CHAPTER 14 UNIVERSITY OF MASSACHUSETTS AT DARTMOUTH

School of Engineering Department of Electrical Engineering Technology North Dartmouth, Massachusetts 02747

Principal Investigator:

Lester W. Cory (508) 999-8482 Robert C. Helgeland (508) 999-8480 Philip H. Viall (508) 999-8240

Single Switch Serial Mouse Emulator

Designer: Joseph Santamauro

Client Coordinator: Prof. Les Cory, UMASS Dartmouth Centerfor Rehabilitation Engineering Supervising Professors: Philip H. Viall, Thomas Ellis Electrical Engineering Technology Department University of Massachusetts Dartmouth North Dartmouth, MA 02747

Introduction

Many computer programs utilize a pointing device such as a mouse. Some programs are enhanced by the use of a mouse, and some require a mouse. Many physically disabled individuals lack the ability to control a conventional mouse. For this reason they are unable to use certain programs. The single switch serial mouse emulator is controlled via a single momentary switch, while appearing to the computer as a conventional mouse. With this device, a disabled person who can operate any momentary switch, can effectively use any software package which relies upon a mouse for input. Since this device is a hardware solution, it does not suffer from the limitations inherent in many software based solutions. A photograph of the single switch serial mouse emulator is shown in Figure 14.1.

The single switch serial mouse emulator has four groups of LED's from which the user selects the desired mouse function. The first group of four LED's (top left) are used to select pointer movement: left, right, up, and down. The second group of three LED's (top right) select one of three button single clicks. The third group of three LED's (bottom left) select hold button and drag functions. The fourth group of three LED's (bottom right) select button double clicks. The emulator scans through each function group by illuminating all LED's in each group in succession. The user selects a group by pressing the switch when the desired group is illuminated. Once a group is selected, each LED in that group is illuminated one at a time. A function is then selected by pressing the switch when its corresponding LED is on.

Summary of Impact

The single switch serial mouse emulator enables a person with spastic quadriplegia associated with cerebral palsy to utilize software which requires the use of a mouse. Previous software solutions were ineffective in that they did not work with all software packages, and often the results were unpredictable.

The mouse emulator plugs directly into the serial port of the computer, and from the computer's standpoint it is indistinguishable from a standard mechanical mouse. Both this project and the other (parallel) mouse designed at the same time will be evaluated to determine if either is more functional than the other for the intended application.

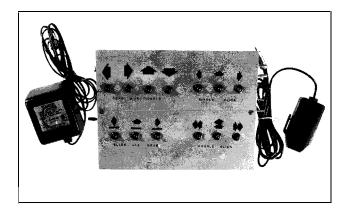


Figure 14.1. Photograph of Single Switch Serial Mouse Emulator.

Technical Description

The brain of the mouse emulator is an 8748 microcontroller, as shown in Figure 14.2. This microcontroller controls the scanning of the LED's through the lower four bits of port 1. The rate of LED scanning is determined by a subroutine programmed into the 8748. The pins of port 1 are connected to a 4-line to 16-line decoder (74154). The 74154 in turn drives each of the LED's. Each LED can be turned ON by applying it's corresponding 4 bit code to the inputs of the decoder. The input code drives the corresponding output low, thereby enabling current to flow through the LED causing it to emit light. Since only one 4 bit code can be applied at once, only one LED can be on at any one time. Although by applying codes in sequence at a rapid rate (multiplexing) more than one LED will appear to be on.

The scanning rate of the LED's is controlled by a set of dip switches that are connected to bit 0 and bit 1 of the Data Bus. The state of these switches is monitored by the delay subroutine to determine how long to wait before returning. The rate at which the mouse pointer (on the computer) moves is controlled by an additional set of switches. These switches are connected to bits 4 and 5 of the Data Bus. These switches determine the rate at which movement codes are sent to the computer, thereby controlling the rate at which the pointer moves on the computer screen.

