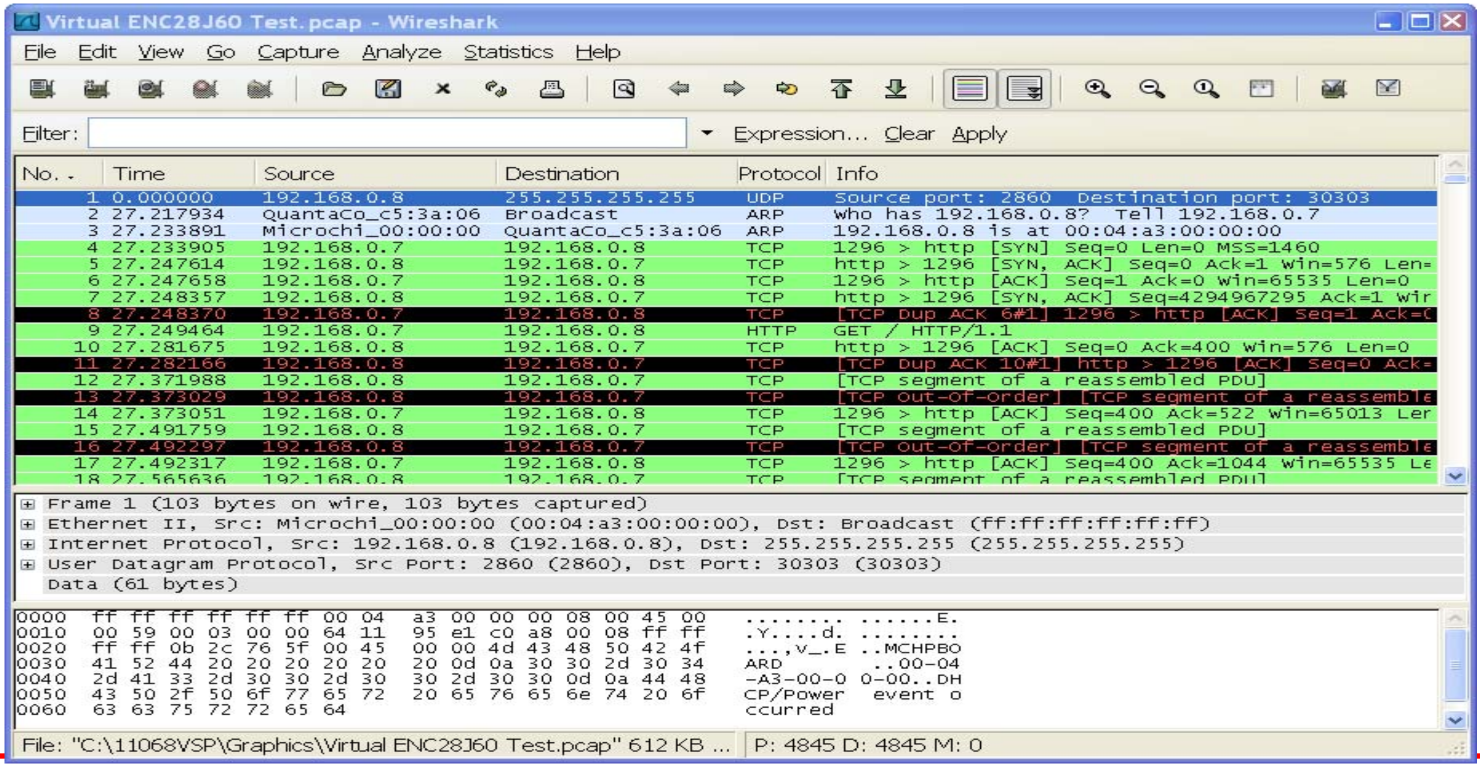


Computer

Virtual Explorer 16 w/Ethernet PICtail™ Plus Board

Lab4 – Adding Ethernet
Observing Network Transactions

- **Observe Network Transactions using Wireshark**
  - World's foremost network protocol analyzer
    - **Allows network traffic to be captured, filtered and analysed**
      - Utilises WinPcap



Lab4 – Adding Ethernet
Open Lab4

- **Open Lab4.mcp**
  - Open Project in MPLAB® IDE
    - **Navigate to C:\Masters\11068\Lab4\Lab4a**
    - **Select Lab4a.mcp**

Lab4 – Adding Ethernet
Lab4 - Demonstration

Follow the Lab Notes

- **Lab4a**
  - Open file TCPIPConfig.h
    - **Navigate to - #define MY_DEFAULT_HOST_NAME**
    - **Use the unique hostname provided**
  - Scroll down to - #define MY_DEFAULT_MAC_BYTE5
    - **Add the MAC Address Provided to**
      - MY_DEFAULT_MAC_BYTE5
      - MY_DEFAULT_MAC_BYTE6
  - Check the Default IP Addr - #define MY_DEFAULT_IP_ADDR_BYTE'n'
    - **Ensure Default IP Addr is 192.168.0.xxx**
    - **Where 'xxx' is the IP Address Provided**
    - **Required should DHCP fail to serve an address**
  - Build Source
  - Connect to Proteus VSM Simulator
    - **Press Green Button**
  - Run Simulation
    - **If running correctly an IP address should appear in the LCD and/or Virtual Terminal**

Lab4 – Adding Ethernet

Lab4 – Additional Excercise

Follow the Lab Notes

- **Lab4b (optional)**
  - Interact with virtual Ethernet
    - **Use served IP address to**
      - Connect to board via
        - ftp
        - http
      - Ping board
      - Use serial port and virtual terminal to connect to device and setup parameters

Delivering the Goods
Our Report

- **Our 10 days is over...the report and presentation is now required**
  - Fundamentally we can achieve all of the desired product additions using readily available software and hardware at a reasonable cost
    - **Microchip Demo's and App Notes provide good application framework**
    - **Virtual Prototyping can be used to**
      - Accelerate project
      - Increase productivity
      - Reduce re-work
      - Get to market faster

Project Approved.....

UNIVERSITY OF MICROCHIP UOFM MASTERs 2007

Summary

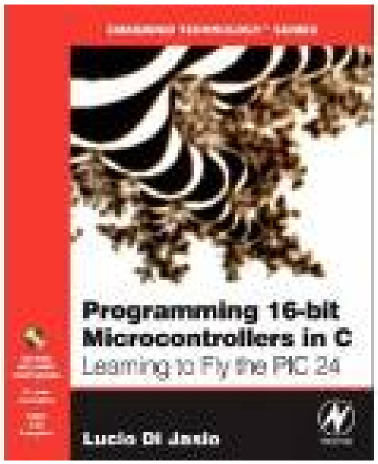
We should now

- Understand the benefits and advantages of Virtual Simulation and Prototyping ✓
- Be able to use Virtual Demo Boards within MPLAB® IDE ✓
- Know how to rapidly test proof of concept ideas prior to full system development ✓
- Use Virtual System Prototyping as part of the product development process ✓
- Use standard Microchip application notes libraries and tools to aid rapid solution development ✓

Want to know more

- **For more details of Proteus VSM**
  - visit www.labcenter.com
  - OR visit Labcenter at the Third Party Booths

16-bit Books



Lucio DiJasio

"Programming 16-bit Microcontrollers in C – Learning to Fly the PIC24"

- **Covers a number of topics similar to this class**
  - Speech/Audio Playback
  - Mass Storage
    - **FAT Filing System**

Feedback

- **Please complete the class feedback questionnaire**
  - If you liked it let us know
  - If you didn't we need to know also

- **Is there anything you would like to see in the class?**

The End

Thank you for attending.

We hope you find it useful.

