LightTouch Limited
Simon Fraser University
Burnaby BC
V5A 1S6
jmy@sfu.ca

March 11, 1998

Steve Whitmore
School of Engineering Science
Simon Fraser University
Burnaby BC
V5A 1S6

Re: ENSC 370 LightTouch Limited Design Specification

Dear Mr. Whitmore,

Hello and thank you for your interest in the technologies and processes used at LightTouch Limited. The enclosed document, LightTouch Limited Design Specification, details design specifications for our prototype PC pointing device. We review a required component list, hardware component operating characteristics, hardware schematics and software flowcharts along with textual explanations aimed at describing the specific composition of our project.

Our design specification will allow you to examine the inner workings of our next generation PC pointing device. If you have any questions or concerns, please feel free to contact me at jmy@sfu.ca.

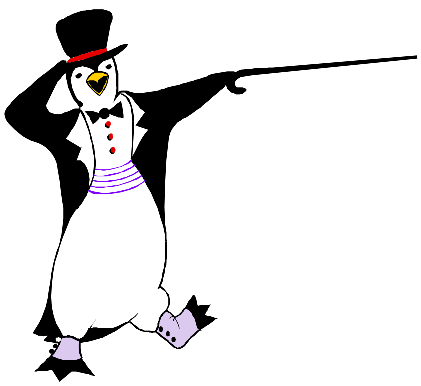
Yours Truly,

Jonathan Young
VP Customer Support
LightTouch Limited

Enclosure: ENSC 370 LightTouch System Design Specifications

LightTouch System

Design Specification



LightTouch Limited
Simon Fraser University
Burnaby BC
V5A 1S6
jmy@sfu.ca

March 11, 1999

Steve Whitmore
Simon Fraser University
8888 University Drive
Burnaby BC

Abstract

The LightTouch system will be a PC user interface device. By physically pointing to the screen, users will be able to create an action similar in concept to clicking a mouse at a pointed to screen location, by disturbing ambient light around the screen. Four linear arrays of optical sensors positioned around the flat plane of a computer display will receive the optical disturbance, sending data to the ADC of a microcontroller. Lenses, physical vision restrictors and light sources will intensify input to the optical sensors. The microcontroller will be responsible for sequencing the input from the four linear arrays into a corresponding 4-bit parallel port output stream. Other responsibilities of the microcontroller will include basic data compression and image processing.

LightTouch PC driver software retrieves image data from the parallel port and compares it with a previous 'null state' image coupled with internal state variables measuring button state, previous cursor location and delay between changes. Driver software will produce a x-y screen coordinate and a user movement type: single click, double click, or drag. Finally, application software, interfacing the with driver library, will report driver information into a form meaningful to the user with onscreen demonstrations of pointer movement.

Table of Contents

List of Figures

1. Introduction

Graphical user interfaces allow us to communicate our desires to our computers quickly and intuitively. The recent development of computer touch screens provides computer users a more interactive and intuitive user interface. However the computer touch screen available on the market is not practical to the large-scale display device, such as a projection screen, because of its size and cost. The LightTouch touch screen system is designed to overcome these two disadvantages of the traditional touch screen, using inexpensive and reliable optical sensors positioned around a screen to determine pointer location. The LightTouch touch screen system will also have portability and flexibility such that it can be easily attached to different display devices of different sizes.

The purpose of this document is to describe the design details of the LightTouch system. The audience for this work is Dr. Andrew Rawicz, Mr. Steve Whitmore, the design engineers of LightTouch Limited, and any external parties interested in the LightTouch system.

2. System Overview

The LightTouch touch screen system recognizes the location of the physical pointing object on the display by detecting the intensity change in the visible light pattern. The intensity pattern is processed and translated into the x-y coordinates of the physical pointer.

Figure 1 shows a detailed LightTouch system block diagram. The functionality of each unit is explained in more detail in later sections.

