Chapter 11:
Process Control Studio Mobile Edition

11.1 Introduction

11.1.1 What is PDA? [1]

Personal digital assistants (PDAs) are handheld computers that were originally designed as personal organizers, but became much more versatile over the years. For more information refer to Appendix.

11.1.2 .NET Compact Framework (.NET CF)

The Microsoft **.NET Compact Framework** (.NET CF) is a version of the .NET Framework that is designed to run on mobile devices such as PDAs, mobile phones and set-top boxes. .NET CF features are listed below [2]

- .NET Compact Framework Capabilities
- Supported Runtime Elements
- The .NET Compact Framework Provides CPU Independence

11.1.3. Operating System Used in This Project

Team of the project came to an agreement to deal with **Windows Mobile for PDAs** using one of the most recent Microsoft's Pocket PC Operating Systems which is **Windows Embedded CE 5 (WinCE 5)** shown in Fig .2**.**

Nowadays Microsoft has launched Windows Embedded CE 6 (WinCE 6), but still having no bread popularity. [3]

[1] For more information refer to "What is PDA?" in the Appendix.

[2]For more information refer to "**.**NET Compact Framework (.NET CF)" in the Appendix

[3]For more information refer to "Windows CE Overview" in the Appendix.

Wednesday, July 04, 2007

Tap here to set owner information

No unread messages

No tasks

No upcoming appointments

Tap here to sign in to Pocket MSN!

Figure 11.1 PDA running WinCE 5 Platform

11.2 Program Description

From the previous introduction about the .NET Compact Framework (.NET CF) world, we move then to discussion of our program that is used in process control that's why we named **Process Control Studio Mobile Edition**.

Process Control Studio Mobile Edition is a program that runs on PDA under WinCE 5 operating system which is used for process view and control; from this goal of our program we merged the program into two tasks:

A. Loading the displayed form of the program by reading an XML file that saves the forms contents and attributes in it, this XML file is the same as the PC (Personal Computer) XML file used to load controls on PC display form. This was done to achieve data consistency between PC and PDA.

B. Dealing with Network Sockets on PDA to receive and send data required for controlling parameters under the constrain of the same protocol implemented on the PC , this is another way of making real world problems come to an agreement to the Handheld PCs world.

A.1 Reading an XML File

The **Extensible Markup Language** (**XML**) is a general-purpose markup language .Markup language as it doesn't do anything except describing data; it is classified as an extensible language because it allows its users to define their own tags. Its primary purpose is to facilitate the sharing of data across different information systems.

A.2. Scaling

This is what is done in our project, the same XML file that is read on the PC program, is also read on the PDA program, but with scaling the controls and the display form to be managed by the size of the PDA display screen.

A.3. Program Sequence

1. When user opens the program (named Process Control Studio Mobile Edition), he can choose to click File menu and select from it Open choice, shown in Fig.11.2.

2. By clicking Open, an Open File Dialog Box will be opened presenting all PCS (Process Control Studio) extension files that now present on PDA specified folder named PCS Files.

3. User at this moment can select which file would he like to open to view its display on the PDA screen, shown in Fig.3.

4. After selecting the file two tabs are created on the form, the first one is for the display form and the other is for viewing a graph representing a process graph, shown in Fig.4.

5. When the form is displayed, the user can click on any control on the form to display its panel to set and change parameters of that control, shown in Fig.5.