The mouse functions (movement, button presses etc.) are controlled by a single momentary switch. The software provides for switch bounce via a delay. The switch is connected to the T1 test line, which is pulled high via a pull up resistor. Pressing the switch completes a path to ground and pulls T1 low. Conditional branch instructions which test the T1 line are used to determine if the switch is being pressed. When the switch is pressed the function corresponding to the current state of the LED's is performed. Mouse codes are sent serially via bit 0 of port 2. The serial data format consists of a start bit, data bits, and two stop bits, at a rate of 1200 bits per second. Since the ports on the 8748 are TTL compatible and the mouse connects to an RS-232 port on the PC, a MAX232 transceiver is used. The MAX232 transceiver converts the TTL input voltages to RS-232 compatible output voltages. Although not used in this design, the MAX232 can convert RS-232 input voltages to TTL output voltages. It has two RS-232 transmit buffers (TTL to RS-232) and two receive buffers (RS-232 to TTL). It is powered via a single 5V supply, utilizing internal charge pumps and external storage capacitors to generate the required RS-232 voltages.

The mouse emulator is powered via a 120-VAC to 7.5-VDC wall adapter. The 7.5-V output of the adapter drives an LM7805 (5-V) regulator. This 5-V output powers the 8748, MAX232, 74154, and each of the LED's.

This device will completely replicate the functions of a Microsoft compatible three button serial mouse, and can be duplicated with a materials cost of about **\$75**.

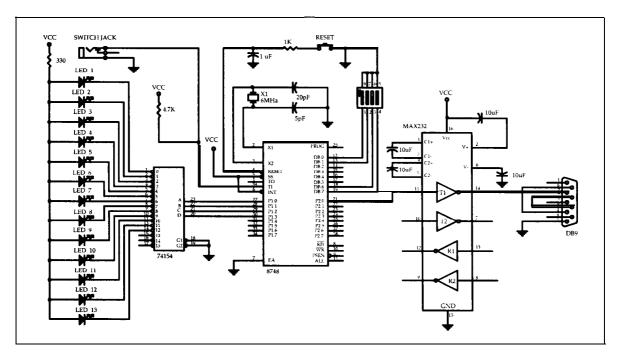


Figure 14.2. Schematic Diagram for Single Switch Serial Mouse Emulator.

Single Switch Bus Mouse Emulator

Designer: Miki J. Barton

Client Coordinator: Prof. Les Coy, UMASS Dartmouth Centerfor Rehabilitation Engineering Supervising Professors: Philip H. Viall, Thomas Ellis Electrical Engineering Technology Department University of Massachusetts Dartmouth North Dartmouth, MA 02747

Introduction

This project is similar to the serial mouse emulator reported elsewhere in this chapter, except that this project emulates a bus mouse (parallel) rather than a serial mouse. Many of today's computer applications require the use of a mouse, or some form of pointing device. Many physically disabled individuals lack the ability to control a conventional mouse. For this reason they are unable to use these programs. The Single Switch Bus Mouse Emulator is controlled via a single momentary switch, while appearing to the computer as a conventional mouse.

The single switch bus mouse emulator has six groups of LED's from which the user selects the desired mouse function. The first group of eight LED's are used to select pointer movement: up, down, left, right, NE, SE, NW, and SW. The second group of three LED's select button single clicks. The third group of three LED's select button double clicks. The fourth group of three LED's select hold button and drag. The fifth group of LED's are used to change the rate of mouse pointer movement. The last group of LED's are used to change the scanning rate. The emulator scans through each function group by illuminating all LED's in each group in succession. The user selects a group by pressing the switch when the desired group is illuminated. Once a group is selected each LED in that group is illuminated one at a time. A function is then selected by pressing the switch when its corresponding LED is on. The device is shown in Figure 14.3.

Summary of Impact

The single switch bus mouse emulator (like the serial mouse emulator) enables a person with spastic quadriplegia associated with cerebral palsy to utilize software which requires the use of a mouse. Previous software solutions were ineffective in that they did not work with all software packages and often the results were unpredictable. The mouse emulator plugs directly into the bus port of the computer and from the computer's standpoint it is indistinguishable from a standard mechanical mouse. Both this project and the other (serial) mouse designed at the same time will be evaluated to determine if either is more functional than the other for the intended application.