References
General

- **Application Notes**
  - AN643 – Adaptive Differential Pulse Coded Modulation using PIC® microcontrollers
  - AN538 – Using PWM to Generate Analog Output
  - AN833 – Microchip TCP/IP Stack
- **Masters Classes**
  - 1096VSM – Virtual Simulation for Microcontrollers using MPLAB® IDE and Proteus VSM
  - 977SPE – Adding Speech to Low Cost Microcontrollers
  - 1042CAL – Using Communication Application Libraries with Microchip's 16-bit Microcontrollers
  - 1040HOE – Hands on Ethernet

References
Libraries and Data

- **Microchip Libraries**
  - FAT16
  - TCP/IP
- **Data Sheets**
  - DS39747C PIC24FJ128GA010
  - DS21498C TC1047A
  - DS39662B ENC28J60

References
Development Tools

- **Microchip**
  - DV164005 – MPLAB® ICD2
  - DM240001 – Explorer-16 Demo Board
  - AC164125 – Speech Playback PICtail™ Plus
  - AC164122 – PICtail Plus for SD/MMC to SPI Interface
  - AC164123 – Ethernet PICtail Plus
  - SW006012 – MPLAB C30 C compiler

UNIVERSITY OF MICROCHIP UofM MASTERs 2007

References
Other Software and Utilities

- **Microchip**
  - Included within Application Note source code and demos
    - **Winspeech**
    - **MPFS**
    - **Microchip Discoverer**

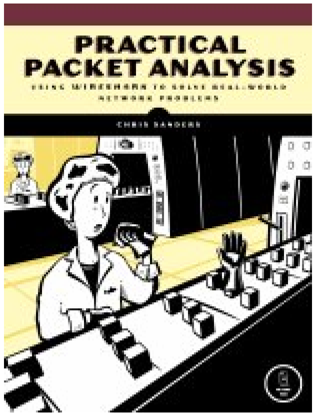
- **Labcenter**
  - Included within Proteus VSM install and Demos
    - **FATutil**
    - **IFLIST**

- **Third Party**
  - Cygus HEX Editor (Free Edition), http://www.softcircuits.com/cygnus/fe/
  - WinPcap, http://www.winpcap.org/
  - Wireshark, http://www.wireshark.org/
  - Goldwave, http://www.goldwave.com/

References
Books

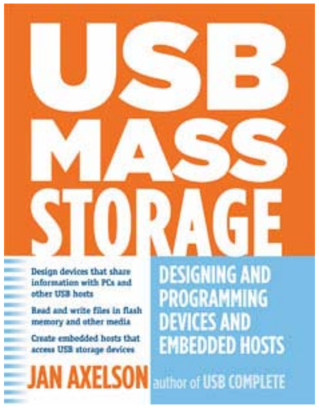
- **Useful Books**
  - **Practical Packet Analysis**
    Using Wireshark to Solve Real-World Network Problems
    by Chris Sanders

    May 2007, 216 pp.
    ISBN-10 1-59327-149-2
    ISBN-13 978-1-59327-149-7



  - **USB Mass Storage**
    Designing and Programming Devices and Embedded Hosts
    by Jan Axelson

    August 2006, 287 pp
    Publication date:
    ISBN# 1-931448-04-3



Appendix

Appendix
Setting the Scene

Setting the Scene
Our Task

- **Investigate available technologies for**
  - Audio Playback
    - **Record may also be required in the future**
  - Storage of Data on PC compatible storage cards
  - Internet connectivity
- **Will allow us to evaluate**
  - Effort required
  - Costs involved
  - Benefits gained
- **This will aid decision making process**
  - Can we enter market cost effectively
  - What resources will be required
    - **Engineering / Product development time needed**
    - **Manufacturing changes**
    - **Training for Installation Personnel**
  - Likely timescales to develop full products

Setting the Scene
What do we have to help us?

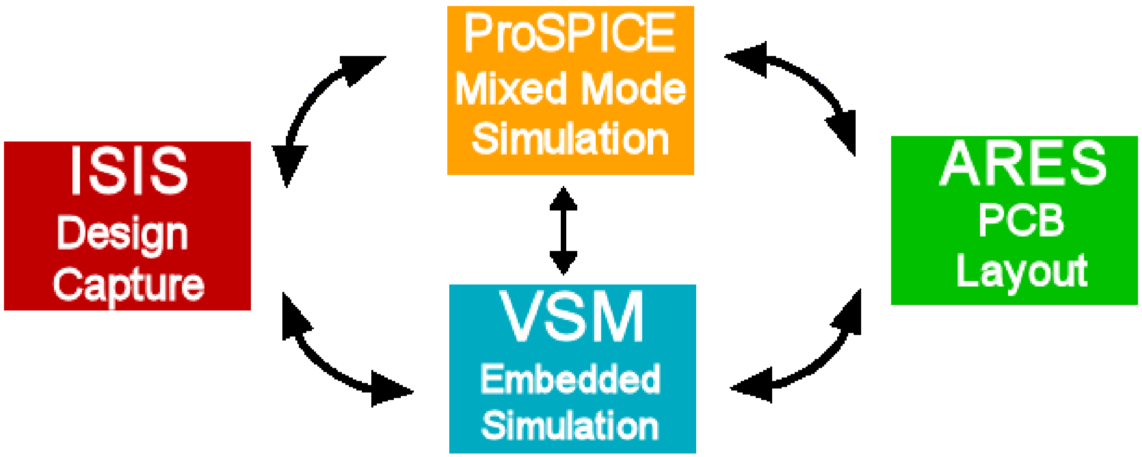
- **Development Tools**
  - Hardware Development Boards
    - **Budget is very limited**
    - **Delivery may delay us by several days**
  - Compiler
    - **MPLAB® C30 C Compiler**
    - **Student/Demo Version – Free Download**
    - **Allows immediate start**
  - Virtual Simulation and Prototyping Tools
    - **Proteus VSM – System Simulation Tool**
    - **Full Featured Demo Version – Free Download**
    - **Comprehensive range of Virtual Demo Boards**
    - **Low setup cost for Full version**

- **Application Notes and Libraries**
    - **Fast, Easy method to evaluate a technology**

Appendix A
Proteus VSM

Appendix A – Proteus VSM
Proteus Overview

An introduction to the Proteus Design Suite.



- A traditional CAD package with extra functionality for embedded systems simulation
- Allows you to simulate your PIC® microcontroller together with any analog or digital electronics connected to it
- Provides a complete software design flow for the embedded engineer

Appendix A – Proteus VSM
The Processor Models

A summary of the scope of PIC® microcontroller models available in Proteus VSM.

Well over 100 modeled PIC® microcontroller variants available :

- PIC10/ PIC12 Family :
  - 6 and 8 pin variants.
- PIC16 Family :
  - 14, 18, 28 and 40 pin variants.
- PIC18 Family :
  - 18, 28, 40, 64 and 80 pin variants.
- PIC24 Family :
  - 64, 80 and 100 pin variants.

Appendix A – Proteus VSM

The Processor Models

A summary of the functionality implemented in Proteus VSM CPU models.

- PIC® MCU Model functionality :
  - Entire instruction set including extended instruction set for appropriate variants
  - Supports all Port and other I/O pin operations
  - Supports all timers in all modes including
    - Watchdog
    - Sleep Mode
    - Wake-up
  - Supports (E)CCP modules in all modes
  - Supports Parallel Slave Port (on appropriate devices)
  - Supports MSSP module including
    - SPI (all modes)
    - I^2C™ (master and slave modes)

Appendix A – Proteus VSM
The Processor Models
A summary of the functionality implemented in Proteus VSM CPU models.

- PIC® MCU Model functionality (continued):
  - (E)USART in all modes
  - ADC Module including voltage reference pins
  - Analog Comparator Module with Internal or external reference
  - Internal Code and Data EE memory including data persistence and code protection
  - ALL Interrupt modes including priority on appropriate devices
  - I/O and other event timing accurate to one instruction cycle
  - Provides consistency checks on system operation
    - Writing to LCD Display while busy
    - Timing violation and contentions etc..
  - Extensively Tested with a suite of over 450 conformance analyses

Appendix A – Proteus VSM

The Peripheral Models

A summary of just some of the peripheral models included with Proteus VSM.

- Proteus VSM Peripheral Models :
  - Thousands of standard ‘building blocks’ – TTL/CMOS, passives, etc.
  - Interactive models for switches, buttons, pots, keypads etc.
  - OptoElectronic models.
  - Motor models and controllers.
  - Memory models.
  - Temperature Control models.
  - Real Time Clocks and Timekeeping models.
  - I^2C™/SPI Protocol models.
  - 1-Wire Protocol models
  - RS232/RS485/RS422 Protocol models.
  - ADC/DAC Converter models.
  - Pulse Width Control models.
  - Power Management models.
  - Many, many others !

