Application Note

AN2202/D
5/2002

Creating a Graphical User Interface (GUI) for the MC3PHAC

By: Steven Torres
Motorola
SPS TSPG 8-/16-Bit Division

Introduction

The MC3PHAC is a complete solution for three-phased AC motor control applications. The MC3PHAC contains high-end features such as three-phase waveform generation, dynamic bus ripple cancellation, and a DSP-filtered speed reference signal.

Using the MC3PHAC reduces development time and shortens time-to-market. Moreover, the MC3PHAC has the flexibility to fit various design implementations. In particular, the MC3PHAC uses two modes of operation:

- Stand-alone mode — In stand-alone mode, the MC3PHAC is controlled via the hardware settings.
- External master mode — In external master mode, the MC3PHAC is software controlled via an external master.

This application note concentrates on the external master mode of operation for the MC3PHAC. It provides a detailed discussion of the external master functional requirements. The discussion focuses on the use of a PC as an external master and on a methodology for creation of a graphical user interface (GUI) on the PC to control the MC3PHAC. The last section of this application note presents and reviews an example of a GUI interface for the MC3PHAC.

Information presented here comes from several sources. These sources include:

- *PC Master Software User Manual* (available at http://motorola.com/sps)
- *PC Master Software Communication Library* (available at http://motorola.com/sps)
- *MC3PHAC Advance Information* data sheet (Motorola document order number MC3PHAC/D)
- *General-Purpose 3-Phase AC Industrial Motor Controller Reference Design: Designer Reference Manual* (Motorola document order number DRM006/D)
- Various HTML and ActiveX references

This application note takes a cross section of the sources, extracts necessary material from these documents, and puts the information in the appropriate context to provide a guide for creating a GUI for the MC3PHAC.

More About MC3PHAC External Master Mode

Several features make controlling the MC3PHAC in the external master mode desirable. First of all, in external master mode, a developer has more control over the individual variables for motor control than in stand-alone mode. In addition, in external master mode, a developer has access to variables not available in stand-alone mode. More information on external master mode is available in the *MC3PHAC Advance Information* data sheet (Motorola document order number MC3PHAC/D).

MC3PHAC Coding Framework

To take-up the task of coding a GUI for the MC3PHAC, a high-level understanding of the external master system configuration is required. This includes learning the protocol for the communication between the devices, the commands available on the MC3PHAC, and the user interface variables on the MC3PHAC. With this background, a GUI developer can best determine, for a particular application, what is the best GUI development environment for the project.

The GUI environment can include tools like C++, Visual Basic, HTML, ActiveX, and Visual Basic Scripting. These GUI development tools must implement all the required elements for the MC3PHAC coding framework. These requirements are detailed in the next sections.

Motor System Overview

Before exploring the methodology of creating a GUI for a MC3PHAC in external master mode, it is important to understand how the GUI fits in an MC3PHAC motor control system. Figure 1 illustrates three major components of a motor control system:

- Motor
- Motor control board
- External master

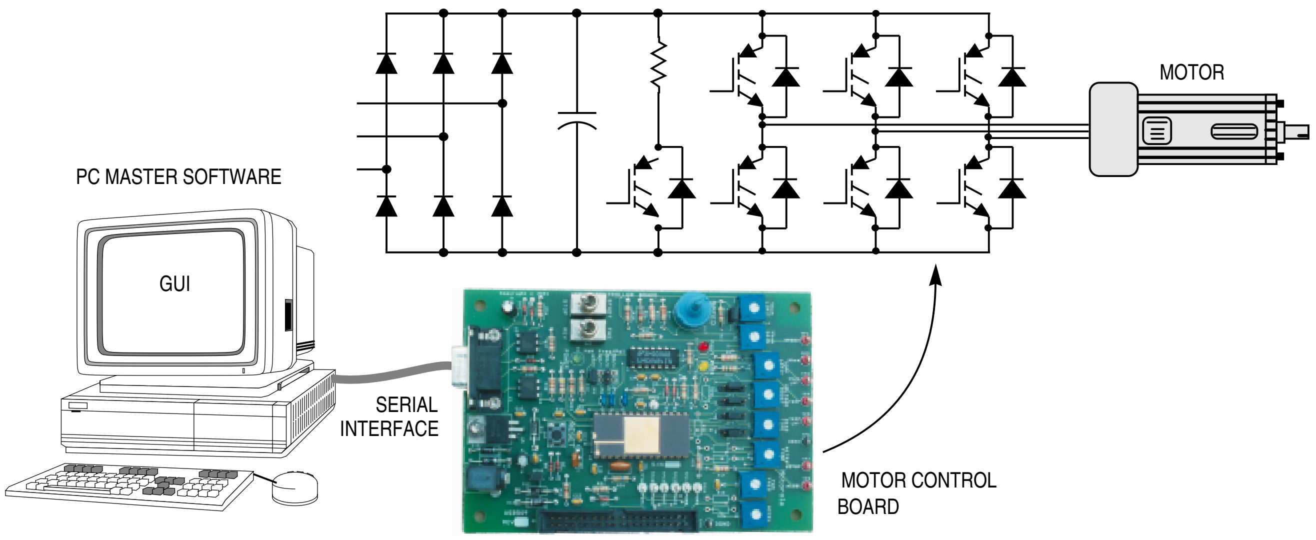


Figure 1. System Overview of MC3PHAC Controlled by an External Master

GUI and External Master

Figure 1 shows that the GUI resides on the PC. In this case, the PC is the external master; however, the external master can also take on other forms, another MCU for instance. For this implementation, the external master is an IBM-compatible PC running Windows 2000®. Other versions of Windows should operate similarly but screen shots may differ slightly. While the PC has two available serial ports, only one is needed to interface to the MC3PHAC. Whatever the form of the external master, it must be programmed to implement the PC master software protocol over a standard UART (universal asynchronous receiver/transmitter).

MC3PHAC Reference Design

A reference design for a motor control board is available for demonstration of the MC3PHAC (see *General-Purpose 3-Phase AC Industrial Motor Controller Reference Design: Designer Reference Manual* — Motorola document order number DRM006/D). The MC3PHAC reference design demonstrates the hardware implementation and integration of the MC3PHAC.

At the heart of the MC3PHAC reference design is the MC3PHAC. Once the external master sets the MC3PHAC through the GUI, the MC3PHAC automatically makes the necessary adjustments to achieve the new settings. The reference design was used in the development of the external master GUI demo presented in the last section of this application note.

External Master Communication Components

Looking at **Figure 1**, it is important to understand how the external master and a reference design are connected and how they communicate with each other. If not provided by an application like the PC master software, these communication protocols will need to be coded along with the GUI.

Understanding how the devices are connected is straightforward. A serial communication cable connects the devices. The requirement for the serial port is that it communicates at 9600 baud with no flow control.

The communication mechanism between the two devices requires a more detailed explanation. The communication requirements dictate how and what can be communicated between the devices. The three components of the communication mechanism are:

- PC master software protocol
- PC master software commands
- MC3PHAC user interface variables

To assist in the understanding of the mechanism requirements, a detailed explanation of both of these items is provided in the next section.

PC Master Software Protocol

The basic requirement for communication between the external master and the MC3PHAC is that the external master must implement the PC master software communication protocol. Some details of the PC master software protocol are presented next. A complete description of the protocol can be found in the *PC Master User Guide*.

The principal behind the protocol involves encoding and decoding messages to and from the MC3PHAC. This encoding and decoding ensures a specifically organized data packet that is used as a common language between the devices. In the protocol, two data packets are defined:

- Command packet
- Response packet

The basic communication mechanism for the devices is that the external master requests an action. Then the MC3PHAC must reply if the action was completed successfully. In other words, the external master sends out a command packet and waits for a response packet from the MC3PHAC.

Command Data Packet Protocol — Figure 2 illustrates the command data packet protocol structure.

start-of-message (1 BYTE)	command (1 BYTE)	data part (known length)	checksum (1 BYTE)

Figure 2. Command Data Packet Protocol

Figure 2 shows that the command data packet protocol consists of four components:

- Start of message (SOM) — A one-byte special character defined as ASCII '+' code (0x2b); the SOM indicates the beginning of a command data packet
- Command — A one-byte command code that is defined by the PC master software protocol
- Data Part — Data to be transmitted
- Checksum — A one-byte two's complement checksum; it is computed by taking the two's complement of the sum of all bytes of a message after the SOM. The checksum is used to verify that the data packet is not corrupted.

A GUI interface must adhere to this data packet structure when encoding and sending a command to the MC3PHAC.

Response Data Packet Protocol — Figure 3 illustrates the response data packet protocol structure.

start-of-message (1 BYTE)	status code (1 BYTE)	data part (known length)	checksum (1 BYTE)

Figure 3. Response Data Packet Protocol

Figure 3 shows that response data packet protocol consists of these:

- Start of message (SOM) — A one-byte special character defined as ASCII '+' code (0x2b); the SOM indicates the beginning of a response data packet
- Status Code — A one-byte operation status code that describes the success or failure of the most recent command
- Data part — Variable length data; the length depends on the status code value
- Checksum — A one-byte two's complement checksum; it is computed by taking the two's complement of the sum of all bytes of a message after the SOM. The checksum is used to verify that the data packet is not corrupted.

A GUI interface must adhere to this data packet structure when decoding messages received from the MC3PHAC.

Two other features of the protocol coding are the checksum and SOM. These features need to be coded as dictated by the *PC Master Communication Library*. While the SOM is easy to define, the checksum requires calculation. Again, the checksum is computed by taking the two's complement of the sum of all bytes of a message after the SOM.

Another feature of the protocol that must be considered is the SOM value, 0x2B, appearing as a checksum value or a parameter in the data packet. If this happens, the protocol would mistake the value as a new data packet. To avoid this problem, the SOM value must be repeated in the data packet. These subtleties and more can be reviewed further in the *PC Master Protocol Library*.

Valid PC master software Commands

The protocol also defines valid commands between the devices. Although the complete PC master software protocol is very extensive, for the MC3PHAC, only a handful of the PC master software commands have been implemented. **Table 1** lists the valid commands available on the MC3PHAC. In the PC master software protocol, commands are identified by a code (see **Table 1**). The commands are passed back and forth between the devices embedded in data packets. The data packets are decoded and executed.