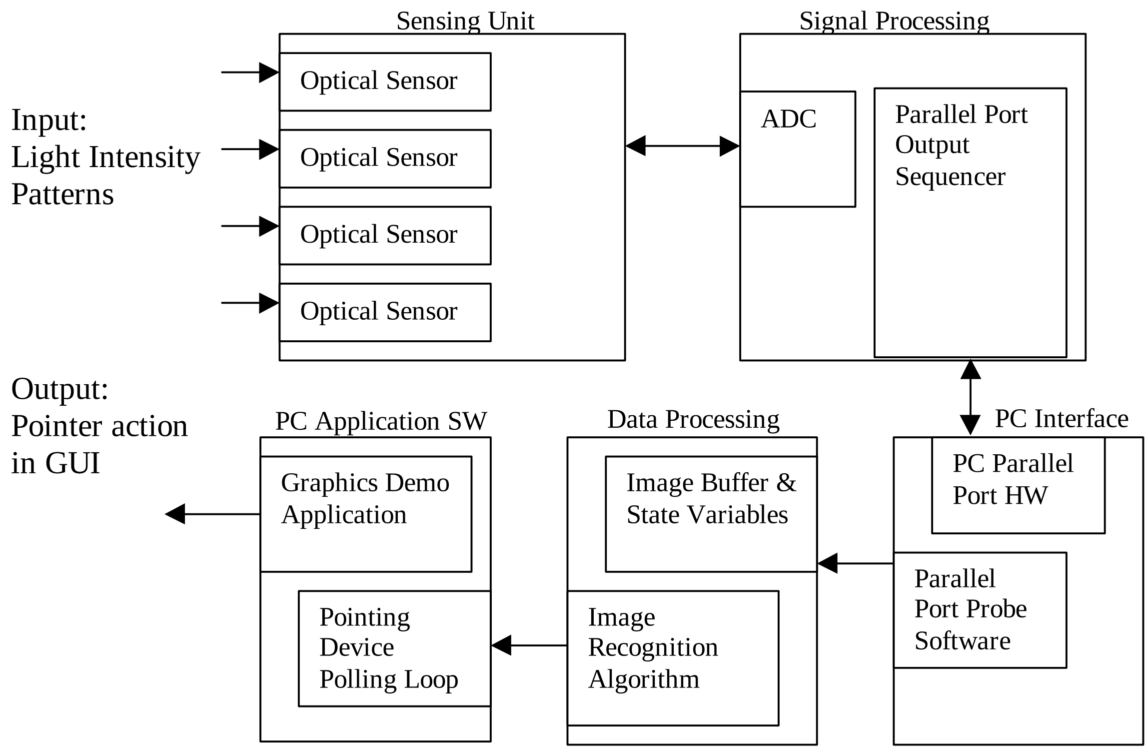


Figure 1.1 Detailed System Block Diagram

A disturbance in the light intensity patterns originating from movement of a physical pointing device around the computer display area is read by four optical sensors. Image data from all four sensors are read and then sent the PC parallel port 4 bits a time. Image data is then moved off the parallel port into an image buffer until the full images from all original optical sensors has been received.

The image recognition algorithm then goes to work comparing the new image with a 'null image' gathered during product setup. Depending on whether the image is significantly different enough to warrant pointer presence, as well as memory of recent pointer states, allows the driver to determine current pointer state and movement type: single click, double click, or drag. Application software will use the state information reported by the driver to show pointer movements onscreen corresponding to the original optical sensor input. It should be noted that the LightTouch system will be implemented using a polling technique, so that the entire process mentioned is initiated at the request of the application software functional block.

Operating characteristics of parts used are mentioned in the References section at the end of this document, and a full parts list is located in the appendix.

3. Product Casing

We will physically package the two sensing units and the processing unit separately, as show in Figure 3.1 below.