Appendix B
Adding Speech

Speech Algorithms

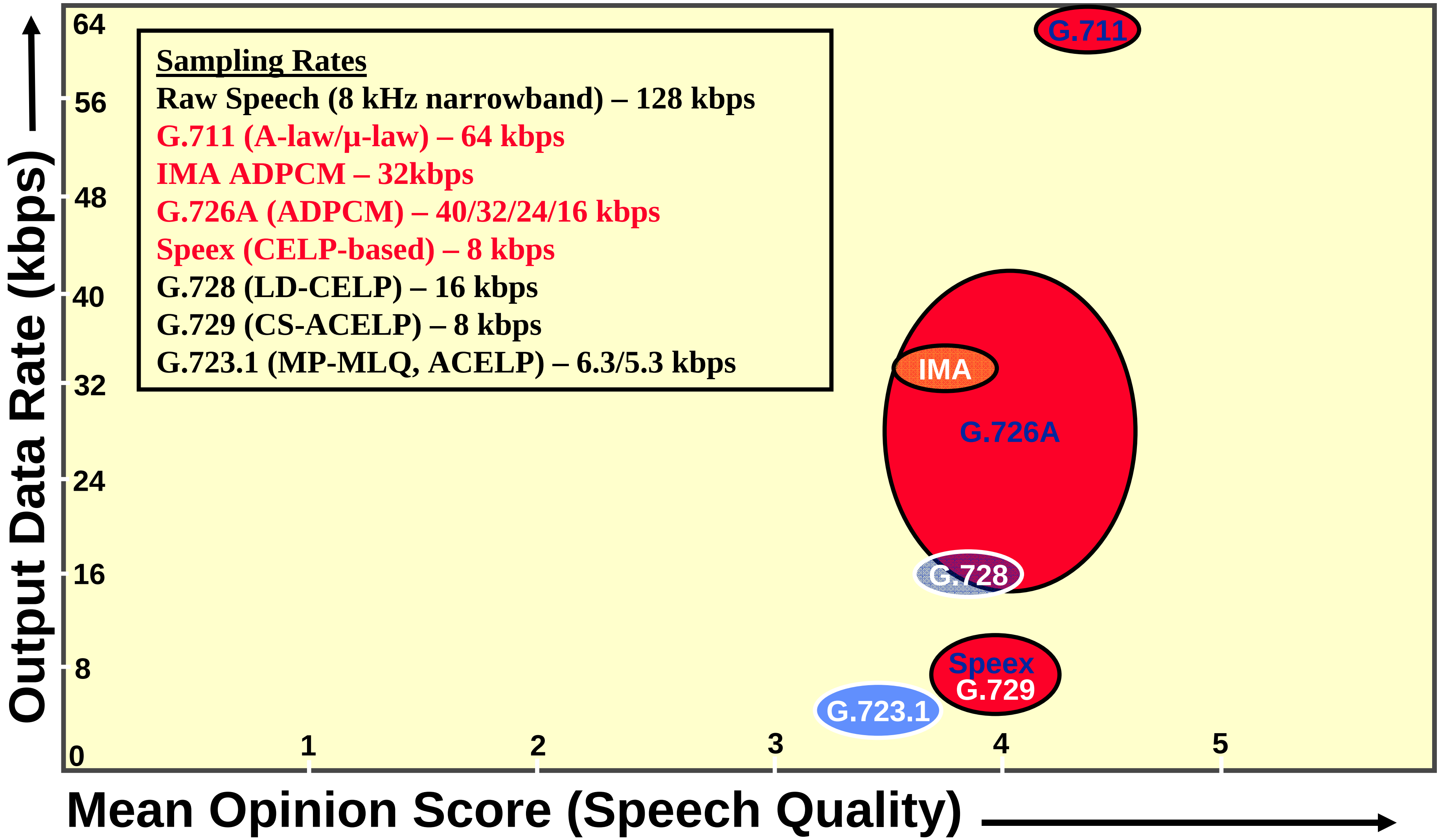
Appendix B – Adding Speech

Speech Algorithms

- **Speech normally 300 Hz to 3300 Hz**
- **Sample minimum of 8 kHz, 8-bit A/D**
  - 64 Kbits/s, ~20x bandwidth of original signal
- **Low-end operate sample to sample**
  - Quantize the difference between samples
  - Adapt quantizer relative to changes to input
- **High-end analyze frames of speech**
  - Find a codebook entry that best fits frame
- **Benefits of Speech Coding**
  - Reduces communications bandwidth
  - Reduces storage requirements

Appendix B – Adding Speech

Speech Algorithms - Quality vs. Bit Rate



Assumes 16-bit mono, 8 kHz input/output (128Kbps raw speech uncompressed)

Appendix B – Adding Speech

Speech Algorithms - Algorithm Specifications

	G.711	IMA ADPCM	G.726A	Speex
MOS	4.3-4.5	3.8	3.4-4.5	3.7-4.2
Compression Ratio	2:1	4:1	3.2:1 – 8:1	16:1
MIPS encode/decode	1	3/2	15	19/3
Flash encode/decode	3 Kb	1 Kb / 1 Kb	6 Kb	30 Kb / 15 Kb
Date Rate	64 Kbps	32 Kbps	16-40 Kbps	8 Kbps
BW reduction	50%	75%	68%- 88%	94%

Assumes 16-bit mono, 8 kHz input/output (128Kbps raw speech uncompressed)

Appendix B – Adding Speech

Available Speech Algorithms for 16-bit Devices

	24F	24H	dsPIC 30F/33F	App Note / Library	Licence Required
G.711	✓	✓	✓	✓	
IMA-ADPCM	✓	✓	✓	AN643	
G.726			✓	✓	✓
Speex			✓	✓	✓

System Considerations

Appendix B – Adding Speech
System Considerations and Trade-offs

- **Required Quality of Output –v- Memory –v- Speed –v- Cost –v- ...**
  - What is the system type?
    - **Playback only**
    - **Playback and Record**
      - Record to what memory type
    - **End to End system**
      - Same as record and playback but real time communications requirement
  - What sample rate at what bit rate is acceptable?
    - **Affects storage requirements**
    - **Has implications on compression algorithm, power consumption, processor speed….**
  - What is the sound to be played back on?
    - **Low Cost Amplifier and Speaker**
    - **Piezo Sounder**
    - **High Quality Audio**
  - What playback technology is to be used?
    - **PWM DAC**
    - **Audio Quality DAC**
    - **CODEC**

UNIVERSITY OF MICROCHIP U OF M MASTERs 2007

Appendix B – Adding Speech

System Considerations and Trade-offs

- What compression algorithm is to be used
  - **Is a royalty payment required?**
    - MP3 and some of the G.7xx algorithms are royalty based
  - **Lossy or Lossless?**
    - Does it affect quality of output when decompressed
    - Will losses be noticed
    - See MOS information for guidance
  - **Processing performance and resources needed to run algorithm**
  - **Does algorithm need DSP or standard MCU?**
- What is the power budget?
  - **Has an effect on MIPS, memory and algorithm**
  - **High sample rates require higher speed DAC's, so more current**
- What memory is available to store data?
  - **Quantity of available memory affects length/quantity of messages**
  - **May require more complex algorithms for higher compression**
  - **Is RAM required for message buffering during record**

- **Many of these factors can be tested using system simulation techniques**

Appendix B – Adding Speech
System Considerations and Trade-offs

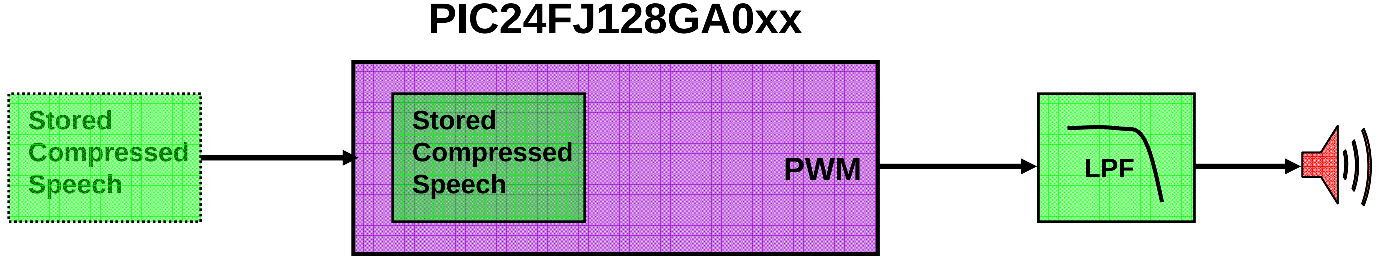
- **Types of Playback methods**
  - PWM Based DAC
    - **Single Output**
    - **Multi Output**
  - Audio DAC
  - CODEC

	Quality	Cost	Current	Comment
PWM DAC				
Single	Low to Med	Low	Low	Fs limited by PWM freq
Multi	Low to Med	Low	Low	Binary Weighted outputs
Audio DAC	Med - High	Med	Low to Med	
CODEC	High	High	Med to High	Needs I2S Port

Appendix B – Adding Speech

What is required?