Table 1. MC3PHAC Valid PC Master Software Commands

Command	Code	Description
READVAR8	0xd0	Read BYTE variable
READVAR16	0xd1	Read WORD variable
READVAR32	0xd2	Read DWORD variable
GETINFOBRIEF	0xc8	Retrieve a subset of board information structure
WRITEVAR8	0xe3	Write BYTE variable
WRITEVAR16	0xe4	Write WORD variable

User Interface Variables

The MC3PHAC gives a GUI developer access to about twenty-five variables that can be used to control or monitor various aspects of a motor. Again, this is far more than are available to a developer in stand-alone mode. **Table 2** itemizes each of the user interface variables.

Each variable has a specific format that must be followed by the GUI interface. User interface variables can be read only, write only, or have read/write capability. In addition, the size and the type of the variable can vary. Variables can be one, two, or four bytes. They can also either be an integer, real, or a discrete value. While some of the details of the user interface variables are provided below, a complete description of each variable is provided in the MC3PHAC data sheet.

Table 2. MC3PHAC User Interface Variables

MC3PHAC Variable	Address		Size (Bytes)
Commanded direction	0x1000	W	1
Command reset	0x1000	W	1
PWM frequency	0x1000	W	1
Measured PWM period	0x00A8	R	2
PWM polarity	0x1000	W	1
Dead time	0x0036	R/W	1
Base speed	0x1000	W	1
Acceleration	0x0060	R/W	2
Commanded frequency	0x0062	R/W	2
Actual frequency	0x0085	R	2
Status	0x00C8	R	1
Voltage boost	0x006C	R/W	1
Modulation index	0x0091	R	1
Maximum voltage	0x0075	R/W	1
Bus voltage	0x0079	R	2
Fault timeout	0x006A	R/W	2
Fault timer	0x006D	R	2
V_{Bus} deceleration value	0x00C9	R/W	2
V_{Bus} R_{Brake} value	0x0064	R/W	2
V_{Bus} brownout value	0x0066	R/W	2
V_{Bus} overvoltage value	0x0068	R/W	2
Speed in ADC value	0x0095	R	2
Setup	0x00AE	R	1
Switch in	0x1000	R	1
Reset status	0xFE01	R	1
Version	0xEE00	R	4

Communication Protocol Example — Below is an example of the protocol. The example shows the encoding and decoding of the data packets required to change the commanded motor frequency of the MC3PHAC.

In addition, this example illustrates software handshaking internal to the PC master software protocol. For every command sent out to the MC3PHAC, a predefined response is expected. All these features need to be considered in the GUI code development. The GUI interface must follow the protocol.

Task: Change Commanded Frequency

- Encoded out-going command packet: 2B E4 00 62 60 00 5A

0x2B	start-of-message character (SOM) = ASCII '+'
0xE4	Command code to write 2-byte variable
0x0062	Commanded frequency address interface
0x6000	New commanded frequency value
0x5A	Checksum

- Incoming response data packet to decode: 2B 00 00

0x2B	start-of-message character (SOM) = ASCII '+'
0x00	Response: operation successful
0x00	Checksum

Why Use the PC Master Software Application for GUI Creation?

The **External Master Communication Components** section describes the communication protocol required by an external master and its GUI interface. In developing a GUI interface, these requirements, specifically the communication link, PC master software commands, and the PC master software protocol, must be implemented in code.

To streamline the GUI development process, use the PC master software application. The PC master software is a free application that reduces the coding requirements. The PC master software application provides UART communication, Internet capability, built-in commands, and data packet packaging, without requiring the writing of code.

In addition, the PC master software provides a detailed-view frame for the creation of a GUI, and provides an ActiveX component to access the PC master software commands via the HTML frame. Moreover, the PC master software has additional features such as a scope to plot real-time MC3PHAC variables and a watch-grid pane to monitor the changing values of the variables. For more information on the operation and configuration of the PC master software capabilities, reference the *PC Master User Manual*.

Even more impressive is the capability of a PC master software application to be operated by a remote computer over the Internet. The server capability of the PC master software application will be covered in another application note. **Figure 4** illustrates the possible Internet connectivity for the MC3PHAC device.

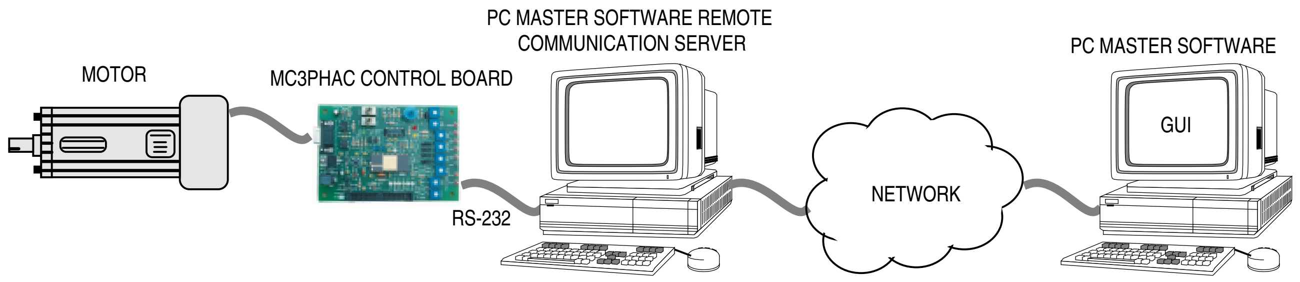


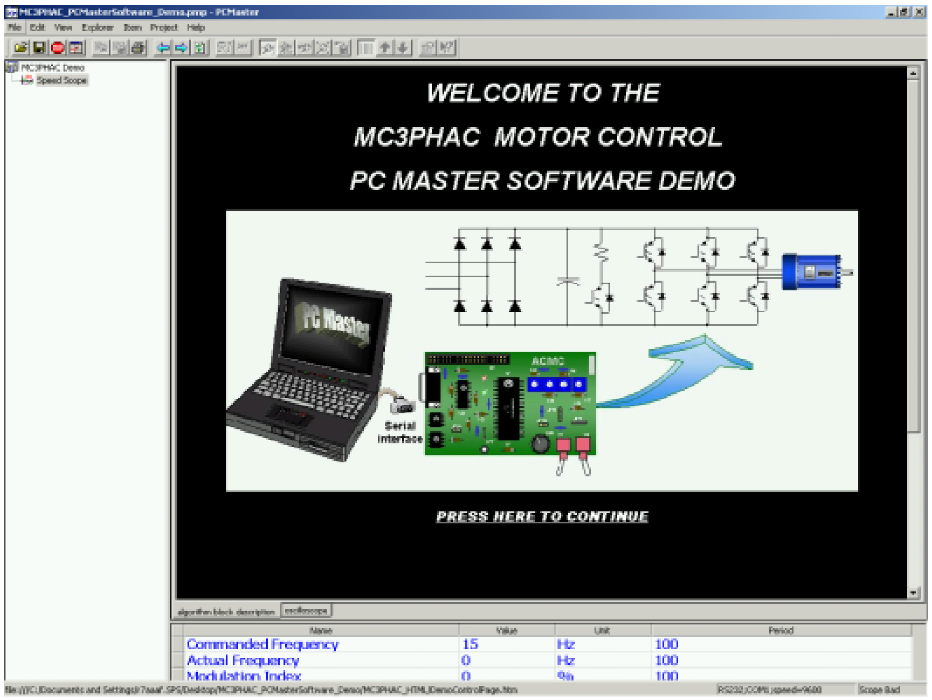
Figure 4. Connecting Motorola Microcontrollers to the Internet

PC Master Software Workspace

Using the PC master software application requires becoming familiar with the PC master software environment and workspace. The PC master software workspace is shown in **Figure 5**. As the figure illustrates, the workspace consists of three window frames. These include:

- A project-tree pane
- A watch-grid pane
- A detailed-view pane

The GUI operates primarily from the detailed-view pane and is displayed as an HTML page.


MC3PHAC_PCMasterSoftware_Demo.pmp - PCMaster
File Edit View Explorer Item Project Help
MC3PHAC Demo
Speed Scope
PC Master
Serial interface
ACMC
algorithm block description
oscilloscope
RS232;COM1;speed=9600
Scope Bad

WELCOME TO THE
MC3PHAC MOTOR CONTROL
PC MASTER SOFTWARE DEMO

PRESS HERE TO CONTINUE

Name	Value	Unit	Period
Commanded Frequency	15	Hz	100
Actual Frequency	0	Hz	100
Modulation Index	0	%	100

file:///C:/Documents and Settings/r7aaaf.SPS/Desktop/MC3PHAC_PCMasterSoftware_Demo/MC3PHAC_HTML/DemoControlPage.htm

Figure 5. PC Master Software Application Workspace

GUI Development Methodology

A methodology for development of a GUI for the MC3PHAC is provided next. This implementation of the methodology uses the PC master software application and several ActiveX components to minimize the coding requirements. The GUI is developed and packaged using HTML. The HTML GUI is created using Microsoft's FrontPage tools. A similar methodology can be implemented when developing the GUI application with other GUI tools.