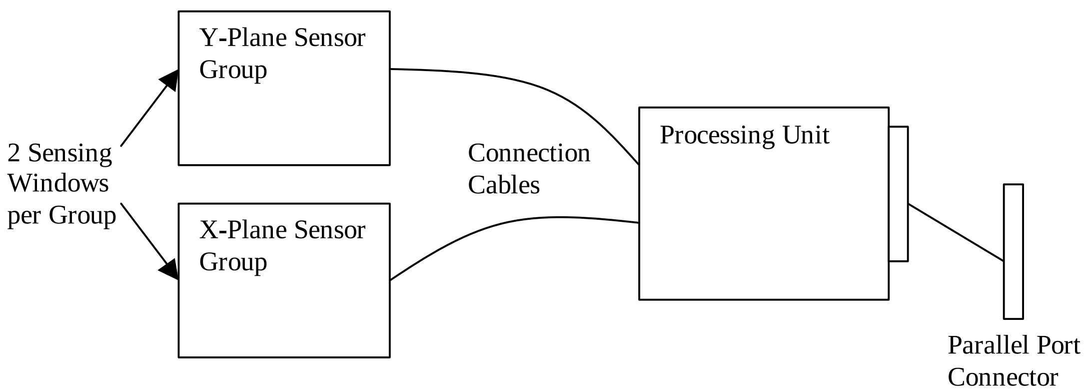


Figure 3.1 Product Casing Diagram

The two sensing units must be in separate cases to facilitate placement around the computer display. The three packages will be made out of sheet metal because it is easy to work with and inexpensive. The X and Y sensor groups will be mounted on the sides of the computer display using pads with removable glue on both sides.

4. Hardware Description

The first major stage of the LightTouch system concerns the physical device hardware component. Figure 4.1 below shows the context diagram of this stage. The functional blocks involving hardware design are highlighted below.

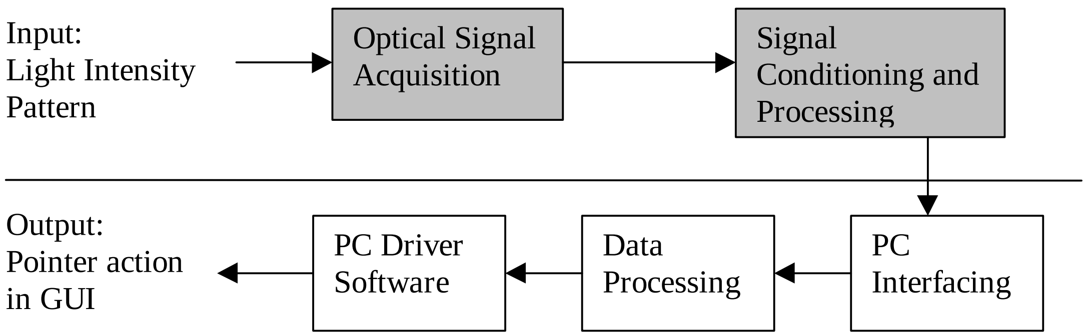


Figure 4.1. Optical Signal Acquisition Stage Context Diagram

4.1. Hardware Unit Design Plan

The Light Touch system's hardware configuration is shown in Figure 4.2.