- **Playback Only**
  - MCU
    - **PWM / Output Compare Module**
      - External R/C to create DAC
  - Output Filter
    - **At least 4th order Low Pass**
      - Cut-off Freq c. 4kHz
  - Memory
    - **Internal or External**
      - Will be determined by combination
        - Message length
        - Compression Algorithm
        - Recorded Sample Frequency
        - Message Construction

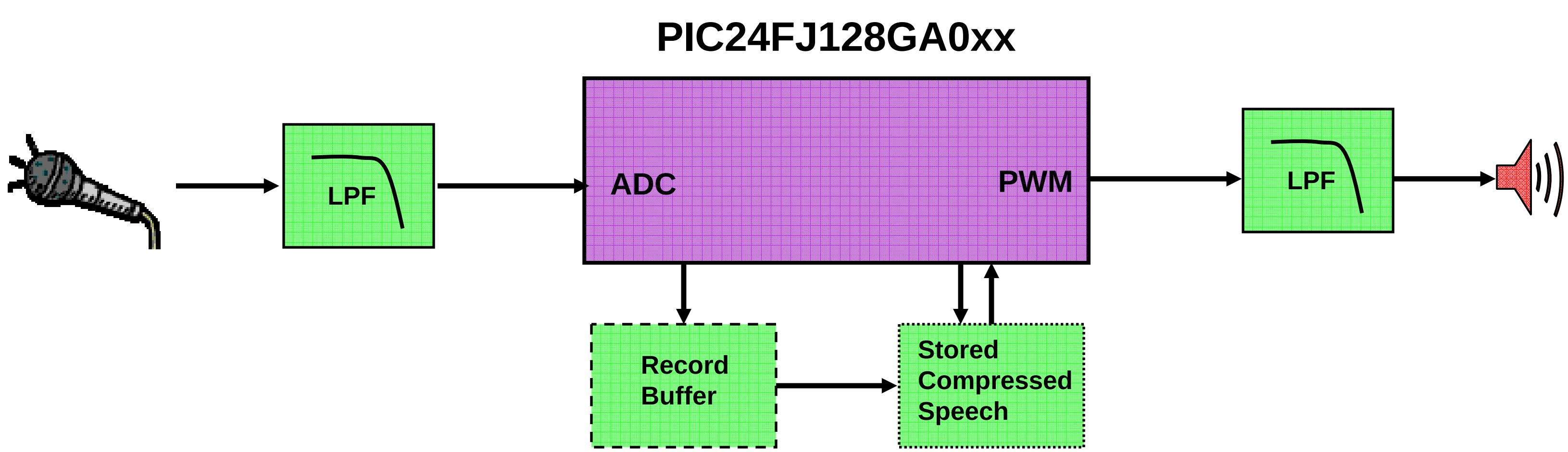


Appendix B – Adding Speech
What is required?

- **Record/Playback**
  - MCU
    - **ADC**
      - 10 or 12 bit
        - Possibly oversampled
      - Microphone Pre-Amp and Filter
    - **PWM / Output Compare Module**
      - External R/C to create DAC
  - Input Filter
    - **Microphone Pre-Amp and Filter**
      - Provides suitable bias and level shift for chosen microphone type
      - At least 4th order Low Pass
        - Cut-off Freq c.4kHz
  - Output Filter
    - **At least 4th order Low Pass**
      - Cut-off Freq c. 4kHz

Appendix B – Adding Speech
What is required?

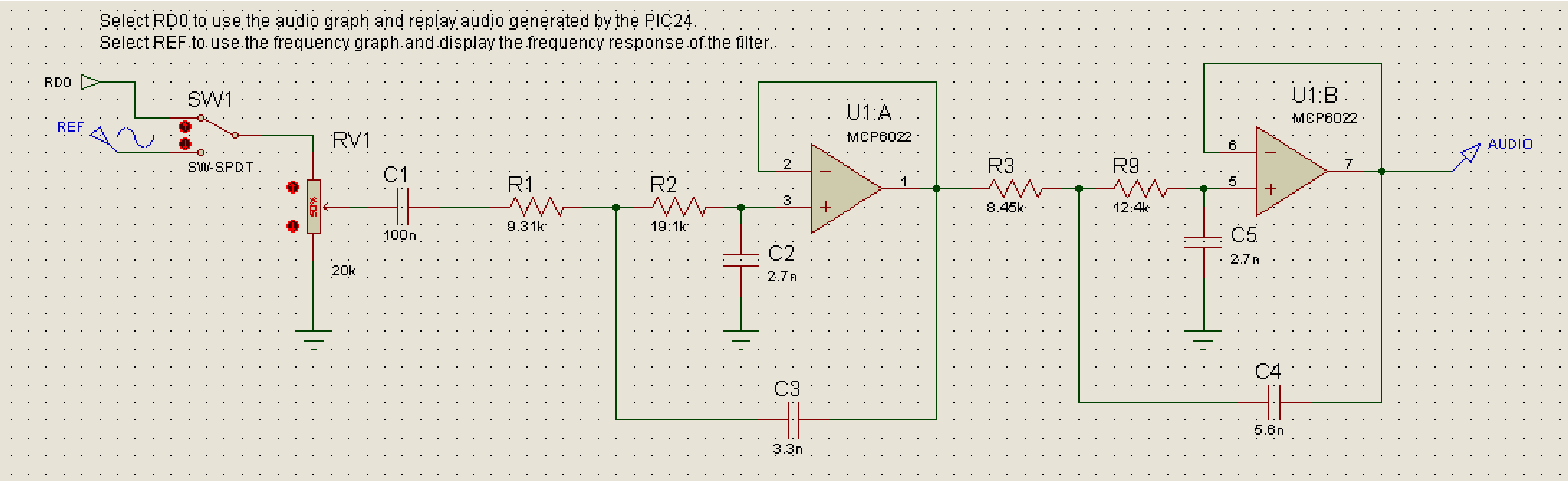
- **Record/Playback**
  - Memory
    - **Will need RAM to buffer when recording**
    - **Internal or External for storage/buffering**
      - Will be determined by a combination of
        - Message length
        - Compression Algorithm
        - Recorded Sample Frequency
        - Message Construction



PWM DAC's

Appendix B – Adding Speech
PWM DAC's

- **A Digital to Analog Converter (DAC) is created using**
  - PWM Output from Output Compare Module
    - **Frequency and Resolution are limited**
  - Analog Filter
    - **At least 4th Order Filter recommended**



