32 is used to register a selection onto the touch input device. The dimensions of the graphic touch entry options on the touch-sensitive input overlay **16** are sized to allow easy selection by a human finger or the physical pointer device.

[**0032**] While only one touch-sensitive input overlay **16** is shown in **FIG. 1, I** envision other types of touch-sensitive input overlays as well, such as a numeric pad (also called an "addition control"), an abbreviated keypad, and a navigation pad with four directions of movement selection, as well as other options consistent with traditional navigation support. Some of these embodiments are presented below and in the accompanying figures.

[**0033**] **FIG. 2** is a hardware and communication flow diagram of the touch-sensitive input overlay. The left side of the diagram shows a functional overview of the hardware and software components **42**, while the right side shows a general data and functional flow graph **44** between these components.

[**0034**] Beginning with the hardware and software components **42**, these are shown as the primary functional components of a system in which my invention can be deployed. A microprocessor **46** is the primary agent for executing program modules and instructions and communicating between devices. In normal operation, this is achieved through additional components (not shown) such as an operating system and device drivers stored in memory. The microprocessor **46** has access to at least two such memory areas: an execution memory **50** (such as RAM) and a persistent memory **48** (such as ROM and disk storage). The microprocessor **46** is further communicatively coupled to a display device **56**, such as a cathode ray tube, active matrix, passive matrix, or liquid crystal display.

[**0035**] Preferably the display device **56** further includes a touch input capability, which is depicted as a touch input device **58**, as it may be integrated with the display device **56** or a separate element capable of detecting a touch input on a display device (e.g., an optical or infrared sensors configured to intercept an object coming into contact with the display device **56**, or a screen that overlays the display device **56**). The touch input device **58**, then, is configured to detect a touch input and generate a signal indicative of the touch input together with a signal indicative of the two dimensional coordinates identifying the location where the touch input was received relative to the touch input device **58** or display device **56**. In this way, a particular control can become the focus.

[**0036**] If the functionality of the touch input device **58** is integrated into the display device **56**, then communication between the touch input device **58** and microprocessor **46** will typically be handled through a communication link between the microprocessor **46** and display device **56**.

[**0037**] Two GUI program modules **51** are called out from the memory areas **48** and **50** to emphasize an embodiment. The first is GUI program module **52**, which is a standard GUI. By standard GUI, it is meant that an application program written over the operating system has a GUI that typically operates with the aid of a mouse or keyboard. (This standard GUI does not have to be modified according to an embodiment of my invention.)

[**0038**] The GUI module **52** can be implemented in a number of different programming languages and for a number of different applications. One example is a GUI programmed in VisualBasic (TM), available from Microsoft Corporation in Redmond, Wash., for a semiconductor manufacturing equipment. The GUI can include a number of controls designed to monitor and regulate the fabrication of semiconductors within the semiconductor manufacturing equipment. Another example is a GUI programmed with a Java (TM) development kit, such as an abstract window toolkit (AWT) or Swing toolkit. Java (TM) implementations of both are available from a number of vendors including Sun Microsystems, Inc. in Palo Alto, Calif. Moreover, the touch-sensitive input overlay can be used for entry of data for control, monitoring, or other record keeping operations for any industrial, commercial, or other purpose.

[**0039**] The standard GUI module **52** includes programming modules that handle data entry or navigation when it is entered with a mouse or keyboard into the parent control. The GUI module **52** further includes the graphics and program operation calls that can drive underlying application processes—for example calls to execute routines that create a new set point or parameter value for an external control process, or calls to routines that perform a calculation based on data that entered into the parent control.

[**0040**] The touch-sensitive input overlay module **54** is new. Its primary function is to cause a touch-sensitive input overlay to be presented near a control when the control is touched (thus becoming the focus), and generate command/input signals for further processing by the touch-sensitive input overlay module **54**, as well as the GUI module **52**. This function can supplant the role of the keyboard, thereby allowing a user to enter data directly into the touch-sensitive input overlay through one or more "touch key" commands directed toward the touch input device **58**. Additional details of the touch-sensitive input overlay module **54** are provided below with reference to **FIG. 4**.

[**0041**] According to one embodiment, the touch-sensitive input overlay module **54** includes an interpreter module **55** that is configured to translate touch inputs received in the touch-sensitive input overlay into corresponding keyboard entries so they can be passed along to the microprocessor **46**, which, in turn, passes them along o the GUI module **52** or the application program (e.g., by adding them to a queue associated with a particular thread). However, in other embodiments, such an interpreter module **55** can be added to the operating system so that incoming touch inputs from the touch input device **58** can be processed without passing through the touch-sensitive input overlay module **54**.

[**0042**] Turning to the functional flow graph **44**, a horizontal line is shown to each element in the function component stack **42**. An arrow shows a direction of communication travel. The description begins with the GUI module **52** instructing the microprocessor **46** to present graphical information (GI). In turn, the microprocessor issues instructions (P1) that cause the display device **56** to present the graphical information, which includes controls.

[**0043**] Once the graphical information is presented, a touch input is received at the touch input device **58**, which then sends a signal (D1) back to the microprocessor **46** indicating that a touch-input has been received. The signal (D1) preferably includes location information indicating coordinates where the touch input was received. The signal is received at the microprocessor **46**.