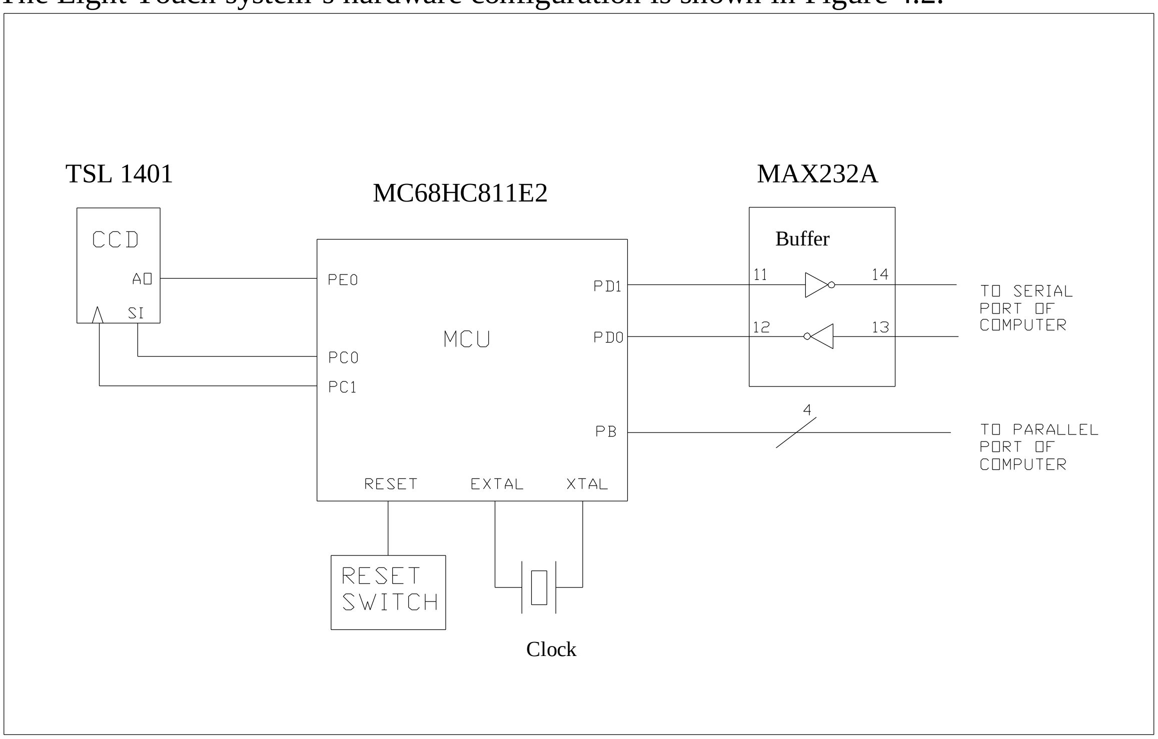


Figure 4.2: Hardware schematic

The CCD's[1] shown in Figure 3.2 is a linear array of photodiodes used to detect the location of the physical pointing device. Two group of CCD's are used, one for the determination of x co-ordinate and the other for y co-ordinate. The output of the CCD is analog voltage representing the light intensity of a particular pixel. The voltages of all the pixels are shifted sequentially to the MCU for digitization.

The CCD chosen for the system is the TSL 1401 from Texas Instruments. It offers 128 pixels per package and a built-in amplifier for output. It also has one of the lowest price per pixel advantage. Because the system requires a minimum resolution is 160 pixels, two TSL 1401 are used together to give a resolution of 256 pixels for each channel (x or y).

The microcontroller is used for controlling the CCD's, digitizing their analog outputs, and transferring them to the PC for image processing. It will drive the CCD's clock at 5kHz, while collecting frame (pixels for 1 channel at a given time) at 10Hz. 10Hz is determined in the functional specification as the optimum for accuracy against the size of the resulting data. SI is the pin used to signal the beginning of a frame.

After receiving the data the MCU converts the analog signal from each pixel to an 8-bit digital signal. These 8-bit signals are then transferred to the computer via the parallel port. The 8-bit signals are sent to the computer immediately after A/D conversion because the MCU's limited RAM will not facilitate storage of data as well as signal processing. The parallel port is chosen for data transfer because its bandwidth is adequate for this application while the serial is not.

The MC68HC811E2[2] from Motorola is the MCU used in the Light Touch system. It is chosen for the following reasons: it offers 8 A/D channels, where the requirement is 4; it offers ~ 40 I/O pins, where the system requires at least 13 pins; and it can be programmed serially without a MCU programmer.

The MAX 232A component is used only when programming the MCU. The MCU is programmed using the serial port. The MAX 232A converts the TTL inputs from the MCU to RS232 signals understood by the serial port.

[1] Specifications for CCD in Appendix
[2] Specifications for MC68HC811E2 in Appendix

4.2. Hardware Unit Test Plan

To test the proper operation of each of the components in the system a piecewise method of construction should be used. Output signals for known input signals is examined for validity. The operation of the CCD's can be tested prior to connecting it to the MCU. The CCD can be driven by a function generator and the output can be viewed with the use of an oscilloscope. Meaningful data should be obtained reliably before data transfer to the MCU should be attempted.

The MCU should be tested for proper analog to digital conversion, timing and I/O signaling. This will involve testing of both the hardware and software of the MCU. Again testing is done prior to the MCU's integration into the system.

Finally when we are satisfied with the proper operations of the internal components the system maybe connected together one at a time. An overall testing of the system will follow.

This sort of piecewise testing will allow us to isolate components and localize any errors. Errors can then be quickly and efficiently determined.

5. Software Description

The second major stage of the LightTouch system is the software component. Figure 5.1 below shows the context diagram of this stage. Software intensive functional blocks are highlighted shown below.