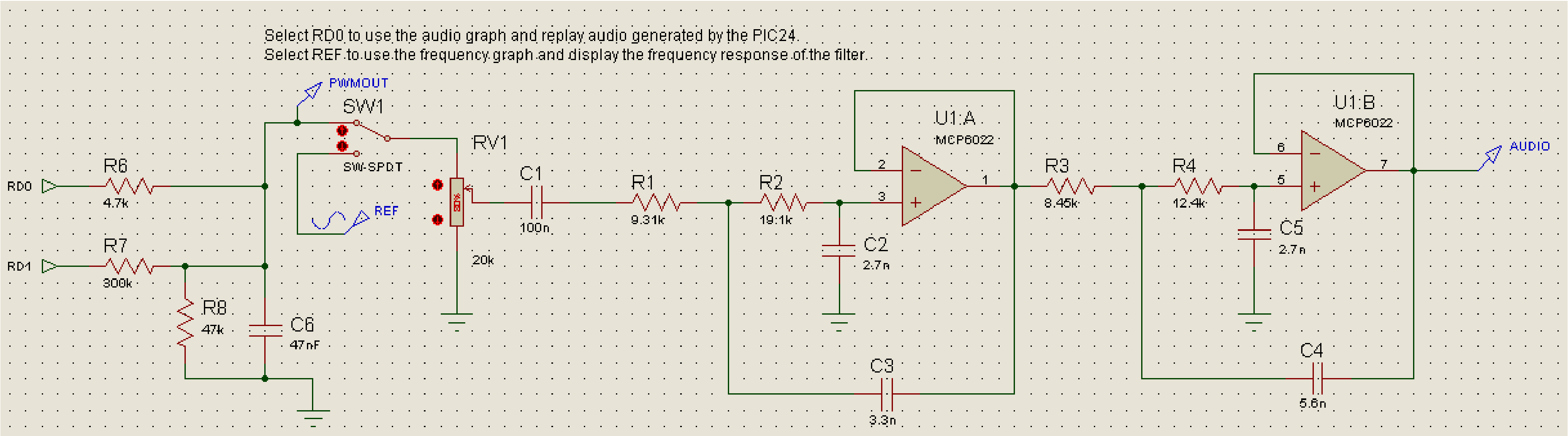
Appendix B – Adding Speech
PWM DAC's

- **Multi Output**
  - Splits number being output to PWM between 2 channels
    - **16bit number split into 2x8bit**
    - **12bit number split into 2x6bit**
  - Uses Binary Weighted output to re-combine outputs.
    - **Split into Low and High outputs**
    - **Resistors weighted**
      - LowR = 2^n x HighR
  - Allows
    - **Increased resolution at same F_{PWM} as single output**
    - **Increased F_{PWM} at same resolution**
    - **Increased F_{PWM} and increased resolution**
  - Analog Filter
    - **At least 4th Order Filter recommended**

Appendix B – Adding Speech

PWM DAC's

- **Multi Output**
  - Virtual simulation allows us to quickly and easily modify circuit to try different configurations



Recording Messages

UNIVERSITY OF MICROCHIP UofM MASTERs 2007

Appendix B – Adding Speech
Recording and Manipulating Messages

- **Messages recorded as '.WAV' files on PC**
  - Usually stored as CD Quality 44.1kHz 16bit Stereo
    - **Raw data and quality in original source**
  - Source edited to required format for compression tool
    - **Downsampled**
      - 44.1kHz to 16kHz
    - **Trimmed**
      - Lead In and Run Out trimmed to remove any excess
      - Reduces message size and hence data size
    - **Converted to Mono**
    - **Saved in required format for compression tool**
      - Save as '.RAW' unsigned PCM 16-bit little endian
      - Compress to IMA ADPCM using Microchip Winspeech utility
  - Provides a file suitable for input to filing system tool
- **'Goldwave' chosen to perform this task**
  - Excellent, intuitive editing tool
  - Provides correct output file format for use by Winspeech

Appendix B – Adding Speech
Adding a Message to Your Target

- **Data can be added as**
  - Arrays/Tables of data
    - **simplest method**
  - Files
    - **Essentially arrays but accessible via filenames**
    - **Some overhead for filing system**

- **For playback from internal memory a thin filing system used**
  - MPFS v2
    - **Used in TCP/IP Stack Application**
    - **PC Based tool for creating files**
      - Creates a <filename>.s file to be added to project
    - **Allows simple 8.3 filename access to be used on source code**
    - **Accessing multiple small files much simpler**

Appendix B – Adding Speech

How Speech Output is Constructed

- **Phrase constructed from a series of individual words**
  - In this case for 23°C, we use
    - **Twenty**
    - **Three**
    - **Degrees**
    - **Celsius**
  - Complete vocabulary stored in minimal memory
- **Vocabulary consists of individual words**
  - 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19 ,20,30,40,50,60,70,80,90,100,Degrees,Celcius, Fahrenheit

Winspeech

Appendix B – Adding Speech

Winspeech

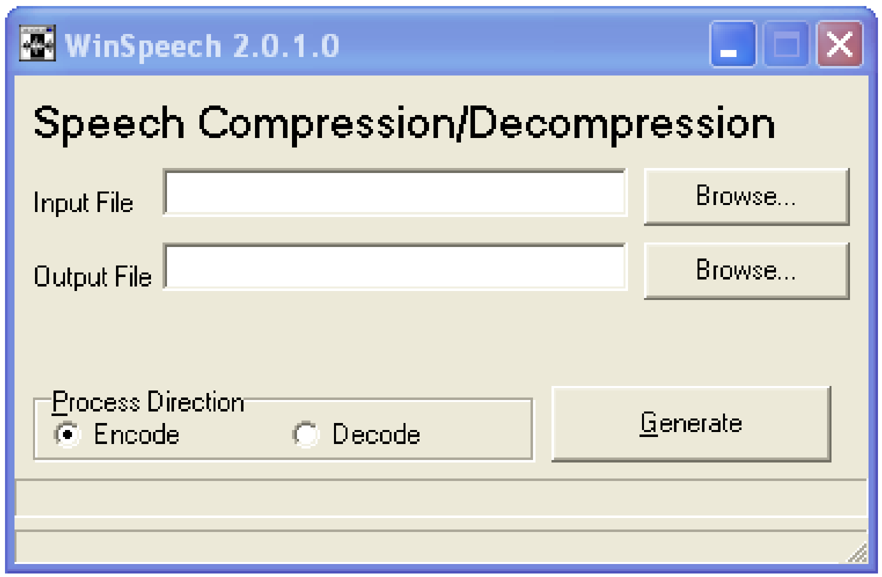
- **Winspeech is a Microchip application supporting**
  - AN643 – Adaptive Differential Pulse Coded Modulation using PIC® Microcontrollers
  - It is included in the support package for 11068 VSP
- **Winspeech performs**
  - Encode to OR Decode from a IMA-ADPCM file
  - A '.dat' filename extension is used
    - **Can be any 3 character extension**
  - Operates on a per file basis
    - **Process will need to be repeated for multiple files**

Appendix B – Adding Speech
Winspeech Input Format

- **Winspeech requires input files to be**
  - PCM unsigned 16-bit, little endian
  - Sample frequency independent
  - Suggest use of
    - **Goldwave – wave editor**
    - **Allows storage of files as ‘.raw’ with required format**

Appendix B – Adding Speech
Winspeech

- **Winspeech Interface**



Appendix B – Adding Speech
Winspeech Usage

- **Setup 3 folders**
  - ..\WAV : for source recordings pre manipulation
  - ..\RAW : for post manipulated recordings prior to input to Winspeech
  - ..\ADPCM : for '.dat' file output from Winspeech

- **For the output from the MPFSv2 utility a further file may also used**
  - ..\MPFS : for output from MPFSv2 utility
  - This can also be placed directly in target source folder

Microchip File System (MPFS)

Appendix B – Adding Speech
Filing Systems - MPFS

- **What is MPFS?**
  - How does MPFS work?
  - Where do I get MPFS?
  - Target resource usage

Appendix B – Adding Speech

Microchip File System (MPFS)

- **Small yet powerful file system**
- **Flexible storage scheme**
  - Internal program memory or external data EEPROM
- **PC based utility to generate MPFS image**
- **8.3 short file names**
- **Case-insensitive file names**
- **Read AN833 for more detail**

UNIVERSITY OF MICROCHIP UofM MASTERs 2007

Appendix B – Adding Speech
MPFS Variants

- **Two variants**
  - MPFS
    - **Command Line based**
    - **Suitable for MPLAB® C18 C Compiler**
    - **Will create images for**
      - Internal Program Memory : <filename>.c
      - External EE memory : <filename>.bin
  - MPFSv2
    - **Graphical Interface**
    - **Suitable for MPLAB C30 C Compiler**
    - **Will create images for**
      - Internal Program Memory : <filename>.s