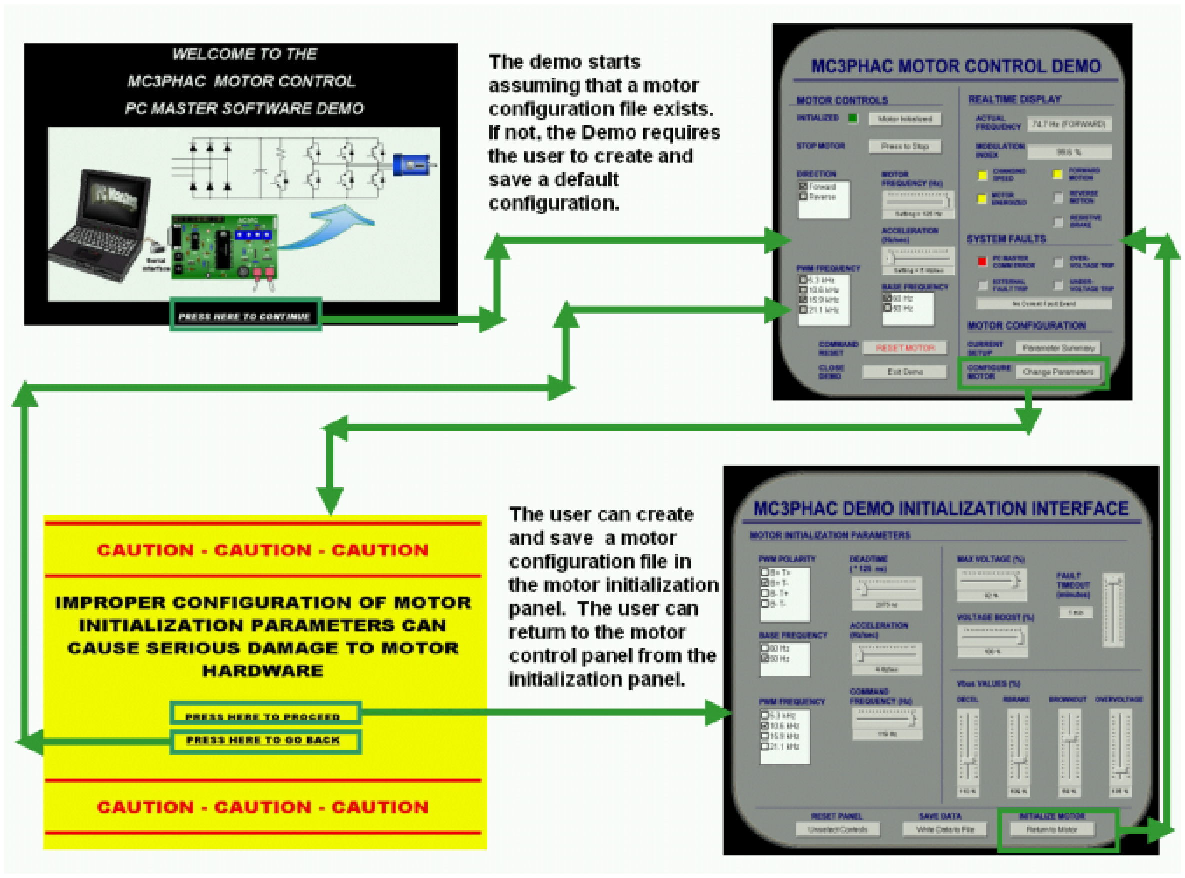
To better understand the process, in this implementation, the GUI development process is broken into several steps. These steps include:

- Conceptualizing the complete GUI interface
- Setting up the GUI capabilities of the PC master software application
- Connecting the components together

Each of these steps is discussed in detail in the next sections.

Conceptualizing the Complete GUI Interface

In this step of the GUI creation process, the overall layout and flow of the GUI demo must be determined. Figure 6 illustrates the level of detail that must be achieved during the conception phase.



The demo starts assuming that a motor configuration file exists. If not, the Demo requires the user to create and save a default configuration.

CAUTION - CAUTION - CAUTION

IMPROPER CONFIGURATION OF MOTOR INITIALIZATION PARAMETERS CAN CAUSE SERIOUS DAMAGE TO MOTOR HARDWARE

CAUTION - CAUTION - CAUTION

The user can create and save a motor configuration file in the motor initialization panel. The user can return to the motor control panel from the initialization panel.

Figure 6. Demo Conception and Organization

Figure 6 also illustrates two important deliverables of this step. The first is determination of the overall organization of the GUI application. If more than one GUI is used, the navigation and interaction between the GUIs must be defined. Blank GUI HTML pages can now be created using the tool of your choice. Although, at this time, some work linking the GUIs with hyperlinks can begin, the major GUI coding should occur after setting up the PC master software application and configuring a PC master software project file. A discussion of the PC master software configuration and of coding the GUI HTML pages is included in later sections.

The second deliverable is the definition of the functionality of each GUI. With this understanding, a developer can start focusing on individual GUIs and determine what variables will need to be defined. More important, a developer can start to document how the GUI will interact with a user. This means creation of a state diagram for each individual GUI. Many questions can be answered with a state diagram for a GUI. For instance, when the GUI is loaded, a state diagram can define what must be initialized or what controls are available to users. These state diagrams will be used in developing the GUI code content. Figure 7 illustrates a state diagram for the MC3PHAC motor control HTML page (see Figure 15 for a screen shot of this GUI).

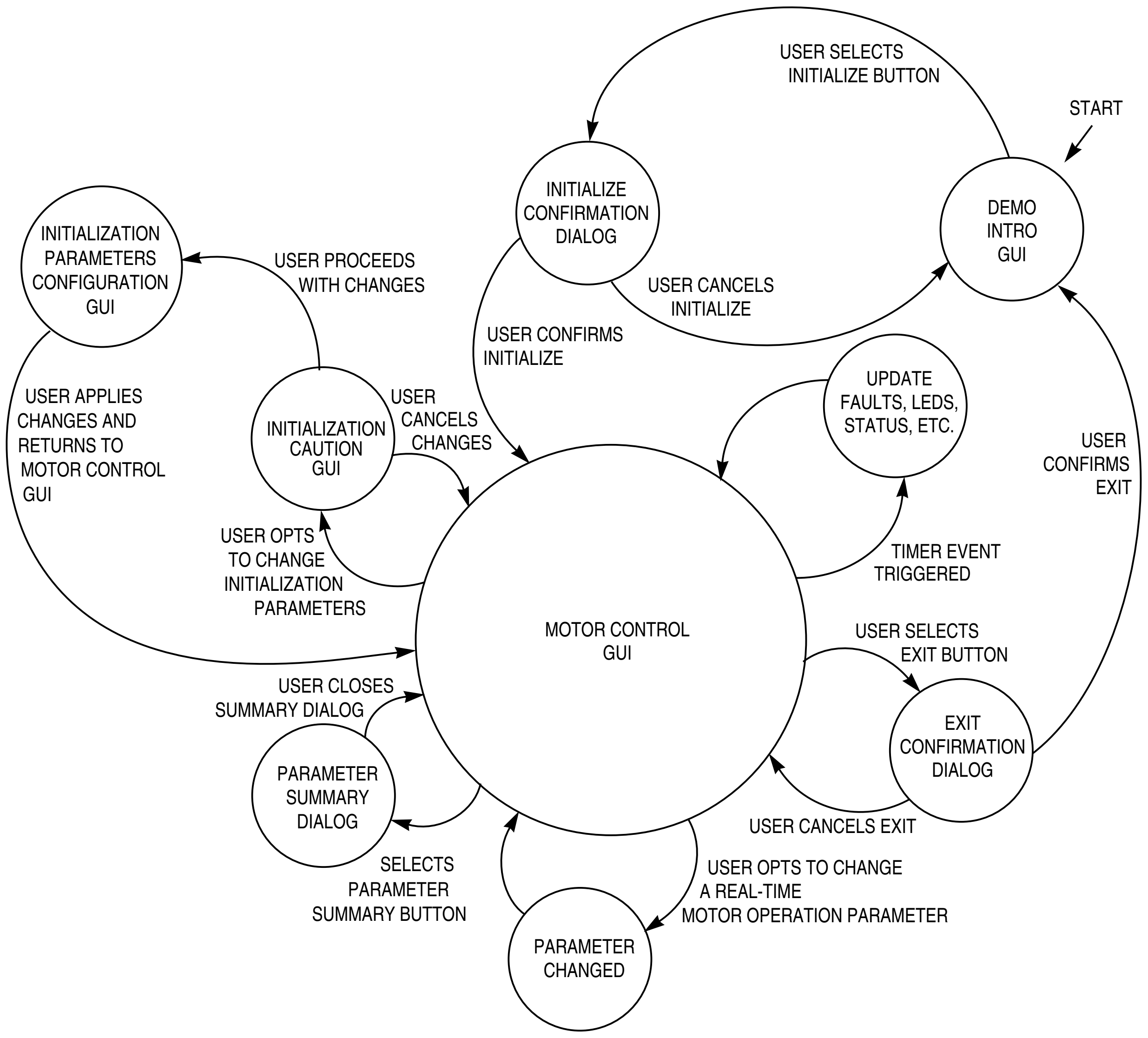

USER SELECTS INITIALIZE BUTTON
START
INITIALIZE CONFIRMATION DIALOG
DEMO INTRO GUI
INITIALIZATION PARAMETERS CONFIGURATION GUI
USER PROCEEDS WITH CHANGES
USER CANCELS INITIALIZE
USER CONFIRMS INITIALIZE
UPDATE FAULTS, LEDS, STATUS, ETC.
USER APPLIES CHANGES AND RETURNS TO MOTOR CONTROL GUI
INITIALIZATION CAUTION GUI
USER CANCELS CHANGES
USER CONFIRMS EXIT
TIMER EVENT TRIGGERED
USER OPTS TO CHANGE INITIALIZATION PARAMETERS
MOTOR CONTROL GUI
USER SELECTS EXIT BUTTON
USER CLOSES SUMMARY DIALOG
EXIT CONFIRMATION DIALOG
PARAMETER SUMMARY DIALOG
USER CANCELS EXIT
SELECTS PARAMETER SUMMARY BUTTON
USER OPTS TO CHANGE A REAL-TIME MOTOR OPERATION PARAMETER
PARAMETER CHANGED

Figure 7. State Diagram

Setting Up the GUI Capabilities of the PC Master Software Application

For this GUI implementation, as a precursor to GUI coding, the PC master software application HTML view, variable table, and communication link need to be set up.