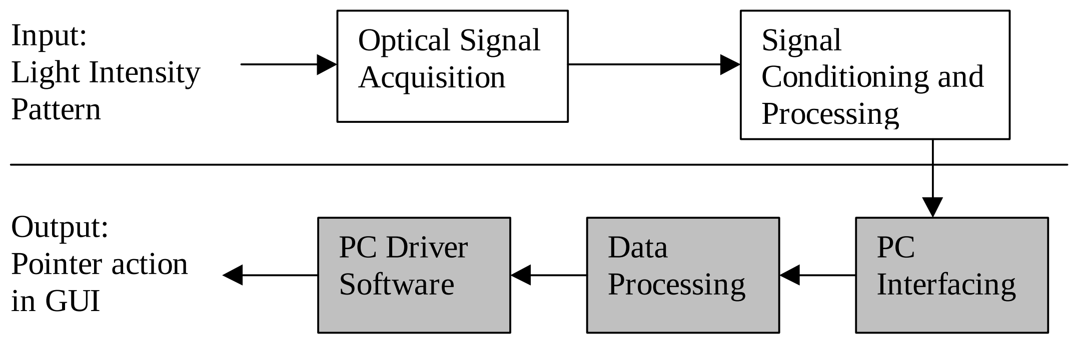


Figure 5.1. Signal Conditioning and Processing Stage Context Diagram

5.1. Description of Algorithm

The image processing step is designed to receive the sampled data from the microprocessor, determine x-y coordinates and pointer status from the input data, and interpret the pointer action. The processing is to be performed on the host computer using the driver software. The software consists of two major algorithms: Image Recognition and Object Location, and Action Interpretation. Figure 5.2 shows the flowchart for Image Recognition and Object Location.