- **Created for TCP/IP stack to aid storage of web pages**
  - Suitable as filing system for other tasks
  - Image size must fit in available memory
  - All files to be added to image must be within same folder
  - “CR LF” stripped from “*.htm” files
  - Reserved block for application specific data

Appendix B – Adding Speech
MPFSv2 Image for MPLAB C30

- **Image is created as '.s' ASM30 file**
  - Data is created as a packed table
  - 3 bytes per program memory address
  - Data accessed via Table Read and Write instructions
- **Basic FAT table created**
  - Allows acces of data table using 8.3 short filename
  - FAT table references filename to a memory location

```
.byte 0x00,0x00
.long paddr(_MPFS_0000)
.byte '1','.','D','A','T', 0 , 0 , 0 , 0 , 0 , 0 , 0
```

Appendix B – Adding Speech
MPFSv2 Image for MPLAB C30

- **FAT table performs cross reference to data table**

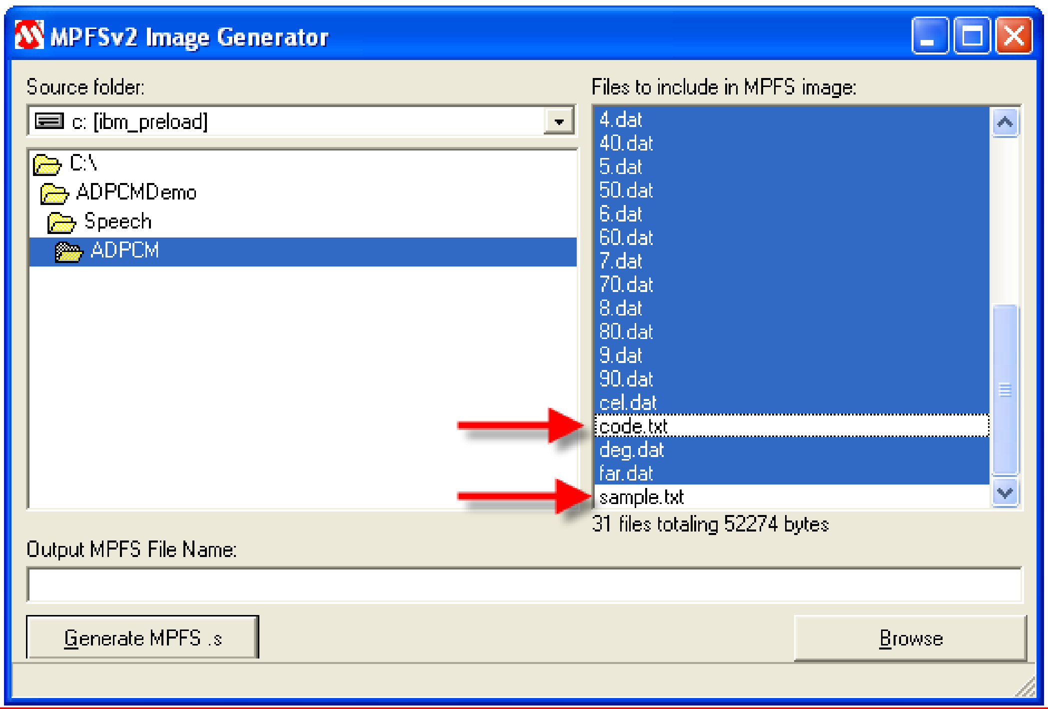
```
;*****************************************************************************************
; Original Filename: 1.dat
; MPFS 8.3 Filename: 1.DAT
;*****************************************************************************************
    goto END_OF_MPFS_0000                ; Prevent accidental execution
                                         ; of constant data.
    .global _MPFS_0000
_MPFS_0000:
    .pbyte 0xFC,0x10……………..
    .pbyte …………………………..
    .pbyte 0x04,0xFF,0xFF,0xFF,0xFF      ; MPFS_ETX, MPFS_INVALID
    END_OF_MPFS_0000:
```

Appendix B – Adding Speech
Using MPFSv2

- **MPFSv2 will**
  - Add all selected files
  - In a single folder
  - To a specified image <filename>.s file

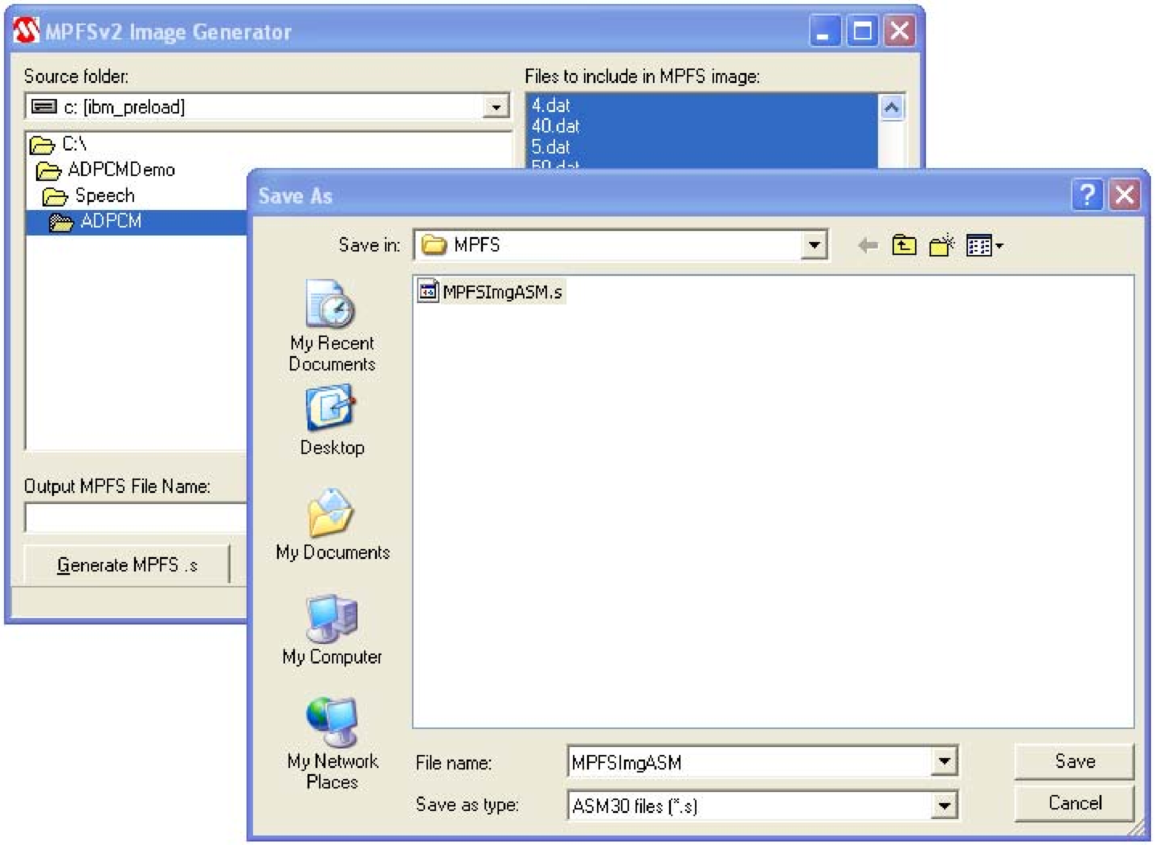
Appendix B – Adding Speech
Using MPFSv2

- **Select source folder**
  - Add all required files
  - Ensure files not required are de-selected
  - Files not required will consume valuable memory resource



Appendix B – Adding Speech
Using MPFSv2

- **Browse for Destination Folder**
  - Add / Create destination file
  - Generate image file



Appendix B – Adding Speech

Using MPFSv2

- **On creation of <filename>.s**
  - Add file to project tree in MPLAB® Project
- **MPLAB Project also requires**
  - Target source files for MPFS filing system
    - **C:\Microchip Solutions\Microchip\Include\TCPIP Stack\mpfs.h**
    - **C:\Microchip Solutions\Microchip\TCPIP Stack\mpfs.c**
- **MPFSv2 application can be found in the TCP/IP stack distribution installation**
  - C:\Microchip Solutions\Microchip\TCPIP Stack\mpfsv2.exe

Appendix C
Adding Mass Storage

Design Considerations

Appendix C – Adding Mass Storage
What options are available?