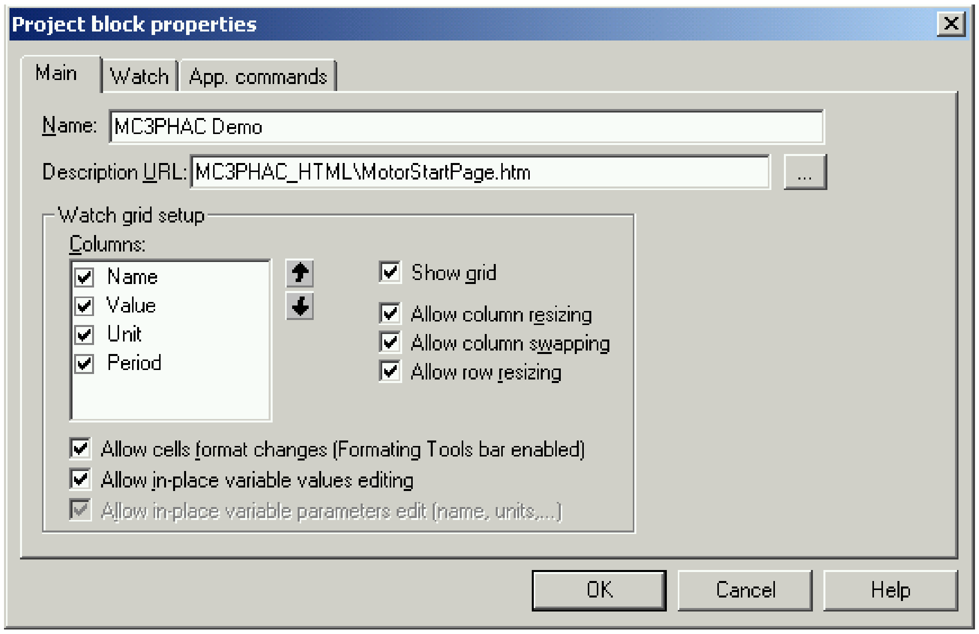
The configuration for this PC master software GUI should then be saved to a PC master software project file. Selecting **Save Project** from the **File** menu does this.

PC Master Software HTML GUI

With a PC master software implementation of a GUI for the MC3PHAC, the GUI is centered in the HTML view of the PC master software workspace. To configure the PC master software application to open a particular HTML page for a project file:

1. From the **Item** menu, select **Properties**.
2. When the project block properties dialog box opens as shown in **Figure 8**, press the **Main** tab and enter the name of the project.
3. Enter the path of the starting page of the GUI in the **Description URL:** text box.

The GUI start page should have been determined during the conception phase of the GUI development. Upon approving this action, an empty HTML page will open in the HTML pane of the PC master software application.



Name: MC3PHAC Demo

Description URL: MC3PHAC_HTML\MotorStartPage.htm

Watch grid setup

Columns:

- ☑ Name
- ☑ Value
- ☑ Unit
- ☑ Period

- ☑ Show grid
- ☑ Allow column resizing
- ☑ Allow column swapping
- ☑ Allow row resizing

- ☑ Allow cells format changes (Formating Tools bar enabled)
- ☑ Allow in-place variable values editing
- ☑ Allow in-place variable parameters edit (name, units,...)

Figure 8. Project Block Properties Dialog Box

PC Master Software User Interface Variable Setup

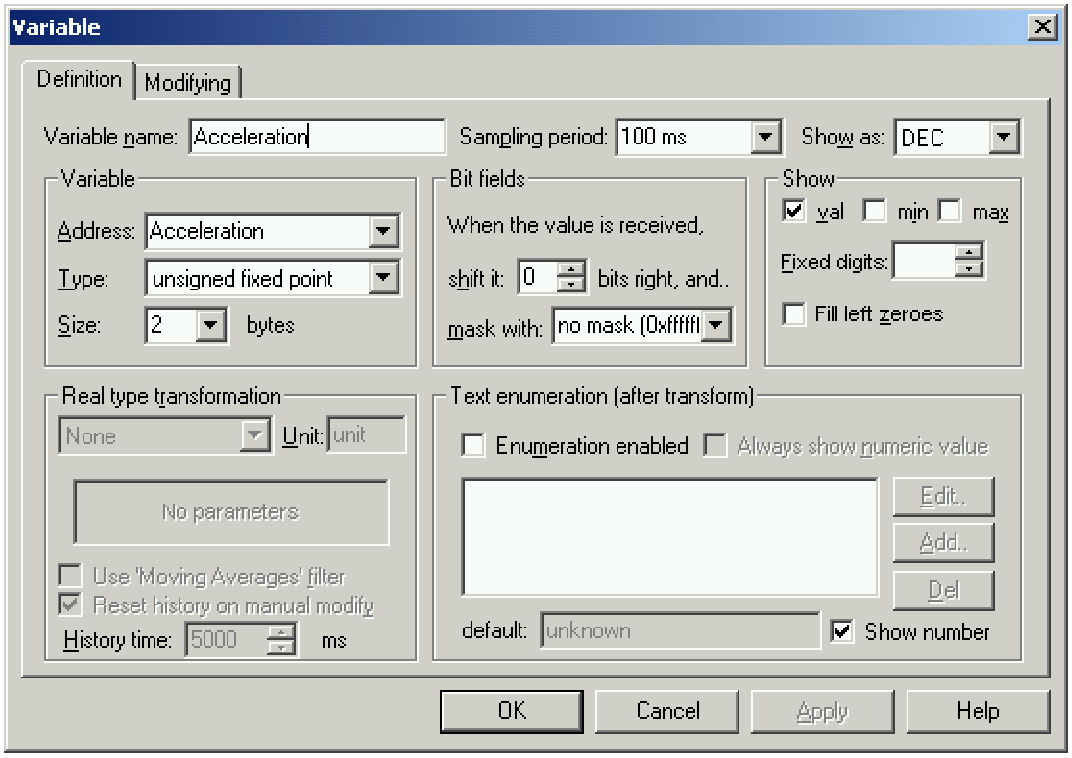
To prepare the PC master software application to communicate with an HTML page, the PC master software project file must be set up to recognize the variables that the HTML page will encompass. In this case, the variables that the PC master software project file must understand are the MC3PHAC user interface variables. All these variables need to be defined in the PC master software variable database that PC master software provides.

Variables can be written into the PC master software database in two ways. The first option is to enter variables one at a time. To read in variables one at a time:

From the **Project** menu, press **Variables**, then press **New**.

The second option is to read a map file into the PC master software variable database. This option can read multiple variables into the variable database in a single operation. For more information about reading variables into the variable database using a map file, refer to the *PC Master Software User Manual*.

The variable configuration dialog box is shown in Figure 9. Setting up the MC3PHAC user interface variables in PC master software requires some user input. The variable size and address need to be exactly as dictated by MC3PHAC documentation.


Variable

Definition | Modifying

Variable name: Acceleration
Sampling period: 100 ms
Show as: DEC

Variable
Address: Acceleration
Type: unsigned fixed point
Size: 2 bytes

Bit fields
When the value is received,
shift it: 0 bits right, and..
mask with: no mask (0xffffffff)

Show
[✓] val [ ] min [ ] max
Fixed digits:
[ ] Fill left zeroes

Real type transformation
None Unit: unit
No parameters
[ ] Use 'Moving Averages' filter
[✓] Reset history on manual modify
History time: 5000 ms

Text enumeration (after transform)
[ ] Enumeration enabled [ ] Always show numeric value
default: unknown
[✓] Show number

Figure 9. Variable Definition Dialog Box

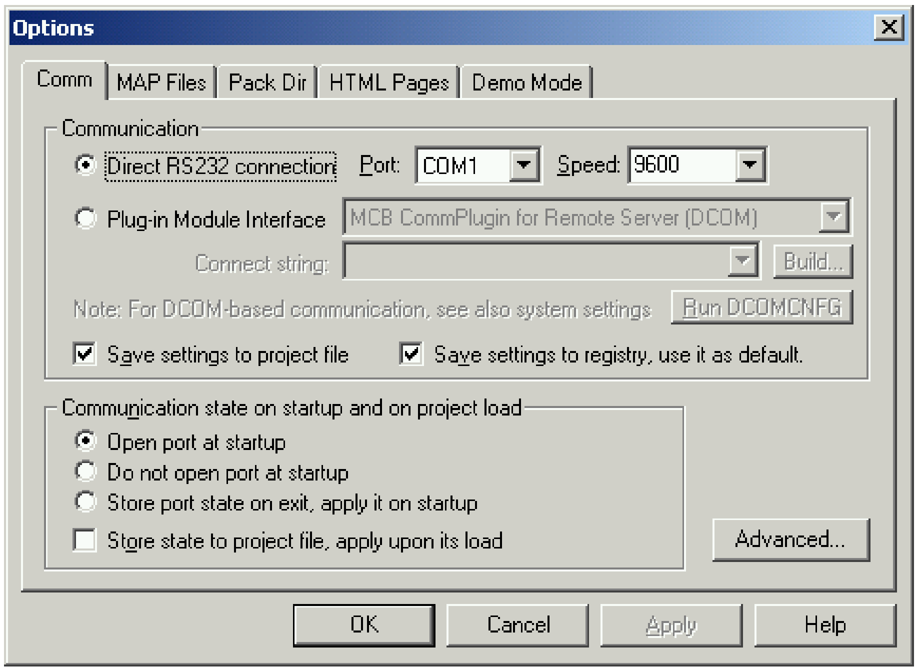
On the modify tab, the variable can be configured as read only. This dialog box also allows many other options to configure a variable. When changing the variable configuration, the variable's modified form will then be used in communication with the GUI. This can get quite messy since the configuration used for all variables will have to be remembered and implemented on the GUI side. To simplify the coding dramatically, stick with one variable configuration in the PC master software variable definition and do all data conversion in the GUI code.

PC Master Software Communication Link

Setting up the communication link for the GUI is straightforward with PC master software because the PC master software application is preprogrammed to use the serial port. To set up the communication port for PC master software, the project options dialog must be opened:

1. From the **Project** menu, select **Options**.
2. When the project options dialog box opens, select the MCB Comm tab and enter the communication port and port speed.

Figure 10 shows typical settings for the PC master software communication options dialog box. All that is needed is a physical connection to the MC3PHAC.