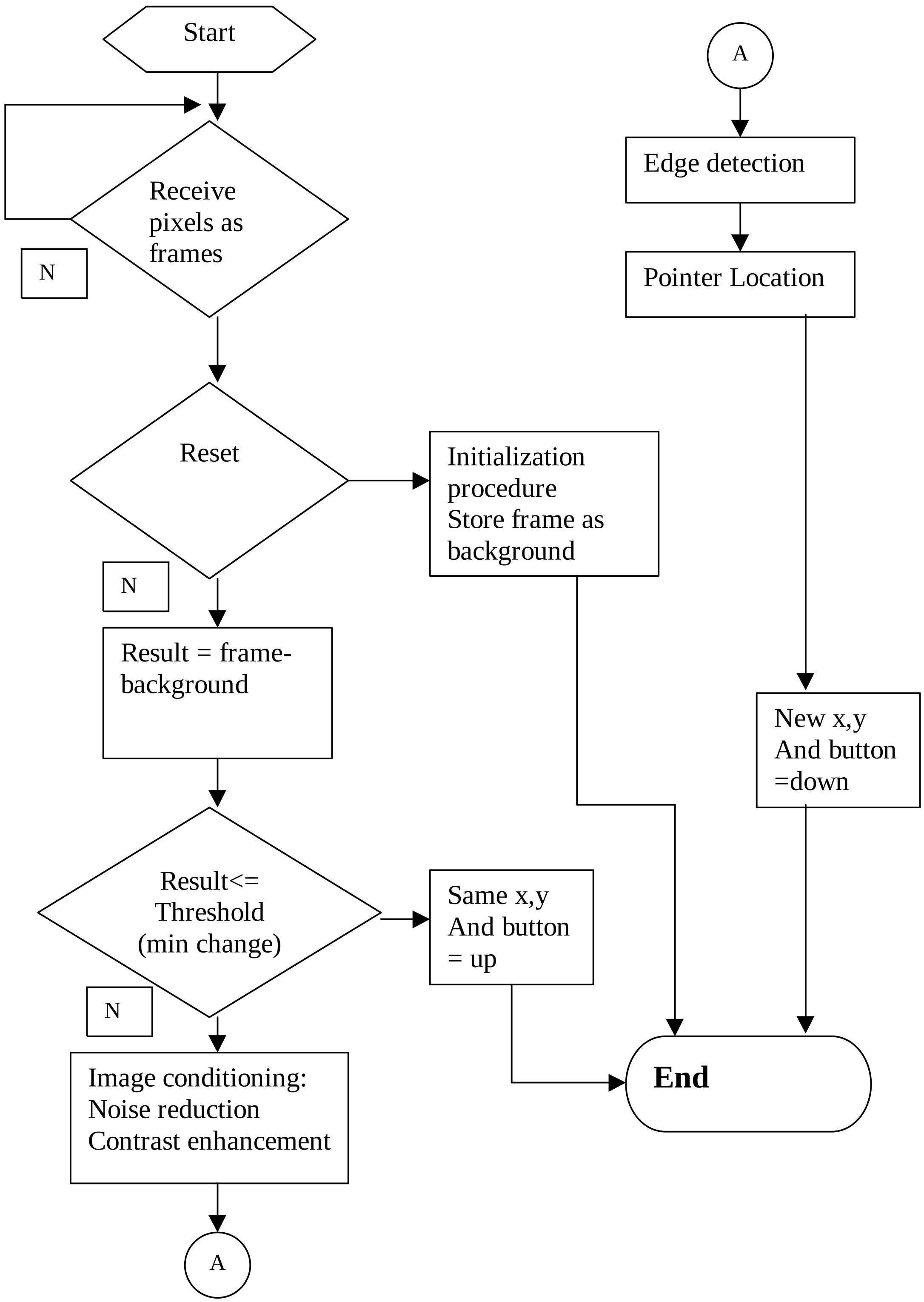


Figure 5.2: Flow Chart for Image Recognition and Object Location

The Image Recognition and Object Location algorithm first receives the sampled data (horizontal and vertical pixels) as frames. During the initialization step, the program saves the frames as the reference background image. If the system is already initialized, the program sets intermediate variable Result, an array of values representing the image change with respect to the reference background.

If the change in background, Result, is smaller than the preset minimum change threshold, then the program reports no change in pointer location, nor x-y coordinates (x,y are the same), and that the pointer object has no contact with the display (button status: b = up).

The program can determine the pointer object location by edge detection as illustrated in Figure 5.3 below.

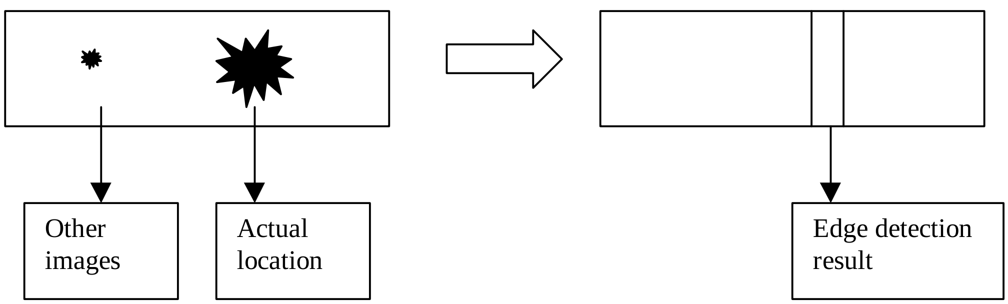


Figure 5.3: Illustration of Edge Detection

The location of the pointer object can be determined as the center between the two edges from the edge detection result. Consequently the Pointer Location procedure will set the new x-y values and set b = down because the presence of the object on the screen.

At the end of the Image Recognition and Object Location algorithm, three new state variables, (x, y, b) will be available to the Action Interpretation algorithm whose flow chart is shown in Figure 5.4 next page.

New state variable (x, y, b) from the Image Recognition and Object Location algorithm
Last state variable (x0, y0, b0)