- **Serial**
  - Data EE
  - Serial Flash / Data Flash
  - SD/MMC Card
- **Parallel**
  - Parallel Flash
  - Compact Flash
  - PATA HDD

Appendix C – Adding Mass Storage
What needs to be considered?

- **What is the amount of memory required?**
- **Is high endurance required?**
  - No. of E/W cycles expected
- **What is the power budget?**
  - Is a low power sleep mode required
  - Access speed is a factor
- **What system reliability is needed?**
  - Some technologies require a card frame
    - **Potential reliability issues in some environments**
  - Directly soldered devices can be limited in memory size
- **What filing systems can be used?**
  - For large or multiple data sets a filing system is advisable
  - Allows for portability in removable memory cards

UNIVERSITY OF MICROCHIP
UofM
MASTERs 2007

Appendix C – Adding Mass Storage
What are the Options?

- **Comparison of available technologies**

	Voltage	Current			Capacity max.	Endurance Max E/W	Cost	Comment
		Read	Write	Standby				
Serial								
Data EE	1.8 - 5.5v	7mA	6mA	10uA/1uA	1Mb/128kB	1M	$	Current for SPI EE
Serial Flash	2.7 - 3.6v	10mA	15mA	2uA	16Mb/2MB	10k - 100k	$	
Data Flash	2.7 - 3.6v	10mA	25mA	25uA	64Mb/8MB	100k	$	
SD/MMC	2.0 - 3.6v	15mA/30mA	15mA/30mA	300uA	2GB	n/a	$$	25MHz/50MHz, SDHC upto 32GB
Parallel								
Parallel Flash	2.7 - 3.6v	10mA	25mA	15uA	64MB/8MB	n/a	$	
Compact Flash	3.3 or 5v	60mA	60mA	600uA	16GB	n/a	$$	FAT32 required above 2GB
ATA HDD	5v	360mA	360mA	20mA	160GB	n/a	$$$	FAT32 required above 2GB, Max160GB for 2.5" PATA, Spin Up Current c.1A

Appendix C – Adding Mass Storage
What Proteus VSM Models are available?

- **Serial**
  - Data EE
    - **Multiple Vendors**
    - **I²C™ and SPI**
  - Serial Data Flash
    - **NEW – 4Mbit AT25F4096**
  - SD/MMC Card
    - **Models MMC Specification**
    - **Uses binary image file**
- **Parallel**
  - Compact Flash
    - **ATA Mode**
    - **Uses binary image file**
  - ATA HDD
    - **Uses binary image file**

FAT16 File System

Appendix C – Adding Mass Storage
Filing Systems – FAT16

- **Standard format from Microsoft**
  - Readable by PDAs, PCs, etc.
  - **Requires a license or waiver from Microsoft**
- **Addresses up to 2 Gbyte**
- **Memory usage**
  - ~14K bytes of program memory, includes card interface
  - ~650 bytes of RAM
- **Supports `card detect`, `fopen`, `fclose`, `fread`, `fwrite`, `rewind`, `remove`, etc.**

Appendix C – Adding Mass Storage
Filing Systems – FAT16

- **Disk Organization**
  - Fundamental unit is sector, typ. 512 bytes
    - **1 Mbyte = 2048 sectors**
    - **20 Gbyte = >40 Million sectors**
  - Group sectors into blocks called clusters
- **Structures**
  - Master Boot Record (MBR)
  - Boot Sector
  - Root Directory
  - File Allocation Table

Appendix C – Adding Mass Storage

Filing Systems – FAT16 File Allocation Table

- **The table has one 16-bit entry (hence, FAT16) for every cluster on the disk**
- **Each entry either points to the next cluster in a file or contains a special value**

Value	Meaning
0x0000	Cluster is available
0x0001	Reserved Cluster
0x0002 - 0xFFEF	Used cluster. Value points to next cluster
0xFFF0 - 0xFFF6	Reserved Values
0xFFF7	Bad Cluster
0xFFF8 - 0xFFFF	Last cluster in file

Appendix C – Adding Mass Storage

Filing Systems – FAT16 Example

Root Directory

Entry A
Entry B

FAT

Position	Value
1	0x0004
2	0x0003
3	0xFFFF
4	0xFFFF

Clusters
File 3, Part 1
File 2, Part 1
File 2, Part 2
File 3, Part 2

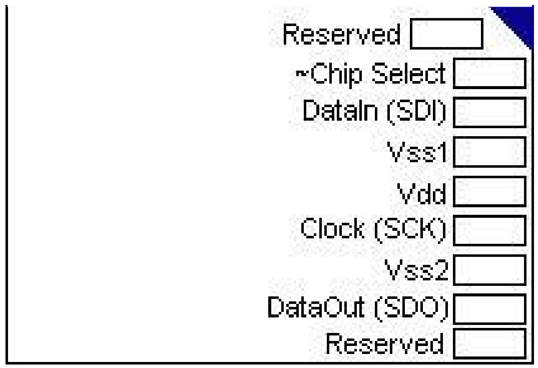
References – FAT

- **Information sources related to FAT filing system**
  - Microsoft FAT32 Filing System Specification
    - **http://www.microsoft.com/whdc/system/platform/firmware/fatgen.mspx**
  - Jan Axelson's USB Mass Storage
    - **http://www.lvr.com/mass_storage.htm**

Flash Memory Cards

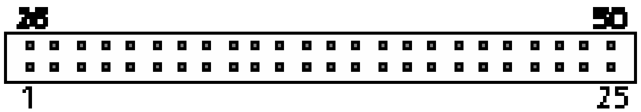
Appendix C – Adding Mass Storage
Flash Memory Cards – Secure Digital Cards

- **64 Mbyte @ $15**
  - Up to 2 Gbytes
- **Communications**
  - 1 wire SPI
- **Additional Pins**
  - Power/Ground
  - Write Enable
  - Card Detect
- **Requires a license from SD Card Association**
  - http://www.sdcard.org/



Appendix C – Adding Mass Storage
Flash Memory Cards – CompactFlash® Card Interface

- **64 Mbyte @ $12**
  - Up to 8 Gbytes
- **Same as PCMCIA-ATA**
  - 3.3 mm or 5 mm thick
- **Communications**
  - Memory mapped address
- **Uses 3 I/O pins + PMP**
  - 8 data, 3 control, 4 address
  - Card Detect, Rdy/Busy, Reset (I/O pins)
- **Requires a license from CompactFlash Association**
  - http://www.compactflash.org/



Virtual Mass Storage Devices

Appendix C – Adding Mass Storage
Virtual Mass Storage Devices

- **VSM provides a variety of models**
  - Can be used raw or formatted
- **No drive letter allocated on PC**
  - Virtual Drives DO NOT have drive letter associated with them
  - Interaction is via a binary image file
  - Created and accessed on PC
    - **Use a Hex Editor to view contents**
    - **FATutil - DOS Utility to Add, Delete or Append image file**

Appendix C – Adding Mass Storage Using FATutil

- **DOS command line interfaceto binary image file**
  - Allows read, write, erase

- **To view an image file a Hex editor whould be used**
- **Only single file operations allowed**

FATUTIL - Read or write to FAT16 binary images.
Usage: FATUTIL [imagefile [Actions...]...]
Actions:
/w filename   Write file to image.
/r filename   Read file from image.
/e filename   Erase file from image.

Appendix C – Adding Mass Storage Using FATutil

- **Example command line operations**

Example (single operation):

To Write <filename>.dat to the image file CFimage.bin, where <filename>.dat and image.bin are located in the same folder

- fatutil CFimage.bin /w <filename>.dat

To Read <filename>.dat from the image file CFimage.bin, where <filename>.dat and image.bin are located in the same folder

- fatutil CFimage.bin /r <filename>.dat

To Erase <filename>.dat from the image file CFimage.bin, where <filename>.dat and image.bin are located in the same folder

- fatutil CFimage.bin /e <filename>.dat

Appendix C – Adding Mass Storage Using FATutil

- To perform multiple operations the MS-DOS® 'FOR' command is used.