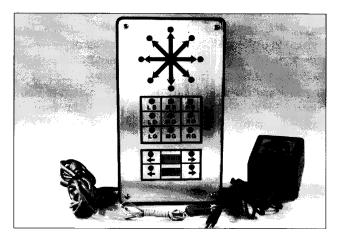


Figure 14.3. Photograph of Single Switch Bus Mouse Emulator.

Technical Description

The brain of the single switch bus mouse emulator (SSBME) is an 8748 micro-controller, as shown in This micro-controller controls the Figure 14.4. scanning of the LED's through the data bus. The lower four bits of port 1 are used to select which of the five 74LS374s (&bit latch) are to latch the value placed on the data bus. Each of the latches are selected by placing its corresponding value on port 1. This value drives a 74LS145 (4 to 16 decoder). The output of the decoder and the write (WR) line of the 8748 is combined using a 7432 (2-input OR gate) to drive the clock line of the latches. The 74LS374s drive each corresponding LED via a current limiting resistor (560 ohms). Each LED can be turned ON by latching a zero into it's corresponding latch. A low bit at the latch drives the corresponding output low,

thereby enabling current to flow through the LED causing it to emit light. The rate of LED scanning is determined by a subroutine programmed into the 8748. This rate can be controlled by the user, to speed it up or slow it down.

The mouse functions (movement, button press etc.) are controlled by a single momentary switch. The software provides for switch bounce via a delay. The switch is connected to the INT test line, which is pulled high via a pull up resistor. Pressing the switch completes a path to ground and pulls INT low. A conditional branch instruction tests the state of the INT line to determine if the switch is being pressed. When the switch is pressed the function corresponding to the state of the LED's is performed.

Mouse codes are sent via port 2. The bus mouse card has nine pin connections. Pin 1 supplies 5 volts, pins 2 and 3 (X2 and Xl respectively) control horizontal movement, pins 4 and 5 (Y1 and Y2 respectively) control vertical movement, pins 6, 7, and 8 indicate button presses for Left, Middle, and Right buttons respectively, pin 9 is circuit ground. The 5 volts at pin 1 is used to power the conventional bus mouse, but the current draw of the SSBME is more than can be supplied by this line. As a consequence,

the SSBME requires an external 5 volt power supply. Button presses are simulated by driving the appropriate button input low. When the button is released the line is returned to its inactive high state. Movement in the horizontal direction is generated by feeding two square waves into the X input lines, one for Xl and one for X2. The direction of movement is controlled by the phase shift of the two square waves. If the Xl line is high during the low to high transition of the X2 line, the pointer is moved one position to the right. If the Xl line is low during the low to high transition of the X2 line, the pointer is moved one position to the left. In the case of the Y1 and Y2 lines, the state of the Y1 line during the low to high transition of the Y2 line specifies the direction of movement in the vertical direction. The mouse emulator is powered via a 120 volt AC to 7.5 volt wall adapter. The 7.5 volt output of the adapter drives an LM7805 (5 volt) regulator. This 5 volt output powers the 8748, 74LS145,74LS374, 7432, and each of the LED's.

This device will completely replicate the functions of an Microsoft compatible three button bus mouse, and can be duplicated with a materials cost of about \$75.

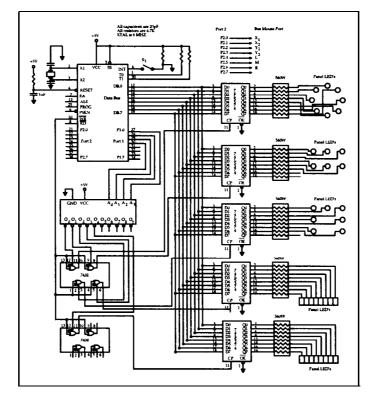


Figure 14.4. Schematic of Single Switch Bus Mouse Emulator.