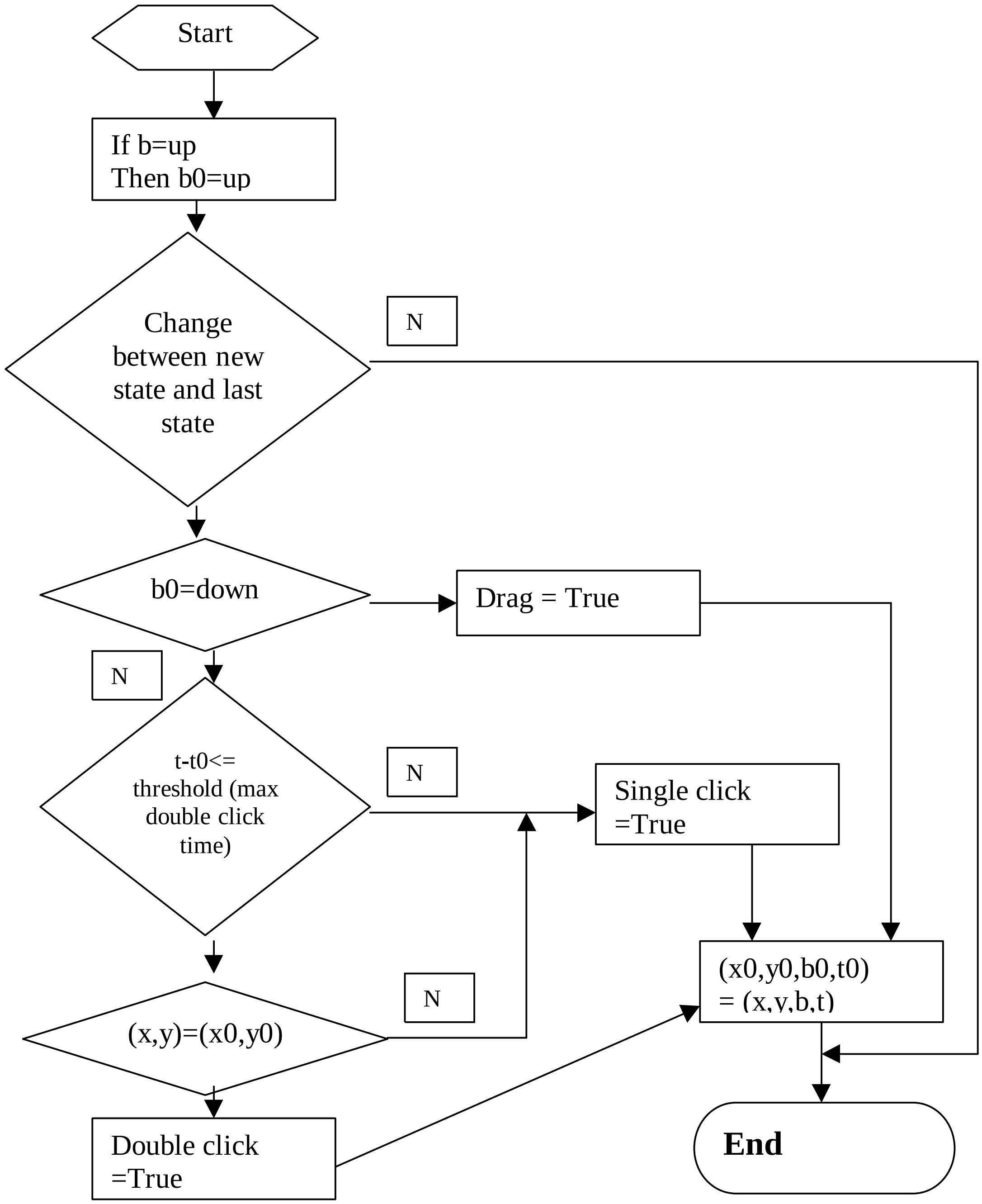


Figure 5.4: Flow Chart for Action Interpretation

The Action Interpretation algorithm first sets b0=b if b=up. This procedure allows the program to determine the pointer action in subsequent steps. The algorithm then compares the new state (x, y, b, t), generated from the Image Recognition and Object Location algorithm, with the stored last state (x0, y0, b0, t0). If there is no change, the program determines that the user has performed no action on the pointer. If there is a change, this change is reported as the user clicking the pointer.

The program will determine whether the user performs a single click or a double click by two comparison procedures. If the time difference between the present click and the last click is more than a preset maximum double click time threshold (t-t0>thresold), then the program considers the click action as a single click. In addition, if t-t0<=threshold, the program will further check whether the click occurs on the same location ((x, y)=(x0, y0)). Different locations reveal that the click action is the single click.

At the end of the Action Interpretation algorithm, both the location and the action of the pointer will be determined and sent to the operating system.