To Write to <filename>.bin with multiple files a wildard (*) can be used.

FOR %i IN (*.dat) DO FATUTIL <filename>.bin /w %i

Requires FATUTIL and image file (<filename>.bin) to be located in same folder. If path is used in (set) eg.

- FOR %i IN (c:\Explorer-16\Demo4\Adpcm*.dat) DO FATutil /w <filename>.bin

FATutil will try to write to image file in the same folder. Therefore ensure you locate the image file to be appended to the same working folder as the dat files

Read and erase operations operate in a similar manner.

Appendix D
Adding Ethernet

Appendix D – Adding Ethernet

What are we going to do?

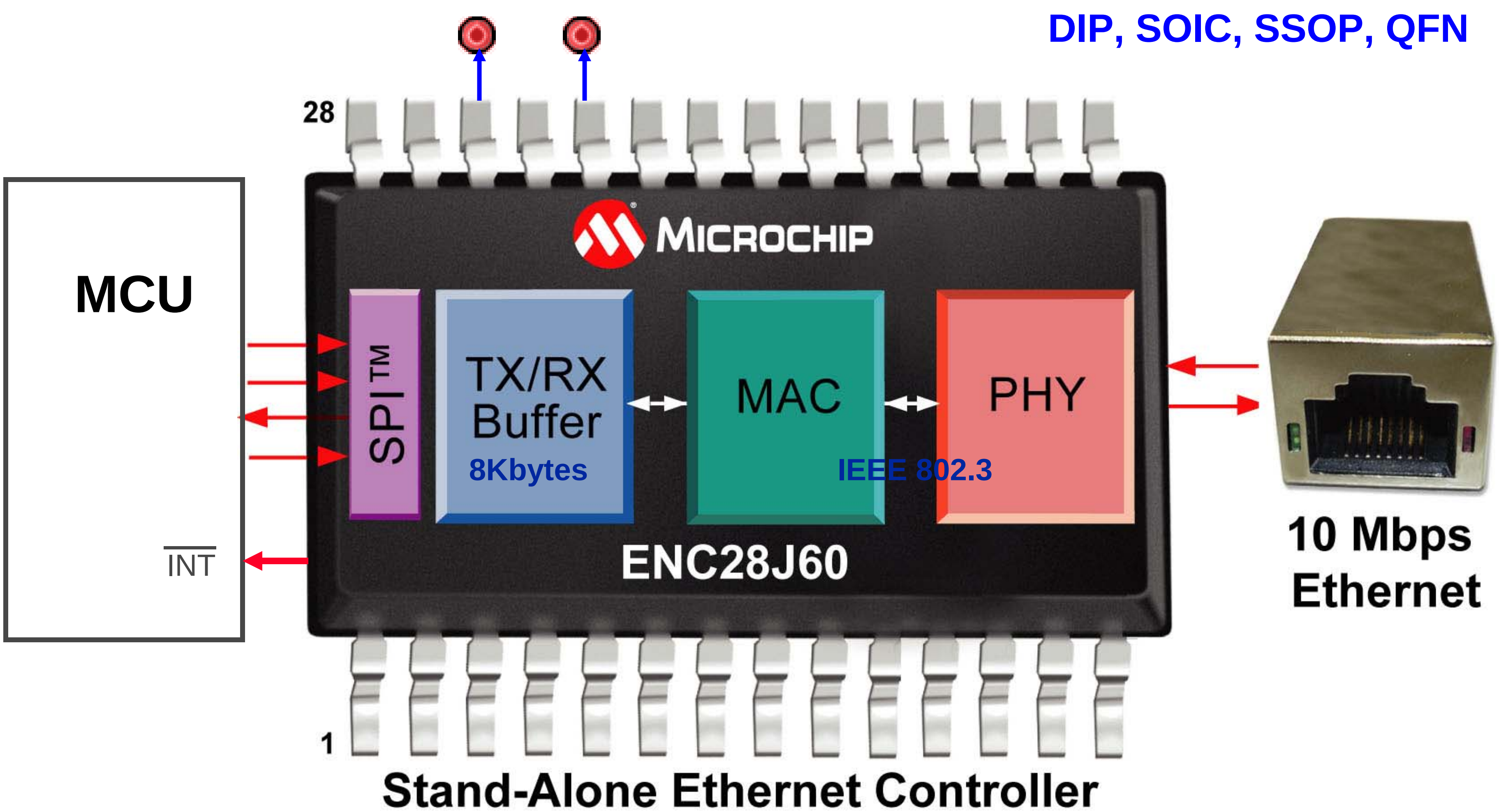
- **Add Ethernet to design**
  - Use Microchip TCP/IP library to evaluate internet capabilities
  - Use basic interaction concepts in TCP/IP Demo
    - **Little time available to create any custom code or web pages**

- **Provides a platform to evaluate**
  - Flexibility of permanent wired internet connection
  - Implications for Embedded Device
  - Microchip TCP/IP Library and feature enhancements this can provide the product

- **Introduces additional software tools**
  - IFLIST – NIC enumeration tool for Virtual ENC28J60 in Proteus VSM
  - Microchip Ethernet Discoverer – Determine Microchip devices on a network
  - Wireshark – Ethernet packet sniffer
  - WinPcap – Packet Capture Interface

Appendix D – Adding Ethernet

The ENC28J60 Ethernet Controller



Appendix D – Adding Ethernet
Connecting to the Real World

- **How does Virtual Ethernet Work?**
  - WinPcap
    - **Packet Capture Interface and Filtering Interface**
    - **Allows applications to capture and transmit network packets bypassing the protocol stack**
    - **www.winpcap.org**
  - Virtual ENC28J60
    - **Uses WinPcap to interface directly to PC NIC**
    - **Allows PC to run normal network traffic and Virtual Demo Board simultaneously**
    - **Active NIC must be selected in device properties**
      - Use IFLIST.exe to identify your active NIC
      - Set NIC number in ENC28J60 Device Properties
    - **Utilises Microchip TCP/IP Stack**
      - Provides fast, easy method to evaluate TCP/IP connectivity in a system design

Microchip TCP/IP Stack

Appendix D – Adding Ethernet
The Microchip TCP/IP Stack

- **Stack Details**
  - Stack is Application Note based – AN833
  - Fully Supported by Microchip Technical Support
  - NO ROYALTY or LICENCING FEES
    - **Licence restricts use to a PIC® microcontroller or dsPIC® digital signal controller**
  - Modular
    - **Only relevant functions need to be used**
    - **Some functions are dependant on others !!**
  - Features Offered
    - ARP
    - IP
    - ICMP
    - UDP
    - TCP
    - DHCP
      - Client and Server
    - SNMP
    - HTTP
    - FTP
    - TFTP

Appendix D – Adding Ethernet

Connecting to the Internet with TCP/IP Stack

- **Resources and Layers**
  - **Details of stack modules, size and relative layer**

Application	HTTP (3.7K bytes)	SMTP (3.8K bytes)	DHCP (1.9K bytes)	DNS (1.5K bytes)
Transport	TCP (11.5K bytes)		UDP (2K bytes)	
Internet & Network Access	IP (874 bytes), ARP (896 bytes)			
Physical	Ethernet – ENC28J60 (3.8K bytes)			

Trademarks

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KeeLoq, KeeLoq logo, microID, MPLAB, PIC, PICmicro, PICSTART, PRO MATE, rfPIC and SmartShunt are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AmpLab, FilterLab, Linear Active Thermistor, Migratable Memory, MXDEV, MXLAB, SEEVAL, SmartSensor and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, ECAN, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, PICkit, PICDEM, PICDEM.net, PICLAB, PICtail, PowerCal, PowerInfo, PowerMate, PowerTool, REAL ICE, rfLAB, Select Mode, Smart Serial, SmartTel, Total Endurance, UNI/O, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.