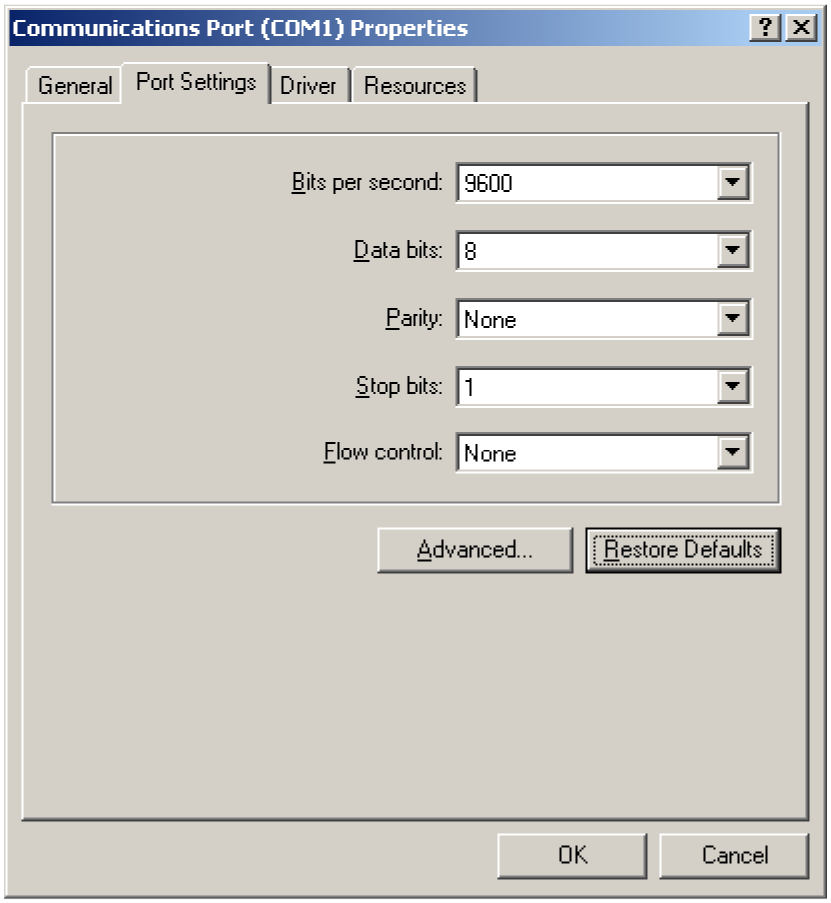


Communication
[● Direct RS232 connection] Port: COM1 Speed: 9600
○ Plug-in Module Interface MCB CommPlugin for Remote Server (DCOM)
Connect string:
Note: For DCOM-based communication, see also system settings
[✓] Save settings to project file [✓] Save settings to registry, use it as default.

Communication state on startup and on project load
[● Open port at startup]
○ Do not open port at startup
○ Store port state on exit, apply it on startup
☐ Store state to project file, apply upon its load

Figure 10. PC Master Software Communication Options Dialog Box

Recall, in this GUI implementation, a PC and the MC3PHAC reference design are being used. A serial cable is first required to connect the PC serial port to the MC3PHAC reference design serial port. With the devices connected, the PC can be set up. For the PC, the port setting must be configured so that there is no flow control on the port, as shown in Figure 11. Select the **Port Settings** tab on the **Communications Port Properties** dialog box to access this setting. The **Communications Port Properties** dialog box can be opened in Windows via the **Control Panel**. (Double click the systems icon and find the communications port device in the device manager to access this dialog box.)



Bits per second: 9600
Data bits: 8
Parity: None
Stop bits: 1
Flow control: None

Figure 11. Serial Port Setting Dialog Box

If PC master software is not used, the communication link can take the form of an ActiveX object, such as MSCOMM32.

Connecting the Components Together

This section will describe the process of coding up the GUI. At this point, state diagrams of the conceptual GUI pages have been drawn up. These state diagrams describe not only the user interface and operational functionality of the page, they also describe how the different GUIs of the complete application interact with one another. The next step is to turn these diagrams into a GUI for the MC3PHAC.

For this particular implementation, the first step is to identify the various components and tools available that assist in conversion of the state diagrams into code. Besides listing these tools, it is important to describe how each will be used. **Table 3** itemizes these tools and components.

Table 3. Tools and Components for GUI creating

Tools and Components	Functionality
PC master software ActiveX component	Access the PC master software communication link, commands, and protocol
Microsoft ActiveX components	Provide advanced user interface objects like sliders and lists
HTML form components	Provide simple user interface objects like buttons
Windows document object model	Provide advance Windows/HTML objects like timers
HTML tags	Formats and organize the HTML GUI page
Microsoft's Visual Basic Script	Tie together code components with user and operational functionality
Visual Basic Script file system object	Provide capability to open, save, and manipulate file on the PC hard drive
GIFs, JPEGs, BMPs, etc.	Provide graphics for GUI

The next step is to combine these elements into an HTML page and then tie them together with Visual Basic Script™ to provide the user and operational functionality of the GUI. An analogous process should be followed for implementations with different GUI development tools.

Tools and Components

Before discussing the Visual Basic Script code used to tie all the components together, a brief review of some of the components is provided in the this section.

PC master software ActiveX Object — The PC master software ActiveX object sets up both the PC master software commands and the PC master software protocol that will be used by the GUI. This ActiveX object is installed and registered on a PC that has the PC master software application installed.

If the PC master software ActiveX is not used, other ways of achieving these functions must be developed in-house.

To access this functionality in a particular GUI, all that is required is embedding the PC master software ActiveX object into a GUI. For the implementation of the ActiveX PC master software component in an HTML GUI, the ActiveX object description must be placed in the HTML page. The PC master software ActiveX object description is provided here:

```
<object name="PCMaster" width="25" height="14"
classid="clsid:48A185F1-FFDB-11D3-80E3-00C04F176153">
</object>
```

As discussed earlier, the ActiveX object exposes several member functions to the HTML GUI. In this example, the two member functions of interest are *WriteVariable* and *ReadVariable.* The syntax of these PC master software ActiveX member functions is provided here:

```
success = PCMaster.WriteVariable(bsVar,vValue,bsRetMsg)
success = PCMaster.ReadVariable(bsVar,vValue,tValue,bsRetMsg)
```

Both of the member functions return a boolean variable, *success*. The boolean variable indicates whether the function was completed successfully. Both member functions also contain the *bsRetMsg* variable. This variable provides additional details about the result of the *success* variable.

The use of these member functions is straightforward. To write a variable to the MC3PHAC, in the *WriteVariable* member function is used. The function takes two arguments for execution:

- *BsVar* is the name of a variable in the PC master software variable database, as described above. (This must match the format for the variable recorded in the PC master software variable database.)
- *vValue* is the value to be written (the format of the value written between PC master software and the Visual Basic Scripting).

After the member function is executed, the success of the operation is provided in the *success* and *bsRetMsg* variables.

To read a variable from the MC3PHAC, a similar procedure is followed using the *ReadVariable* member function. A complete description of these member functions is provided in the *PC Master User Guide*.

ActiveX, HTML, and Other Components — These components provide the majority of content for the GUI in the example below. The components include visible GUI items like sliders, checkboxes, buttons, LEDs, lists, status bars, and so on. These components also include invisible functions like timers and object events. Although the example below is limited to the components mentioned, other components are also available from third parties or can be developed in-house.

As with the PC master software ActiveX object, these components need to be embedded into the HTML GUI. In this case, many of these components can be placed in the GUI HTML using an application like Microsoft's FrontPage. For example, to embed an ActiveX into the HTML GUI:

> From the **Insert** menu in FrontPage, select **Advanced** and then **ActiveX** control (see **Figure 12**). An ActiveX object can then be selected from the ActiveX list displayed.

Inserting HTML form components follows a similar process.