5.2. Software Units Test Plan

Two test programs will be required to test the accuracy and robustness of the Image Recognition and Object Location and the Action Interpretation program.

To test the Image Recognition and Object Location program, manually generated image data simulating the presence of the pointer object on the screen will be used to test the program for function correctness. The main functions targeted by this test are:

- whether it can respond to reset properly and initialize the reference background,
- whether it can determine the presence of the pointer object on the screen (b=down/up),
- and whether it can accurately locate the x-y location of the pointer object.

Manually generated sequences of x and y locations are used as the test stimulus of the Object Location and the Action Interpretation program. The objective of the test is to verify the following:

- whether it can detect any clicking action,
- whether it the program can detect the dragging action,
- whether it can detect the single click action,
- and whether it can detect the double click action.

5.3. PC Demonstration Application

The PC Demo Application corresponds to the PC Driver Software functional block. It's purpose is to demonstrate that all tests mentioned in section 5 can be successfully performed using the LightTouch system. Pointer accuracy, minimum pointer size, minimum time between separate button clicks, and pointer movement tracking will be able to be performed on the demo application. The demonstration application will be DOS based, allowing a user to perform single click, double click, and window drag actions on the computer display, using a pseudo Windows-like application. The application will include a window which can be dragged, across the computer display, as well as a pull down menu. We also have a simple function which displays onscreen where the LightTouch system believes the current location of the pointer is, and what pointer action type, single click, double click, drag, or no action, is currently being performed.

6. Overall System Test Plan

Overall testing of the LightTouch system, measuring performance from a user's perspective, will be performed in the following areas:

Pointer Accuracy

We will test pointer accuracy in the following nine section of the screen, shown in Figure 6.1:

A1	B2	C2
A2	B2	C2
A3	B3	C3

Figure 6.1 Pointer Accuracy Test Zones

For each section, we will test pointer accuracy by pointing to a random point in each section, and measuring the distance of error between our physical pointer location and the computer interpreted pointer location. Distance of error in each case should be less than 20 pixels or 3cm, whichever is larger.

Minimum Pointer Size

We will test for LightTouch system recognition of pointers of roughly circular shape. Using our finger as a pointer will be tested, as well as a standard pencil and a human fist. We should be able to pick up a pointer of human fingertip size.

Button Click Time Delay

We will use a software timer on our PC to ensure that the maximum time delay between separately recognizable button clicks is 1 second.

Pointer Movement Tracking

The LightTouch system should be able to follow a moving pointer on the computer display in a continuous, button down state if the pointer moves at a maximum of 5 cm per second.

If our product passes all these tests, we will consider our functional requirements fulfilled.

7. Reference

[1] Dhananjay V. Gadre. *Programming the Parallel Port.* R&D Books, Lawrence: 1998.

[2] S.W. Awcock & R. Thomas. *Applied Image Processing.* McGraw Hill Incorporated, New York: 1996.

[3] Maxim 232 Multichannel RS-232 Driver.
http://209.1.238.250/arpdf/1798.pdf

[4] TSL1401, 128 x 1 Linear Sensor Array with Hold.
http://www.ti.com/sc/docs/folders/analog/tsl1401.html

[5] Motorola HC11 Microcontroller.
http://www.mcu.motsps.com/hc11/home.html

8. Appendix

Parts List

Item	Manufacturer	Part Number	Quantity
HC11 Microcontroller	Motorola	MCU-MC68HC811E2	1
Optical Sensor, Linear Array	Texas Instruments	TSL1401	4
RS232 Line Driver	Maxim	MAX232	1
Header Fin Connect Jumper			1
Push Button Switch			1
8.0 Mhz Clock Crystal			1
HC11 Socket			1
Parallel Cable and connectors			1
Serial Cable and connectors			1
AC Adapter			1
Resistor, 0.25w, 1%		10 KΩ	19
Resistor, 0.25w, 1%		10 MΩ	1
Resistor, 0.25w, 1%		4.7 KΩ	4
Resistor, 0.25w, 1%		1 KΩ	1
Capacitor, 0.25w, 1%		4.7 uF	1
Capacitor, 0.25w, 1%		1 uF	5
Capacitor, 0.25w, 1%		0.1 uF	1
Capacitor, 0.25w, 1%		18 pF	2