6. Program termination occurs when user clicks Close at any time of the program sequence.

A flowchart is shown in Fig.11.6 to illustrate the program sequence

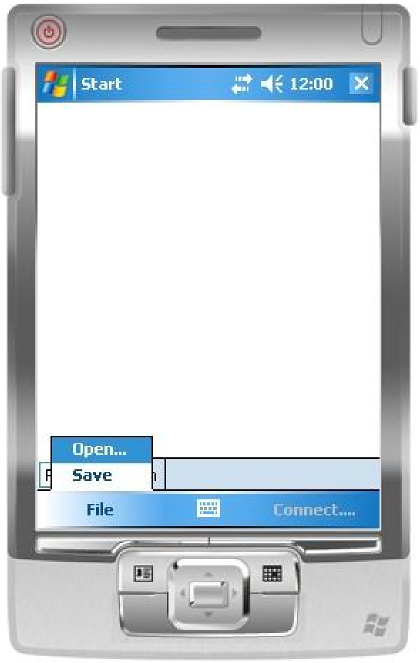
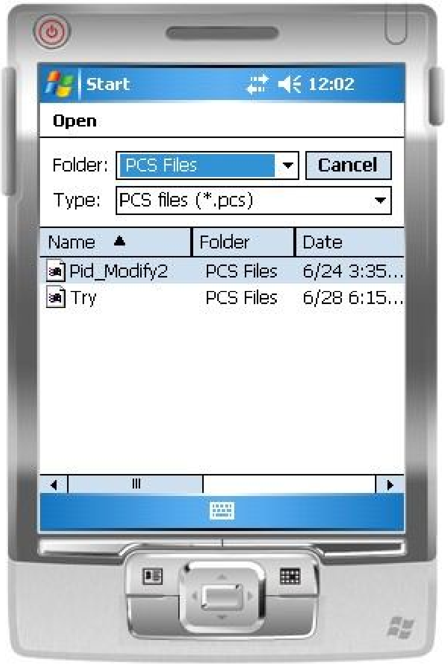


Figure 11.2 User clicks Open from File menu



Open

Folder: PCS Files

Type: PCS files (*.pcs)

Name ▲	Folder	Date
Pid_Modify2	PCS Files	6/24 3:35...
Try	PCS Files	6/28 6:15...

Figure 11.3 User selects a file to open

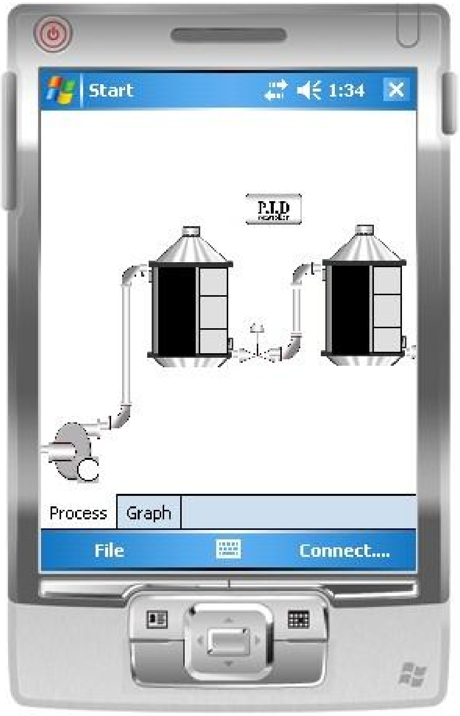
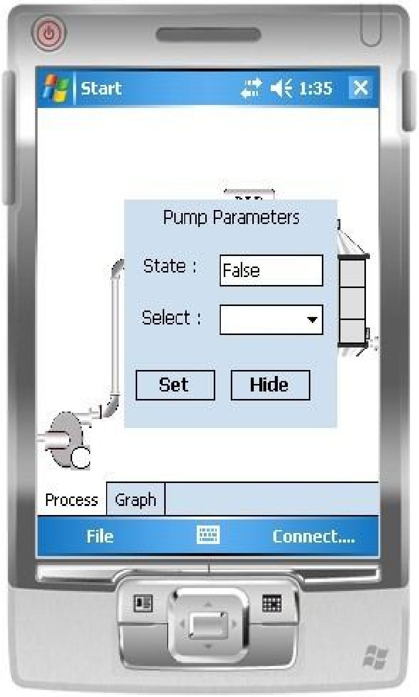


Figure 11.4 Form display



Pump Parameters

State : False

Select :