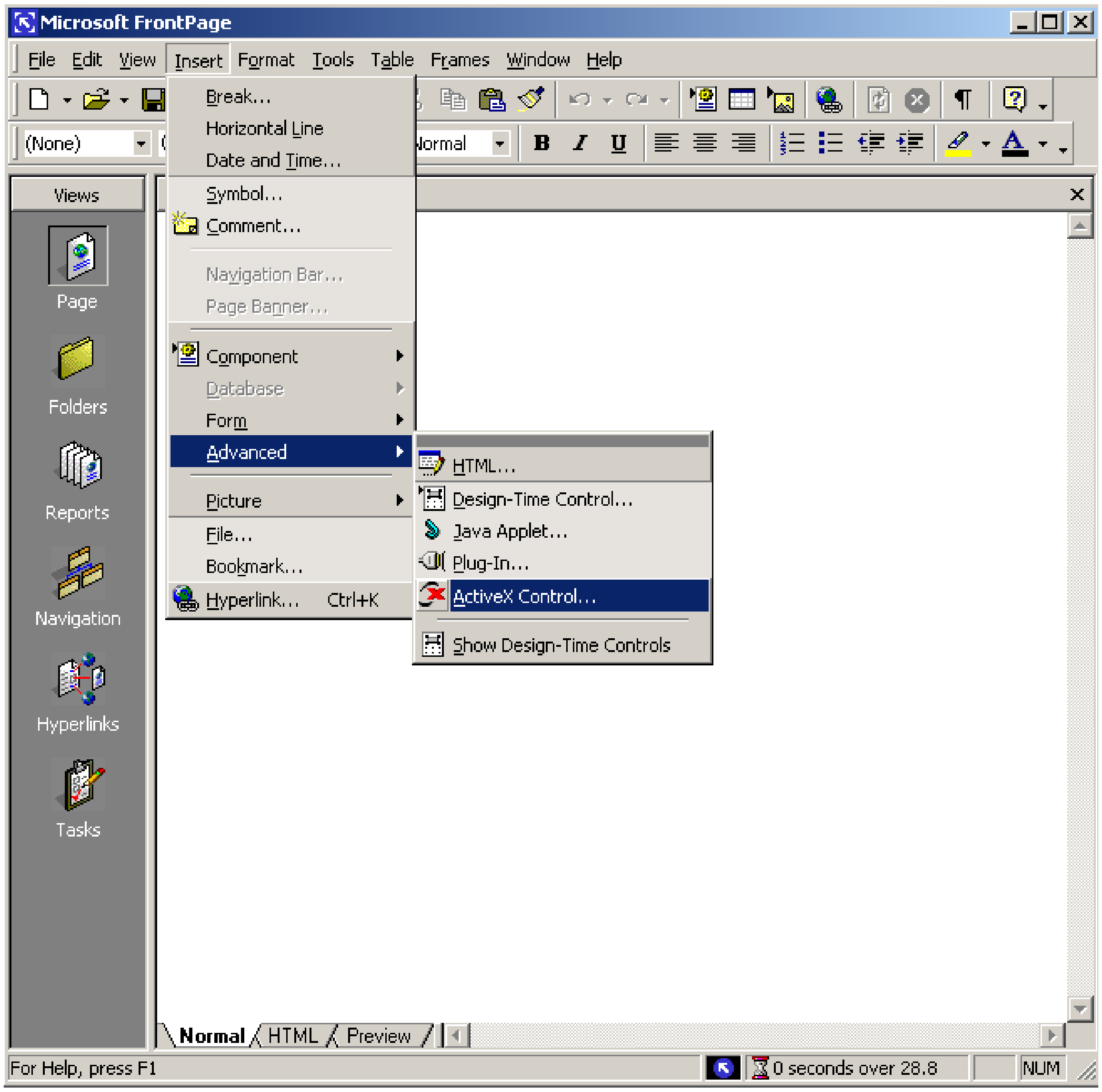


Figure 12. Inserting GUI Components with FrontPage

Each of these components is typically packaged with a collection of properties, methods, and events that may be invoked in the development of the GUI. An example of this is evident with a slider ActiveX object. Several slider properties can be set, including the slider range, tick gradation, orientation, and size. Methods are also available with a slider. A method for the slider can take the form of a function that can be used to set the slider to a particular value. As for events, the slider can call a function based on user input. For instance, it can call a specified function if its value is changed.

Exploiting the features of ActiveX and the other components is required to develop a user friendly and interactive GUI. More information on these components can be found in Microsoft's MSDL Library or in the components' documentation.

Table 4 details the type and quantity of components used in the motor control HTML GUI page discussed below.

Table 4. Components Used in Example Motor Control HTML GUI

Components	Type	Quantity
List view	ActiveX	3
Slider	ActiveX	2
Status bar	ActiveX	5
Buttons	Form	6
LED	Form	10
Timer	Windows Object	1
Control graphics	BMP	1

The components were oriented and organized over a graphic to resemble a motor control panel. The functionality of these components is brought out using Visual Basic Scripting. Visual Basic Script is a programming language developed by Microsoft and typically used in conjunction with HTML. References for the language are widely available. An online Visual Basic Script language reference is available on the Microsoft Web site: http://Microsoft.com The next section reviews this process in more detail.

Using Visual Basic Script to Develop GUI Functionality

Once the HTML GUI has been organized and developed with the GUI objects, use a scripting language to build in the user and operational functionality required by the GUI.

Understanding Visual Basic Scripting by Example — An example piece of code provided below illustrates using Visual Basic Scripting to tie together the PC master software ActiveX, Windows ActiveX, and HTML form components. The code is a subroutine for a slider that is called by a user event. The subroutine connects an event to a desired action. The event that triggers this subroutine is a user attempting to change the commanded frequency of the MC3PHAC. Upon the slider value change, the subroutine is executed. The remainder of this section describes the code.

```
      '=================================================================
      '  NAME: MotorFreqSlider
      '
      '  DESCRIPTION:    Activates upon indicated event.  Accepts user input and
      '                  transmits via PC master software. Changes Motor Frequency.
      '=================================================================
1      Sub MotorFreqSlider_Change()
2
3            Dim bsVar,vValue,bsRetMsg, success
4
5            If MotorSetup Then
6                    'Send user input to PC master software
7                    bsVar = MotorFreq
8                    vValue =  MotorFreqSlider.Value
9                    success = PCMaster.WriteVariable(bsVar,vValue,bsRetMsg)
10                   If success then
11                           PCMasterError = FALSE
12                           CurrentMotorFreq = MotorFreqSlider.Value
13                           SetSpeed.Panels(1).Text = "Setting = "_
14                                         & CurrentMotorFreq & " Hz"
15                   Else
16                           PCMasterError = TRUE
17                           msgbox "PC master software Successful:  "& success_
18                                  & ",  Address: " & bsVar_
19                                  & ",  Value: " &MotorFreqSlider.Value&Chr(10)_
20                                  &"Error Message: "&bsRetMsg,16
21                           MotorFreqSlider.Value = CurrentMotorFreq
22                   End If
23           End If
24     End Sub
```

Line 1 is the beginning of the subroutine. It executes upon a change in the slider value, which is the user input event. The next step is processing the user request to change the commanded frequency of the MC3PHAC. Before allowing the change, a global state variable, *MotorSetup,* of the motor is checked (line 5). The command is processed only if the motor is operating correctly.

To change the commanded frequency of the MC3PHAC, the PC master software ActiveX object member function, *WriteVariable,* is used. Lines 7 thru 9 illustrate these operations. The arguments passed to *WriteVariable* are the PC master software variable identifier, *bsVar*, and the value to be written, *vValue*. Line 10 then checks to see whether the *WriteVariable* operation is successful.

If the operation is successful:

- A global state variable for PC master software communication errors is set to false (line 11)
- A global value of the current command frequency is set to the new commanded frequency (line 12)
- The motor frequency display, a status bar ActiveX component, is updated to reflect the new commanded frequency (line 13)

If the operation is not successful:

- A global state variable for PC master software communication errors is set to true (line 16)
- A PC master software error message box is displayed (lines 17–20)
- The command frequency is set back to its previous setting (line 21).

Although this is a simple example, it shows a basic coding process and style used to tie the various objects together with Visual Basic Scripting. By applying this process to all of the GUI components, and using a state diagram of the desired functionality as a guide, the GUI can be completely coded. The next section reviews in detail an example GUI interface that was developed for the MC3PHAC. The example GUI was developed using the process described above.

MC3PHAC PC Master Demo Example

The MC3PHAC PC master software demo provides an example GUI that interfaces to the MC3PHAC with PC master software protocol and its serial communication link. The demo shows the MC3PHAC real-time control and operation through PC master software. The demo also illustrates the versatility of Motorola's PC master software motor control application.

The demo was tested and developed with an IBM-compatible PC running PC master software Version 1 Build #83 on Windows 2000. The demo is best viewed on displays with a screen size set to at least 1024 by 768 pixels. The demo has also been tested with the PC master software, version 1.2.0.9.

The demo should also be compatible with Windows 98, 98 SE, and NT, though this has not been tested. It is recommended, however, to use the latest version of Motorola's PC master software Motor Control Application. Moreover, because the demo uses Microsoft ActiveX components, the demo recommends the presence of Internet Explorer Version 5 or later. Specifically for the ActiveX components to operate, the following file should be installed by Internet Explorer: MSComCtl.ocx.

Demo File Organization

The complete MC3PHAC PC master software demo file package is provided in a ZIP file. In this release, to install the demo, the ZIP file must be unzipped to the hard drive. The diagram below illustrates the directory tree of the unzipped MC3PHAC PC master software demo package. All demo files are contained in a directory named MC3PHAC_PCMaster_Demo.

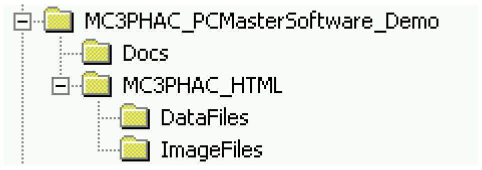


Figure 13. MC3PHAC PC Master Software Demo Directory Tree

The demo is launched via the PC master software project file named MC3PHAC_PCMaster_Demo.pmp. This file can be found in the MC3PHAC_PCMaster_Demo directory. Left-mouse double-clicking the MC3PHAC_PCMaster_Demo.pmp file can open the demo. Alternatively, opening PC master software motor control application and selecting **Open Project** from the **File** menu can start the demo.

Besides the PC master software project file, the demo requires several supporting files to operate successfully. These files include a map file that contains a description of the MC3PHAC user interface variables and a Web folder that consists of a collection of HTML and supporting files describing the MC3PHAC PC master software demo GUI. These files are referenced by the PC master software project file and are located in the MC3PHAC_HTML directory.

MC3PHAC PC Master Software Demo Operation

PC Master Software Startup

Opening the MC3PHAC demo PC master software project file brings up the PC master software environment and workspace. As the PC master software environment loads, it checks the serial communication port for the MC3PHAC reference board. If the MC3PHAC reference board is found and communication is successful, the PC master software environment is ready to communicate PC master software and MC3PHAC commands to the MC3PHAC.

This communication is paused if the initial communication is not successful. The communication problem must be resolved before the demo can operate successfully. Possible communication problems can be the result of a mismatched communication port (default is COM 1), an improperly attached serial or power cable, or a problem with the hardware settings on the MC3PHAC reference board. Whether or not communication is successful, the PC master software workspace will open.

Other relevant options on the PC master software workspace can be accessed through the PC master software file menu. These are listed below.

To specify the HTML file (description URL):

1. Go to the **Item** menu.
2. Choose **Properties**.
3. Click on the **Main** tab.

To load a user interface variable:

1. Go to the **Project** menu.
2. Select **Reload Map File**.

To generate/edit variables:

1. Go to the **Project** menu.
2. Select **Variables**.
3. Click the **Generate** button.

To map the file specification (default MAP/ELF file)

1. Go to the **Project** menu.
2. Select **Options**.
3. Click the **Map Files** tab.

To specify the COMM port for PC master software:

1. Go to the **Project** menu.
2. Click **Options**.
3. Go to the **MCB Comm** tab.

To open the demo or other PC master software project files:

1. Go to the **File** menu.
2. Click **Open Project**.

To pause/reactivate the PC master software serial communication link

1. Go to the **File** menu.
2. Click the **Stop Communication** button.

For this demo, the PC master software project file is pre-configured and ready to use without modification.