Figure 11.5 Control parameters

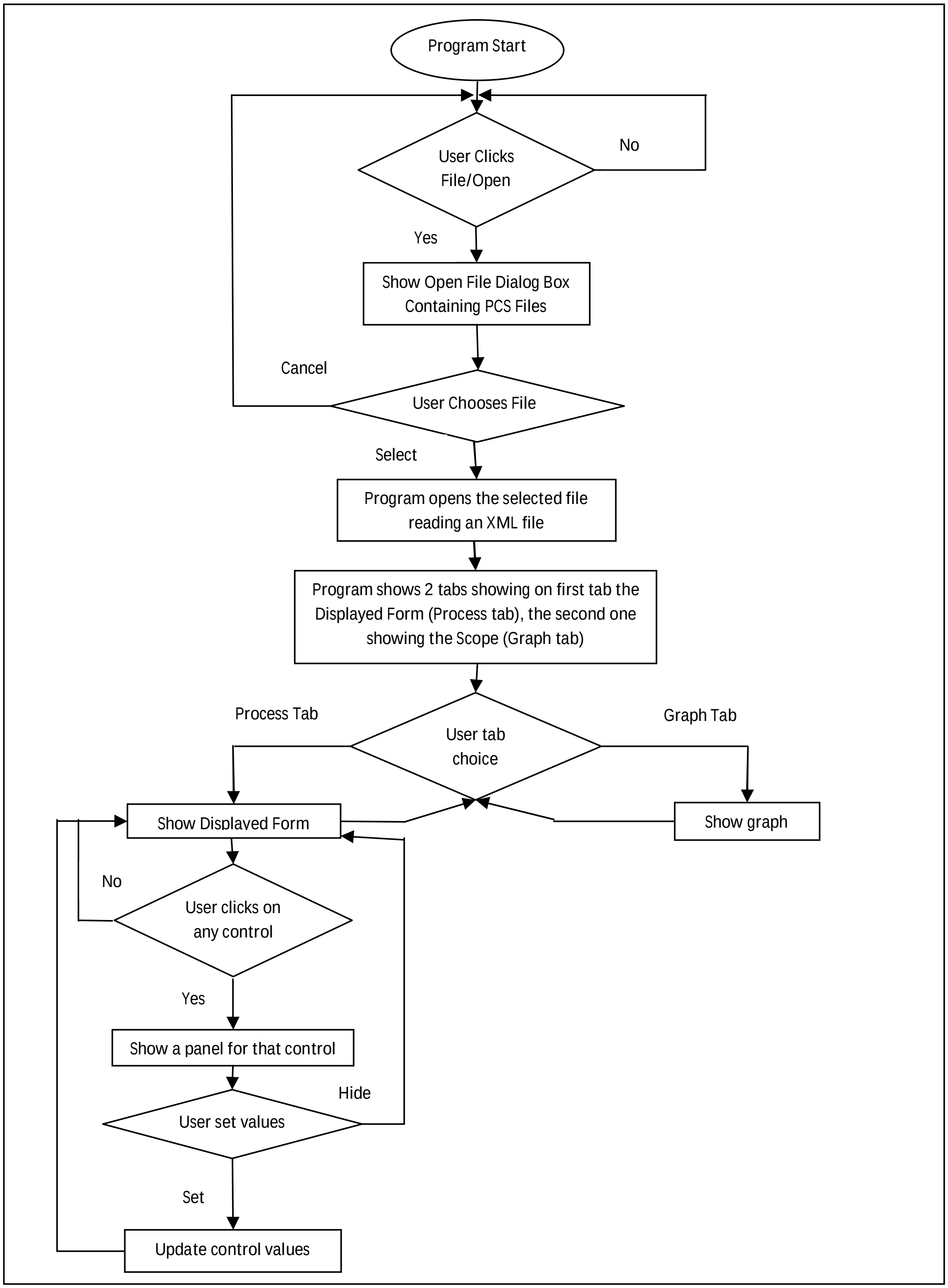

Program Start
User Clicks File/Open
No
Yes
Show Open File Dialog Box Containing PCS Files
Cancel
User Chooses File
Select
Program opens the selected file reading an XML file
Program shows 2 tabs showing on first tab the Displayed Form (Process tab), the second one showing the Scope (Graph tab)
Process Tab
User tab choice
Graph Tab
Show Displayed Form
Show graph
No
User clicks on any control
Yes
Show a panel for that control
Hide
User set values
Set
Update control values

Figure 11 .6 Flowchart showing sequence of XML reading part in the program, note that termination of the program can occur at anytime of clicking Close

B.1 Socket Programming [4]

Network programming in WinCE is possible with sockets. A socket is like a handle to a file. Socket programming resembles the file IO as does the Serial Communication. Sockets are a communication mechanism.

A **socket** is defined by a group of four numbers, these are

- The remote host identification number or address
- The remote host port number
- The local host identification number or address
- The local host port number

B.2 Protocol Used in This Project

As shown in the figure each layer of the Network Protocol Standards consists of series of protocols our concern will be the protocols of the Transport layer which are TCP/IP (Transmission Control Protocol/Internet Protocol) and UDP (User Datagram Protocol).In our project we agreed to use the TCP/IP protocol for it's efficiency and connection reliability (Connection Oriented Protocol).Also TCP/IP is also in widespread use on commercial and private networks. In according to that we managed to create a protocol of our own that is included in the index of this documentation.

B.3 Program Sequence:

1. After reading all the controllers from the xml file the user would have to press the connect button and select the name of the process he wants to connect to.

2. The program goes through the protocol illustrated in the index to connect to the server built on the microcontroller.

3. If no error occurred or nothing goes wrong the program will start a thread to listen and receive data from the server obtaining all information about the process and redrawing all components and controllers with their received values on the process tab of the form.

4. Pressing the graph tab will display a tab page that contains a graph that represents the response of each tank on the displayed form with different colors for different types of controlled elements, shown in Fig.7 and Fig.8 .

[4] For more information about sockets refer to “PDA Sockets Programming” in the Appendix.

5. The user is now able to see the current running process with its values and he can adjust any of the control parameters of any component in a friendly user interface way

6. If any of the control parameters is adjusted a send button will become visible to the user.

7. Pressing that send button will start a loop whose task is to get all the parameters of the controls displayed on the form and send them to the microcontroller which in turn will adjust the actual process to the received values.

8. It will start the thread that receives from the server the parameters of the controllers and redraw them with their adjusted values.

9. When ever the displayed form is closed the socket built to handle this connection is closed and the connection is terminated.

A flow chart is shown in Fig.9 representing the sequence of the connection part of the program.