Demo HTML Components

Several HTML pages have been developed for the demo. The HTML pages provide not only a tool to navigate the demo; more importantly, they provide an interface to the MC3PHAC. PC master software loads each HTML page in its HTML view frame while the demo is running according to the user's interaction.

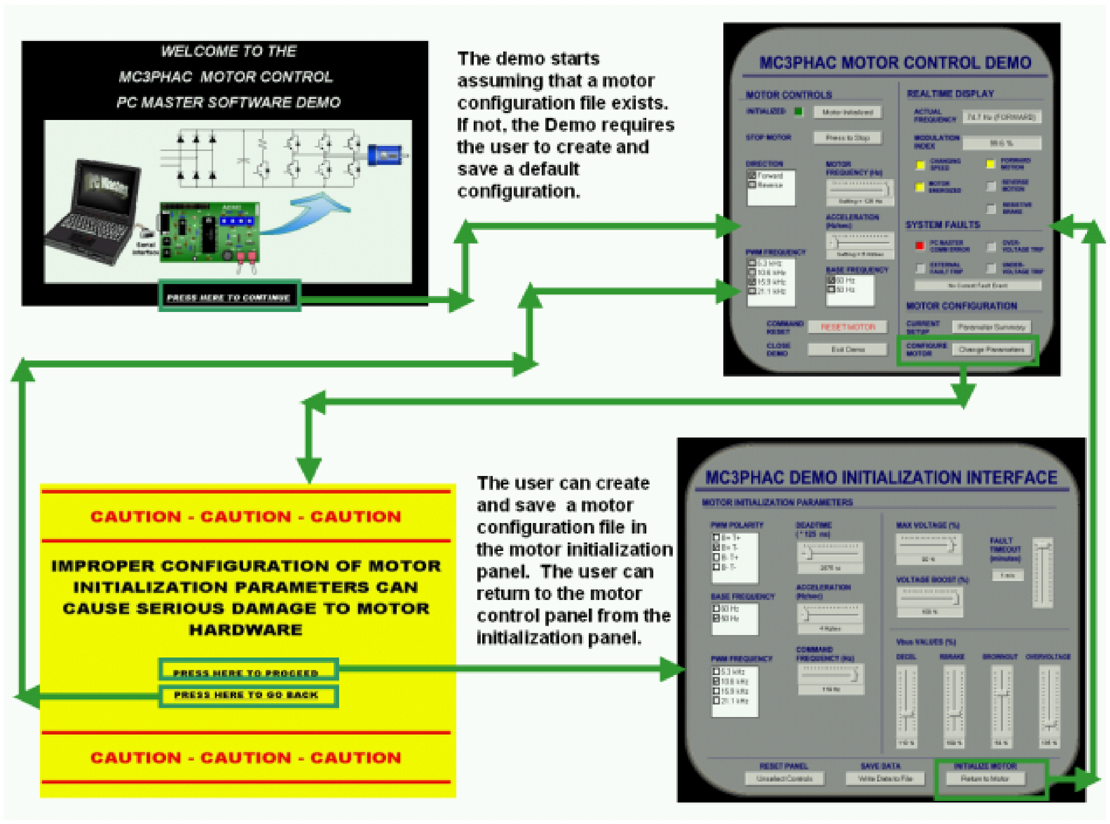
The MC3PHAC PC master software demo starts by loading the demo's Demo Introduction HTML page. Hyperlinks link this page to remaining demo pages. **Figure 14** illustrates the HTML layout and flow for the demo. **Figure 14** includes a screenshot of each of the four demo HTML pages:

- Demo Introduction
- Motor Control
- Initialization Caution
- Motor Initialization

On the left side of **Figure 14**, the Demo Introduction (top) and Initialization Caution (bottom) HTML pages are shown. These HTML pages are mainly used to display messages and/or for navigation.

The other two HTML pages for the demo are more functional. The Motor Control HTML page (**Figure 14** top right side), in particular, is designed to have a real-time interface with the MC3PHAC via the PC master software protocol. The Motor Initialization HTML page (**Figure 14** bottom right side) is used primarily to save a default motor configuration.

Internally, the latter two HTML pages are designed with HTML components, Visual Basic Script functions, and various ActiveX objects as described in the **Connecting the Components Together** section. Since the demo uses ActiveX objects, occasionally, the demo will display a warning message about ActiveX activity. This warning is normal, and, to avoid a demo runtime error, it is important that the user allow the ActiveX activity. More details about these HTML pages are provided in the subsequent sections.


WELCOME TO THE
MC3PHAC MOTOR CONTROL
PC MASTER SOFTWARE DEMO
PRESS HERE TO CONTINUE
MC3PHAC MOTOR CONTROL DEMO
MOTOR CONTROLS
REALTIME DISPLAY
SYSTEM FAULTS
MOTOR CONFIGURATION
CAUTION - CAUTION - CAUTION
IMPROPER CONFIGURATION OF MOTOR INITIALIZATION PARAMETERS CAN CAUSE SERIOUS DAMAGE TO MOTOR HARDWARE
PRESS HERE TO PROCEED
PRESS HERE TO GO BACK
CAUTION - CAUTION - CAUTION
MC3PHAC DEMO INITIALIZATION INTERFACE
MOTOR INITIALIZATION PARAMETERS

The demo starts assuming that a motor configuration file exists. If not, the Demo requires the user to create and save a default configuration.

The user can create and save a motor configuration file in the motor initialization panel. The user can return to the motor control panel from the initialization panel.

Figure 14. MC3PHAC PC Master Software Demo HTML Layout

Motor Control HTML Page

This Motor Control HTML page is the primary interface to the MC3PHAC for the demo. It is the only demo HTML page that has the PC master software ActiveX interface embedded in it. The PC master software ActiveX interface enables the Motor Control HTML page to communicate with the MC3PHAC using the PC master software protocol.

Upon loading, the demo determines the current state of the motor. If the motor is on, the user has the option to not initialize the motor and instead command the demo to read the current state of the motor. Reading the state of the motor does not interrupt the current motor operation. Once the current state of the motor is read and processed successfully, the Motor Control HTML page is put in an initialized state. Alternatively, the user can command that the motor reset and that the motor initialize using a default motor configuration file.

If the motor is off, the motor will automatically attempt to initialize by opening and processing the default motor configuration file. The demo then initializes the motor by writing data to the MC3PHAC and, when successful, puts the Motor Control HTML page in an initialized state. The default motor configuration file is created by the user using the Motor Initialization HTML page. Details regarding the Motor Initialization HTML page are provided in the next section.

Once the Motor Control HTML page is in an initialized state, the user is allowed to change, in real-time, motor parameters like the frequency, acceleration, and direction using the Motor Control HTML page controls. In the Motor Control HTML page, most of the controls are linked to the MC3PHAC using the PC master software ActiveX interface, and, because of this link, a change in the controls results in a change to the current motor state.

Several user interface controls are provided on this page to facilitate user commands. The controls include buttons, lists, and sliders. The basic operation of these controls is point and click. The **Operation of Demo ActiveX Controls** section describes the operation of these controls in more detail.

Besides the user interface controls on the Motor Control HTML page, the page also includes a number of motor status displays like LEDs and text boxes. The LEDs are used primarily to display and communicate the current state of the motor including faults, motion, and direction. The text boxes are used to display specific messages.

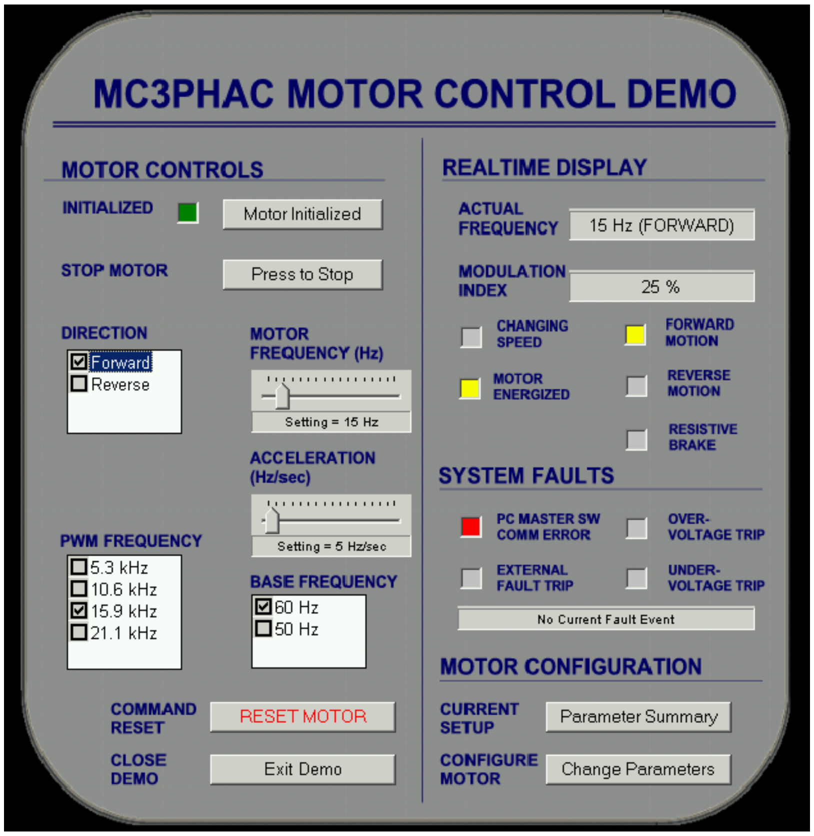


Figure 15. MC3PHAC PC Master Software Demo Motor Control HTML Page

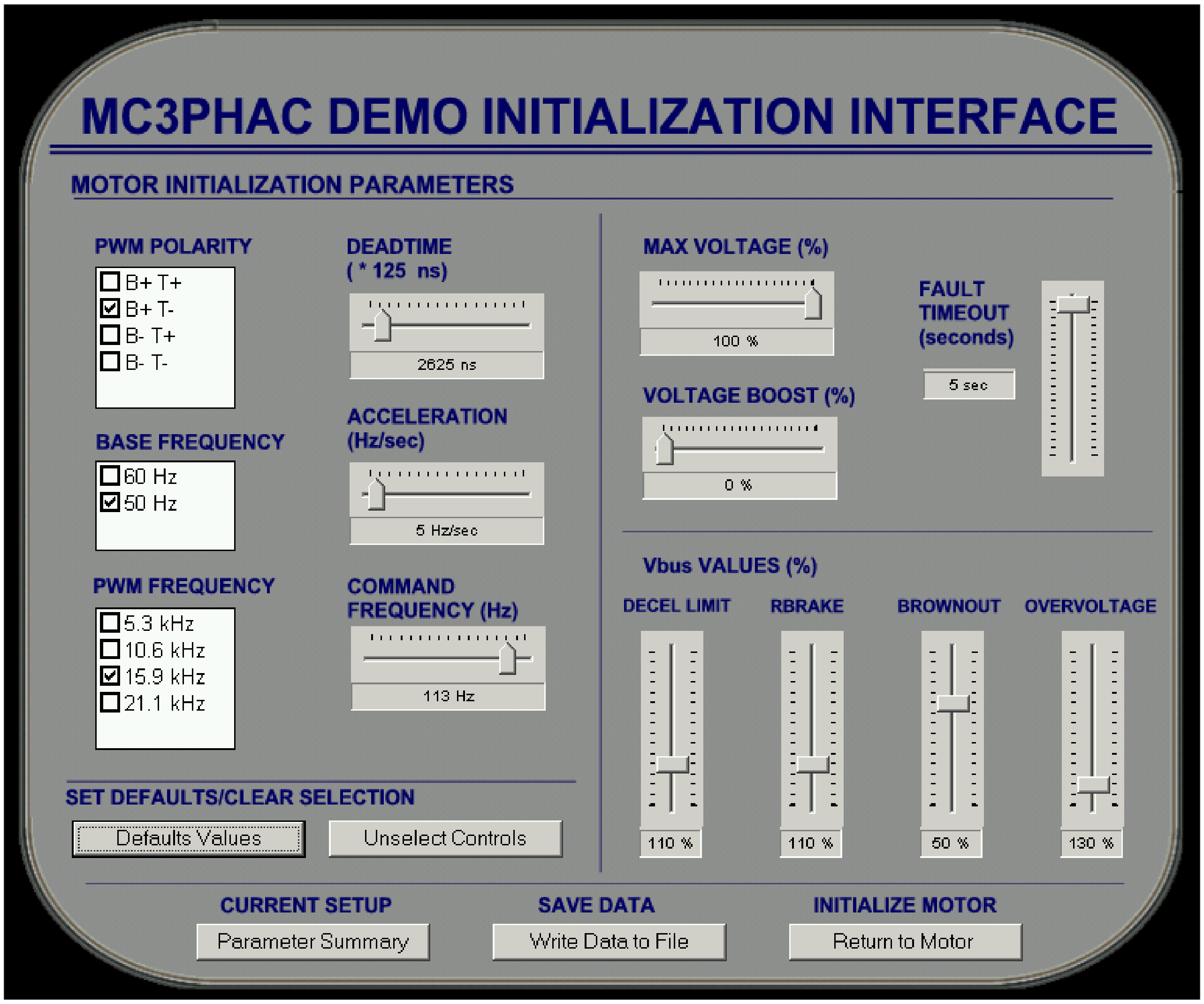
Motor Initialization HTML Page

The Motor Initialization HTML page provides a means to save configuration data to a motor configuration file. The parameters include items like dead time, PWM polarity, PWM frequency, base frequency, acceleration, motor frequency and various V_{Bus} values. Each parameter must be set before attempting to write the configuration to a file since only a complete parameter list can be saved.

While saving a default motor configuration file may sound like a trivial task, it is a very serious undertaking and must be approached with caution. The user must customize the initialization parameters based on their particular motor control application and hardware. Incompatible motor initialization settings can result in undesirable consequences. With this in mind, it is recommended that a qualified user set up the default motor configuration file.

The Motor Initialization HTML page has user interface controls similar to those on the Motor Control HTML page. These controls operate in the same manner as previously described except that the controls are not linked to the MC3PHAC.

Because no default motor configuration file is provided in the demo file package, the demo requires that default motor initialization parameters be set up during the first demo session. The configuration data is then called upon every time the Motor Control HTML page is loaded. It is used by the Motor Control HTML page to initialize the motor via PC master software. If the motor configuration file must be subsequently changed or updated, the user can return to the Motor Initialization HTML page to make modifications.


MC3PHAC DEMO INITIALIZATION INTERFACE

MOTOR INITIALIZATION PARAMETERS

PWM POLARITY
- ☐ B+ T+
- ☑ B+ T-
- ☐ B- T+
- ☐ B- T-

DEADTIME (* 125 ns): 2625 ns

MAX VOLTAGE (%): 100 %

FAULT TIMEOUT (seconds): 5 sec

BASE FREQUENCY
- ☐ 60 Hz
- ☑ 50 Hz

ACCELERATION (Hz/sec): 5 Hz/sec

VOLTAGE BOOST (%): 0 %

PWM FREQUENCY
- ☐ 5.3 kHz
- ☐ 10.6 kHz
- ☑ 15.9 kHz
- ☐ 21.1 kHz

COMMAND FREQUENCY (Hz): 113 Hz

Vbus VALUES (%)

DECEL LIMIT	RBRAKE	BROWNOUT	OVERVOLTAGE
110 %	110 %	50 %	130 %

Figure 16. MC3PHAC PC Master Software Demo Motor Initialization HTML Page

Operation of Demo ActiveX Controls

Slider Control User Changes

The user can change the slider control value after the motor is initialized. When a slider is changed, the demo picks up the user input and sends the data to the MC3PHAC via the PC master software serial communication interface. The communication with PC master software only occurs on the Motor Control HTML page.

To operate the slider, the user must click and hold the left mouse button, and drag the mouse so that the slider control moves to the desired value. When the desired value is obtained, the user releases the mouse button.

Alternatively, the user can left mouse click the control and then use the cursor keys to change the slider values. Using the cursor keys is preferred when a small change in the slider is desired. Each click on the cursor keys in a particular direction results in a one-unit change in the slider value in that direction.

Checkbox Click

The user can change the checkbox control after the motor is initialized. To operate the checkbox, the user must click the left mouse button inside the desired checkbox. The checkboxes are mutually exclusive. Each list control can have only one item checked at any given time. When a checkbox selection is changed, the demo picks up the user input and sends the data to the MC3PHAC via the PC master software serial communication interface.

PC Master Software Scope and Watch-Grid Pane

Besides the demo HTML interface, the demo PC master software MC3PHAC project file is configured to use other PC master software analysis tools. Specifically, the demo is set up to demonstrate two of PC master software’s built-in features.

The first of these features is the watch-grid pane. The watch-grid pane has been populated to display several demo parameters. This is an alternate view for the parameters, and, while the parameter values can be changed in the watch-grid pane, it is recommended that parameter values be changed using the demo HTML components that have been provided.

The more graphical of these built-in features is the scope. In the tree project pane of the PC master software workspace, the user must select **Speed Scope** to access the scope. The scope is then viewed by selecting the scope tab in the detailed-view pane. For the demo, the scope has been configured to provide a graphical display of frequency and other parameters plotted against time. **Figure 17** illustrates a typical scope display for the demo.

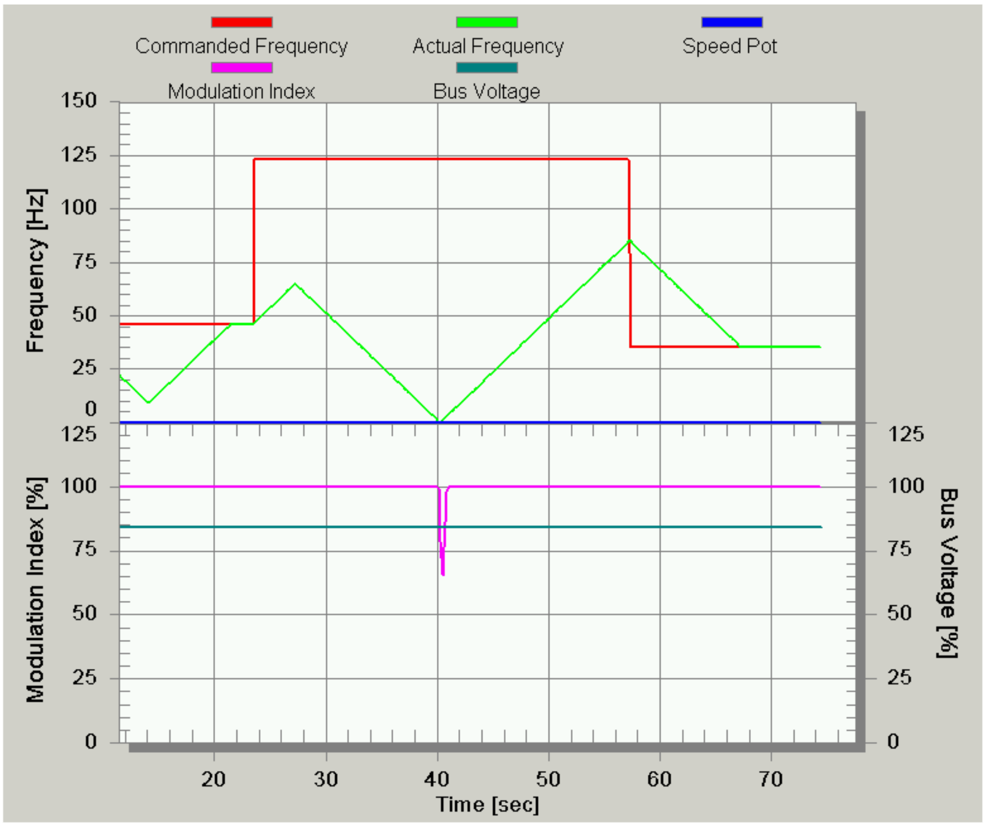


Figure 17. PC Master Software Speed Scope for the MC3PHAC Demo

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com