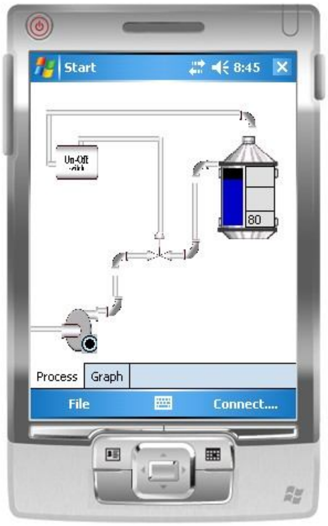


Figure 11.7 A process running on the process panel

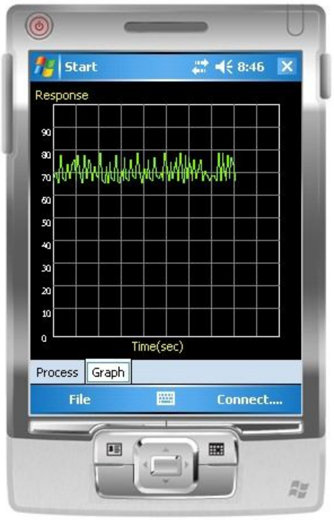


Figure 11.8 The corresponding graph for the tank shown in Fig11.7

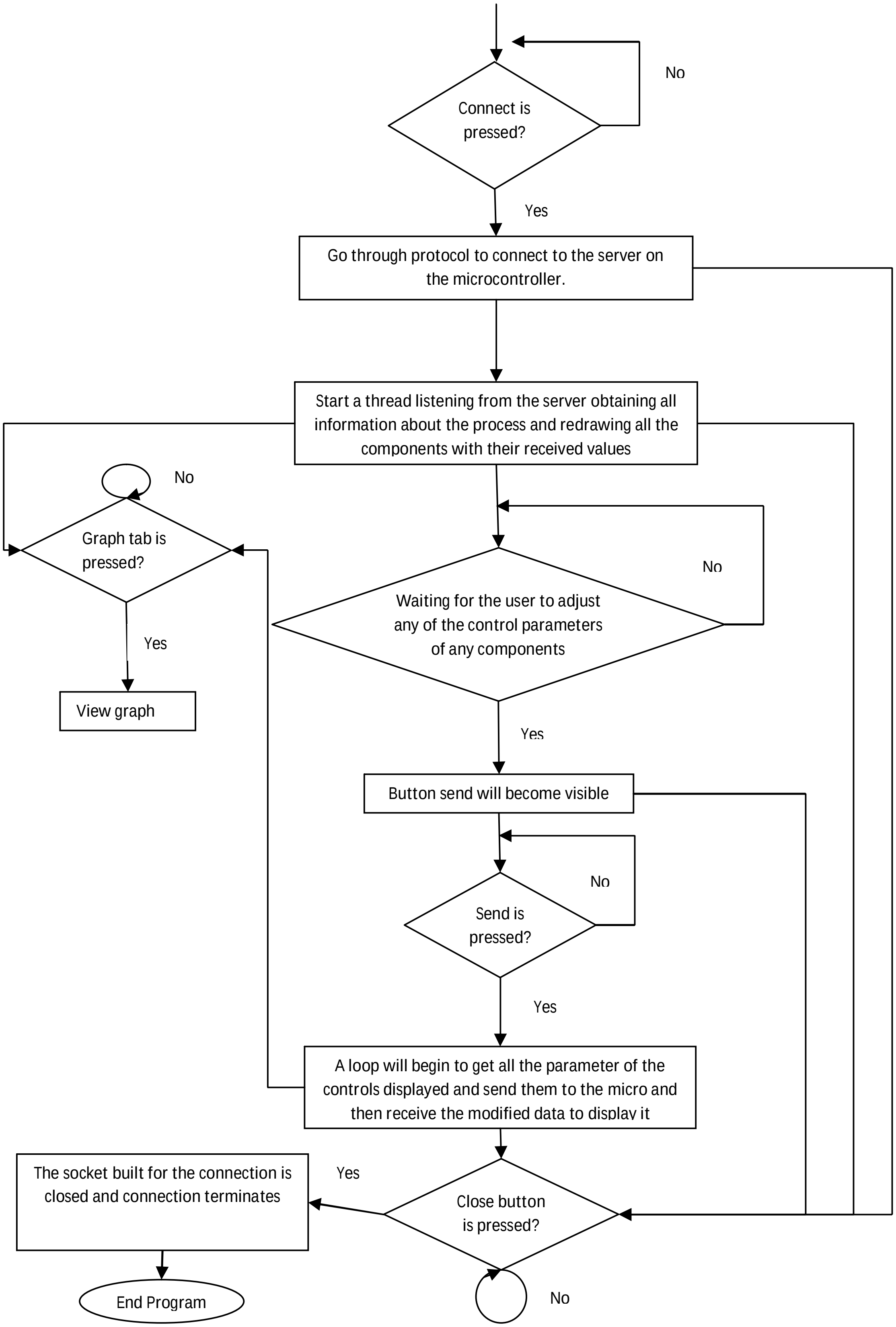

No
Connect is pressed?
Yes
Go through protocol to connect to the server on the microcontroller.
Start a thread listening from the server obtaining all information about the process and redrawing all the components with their received values
No
Graph tab is pressed?
Yes
View graph
No
Waiting for the user to adjust any of the control parameters of any components
Yes
Button send will become visible
No
Send is pressed?
Yes
A loop will begin to get all the parameter of the controls displayed and send them to the micro and then receive the modified data to display it
The socket built for the connection is closed and connection terminates
Yes
Close button is pressed?
End Program
No

Figure 11.7 Flow chart of the connection procedure.