Stephanie Hsu
6.111 Project Proposal
3 November 2006

Writing Recognition

Overview

The project seeks to implement an algorithm that recognizes characters traced out by the movement of a mouse and prints the output onto a screen. The display will be divided into two sections: an area where the mouse traces the character and an area where the recognized character is displayed. A possible layout is shown in Figure 1.

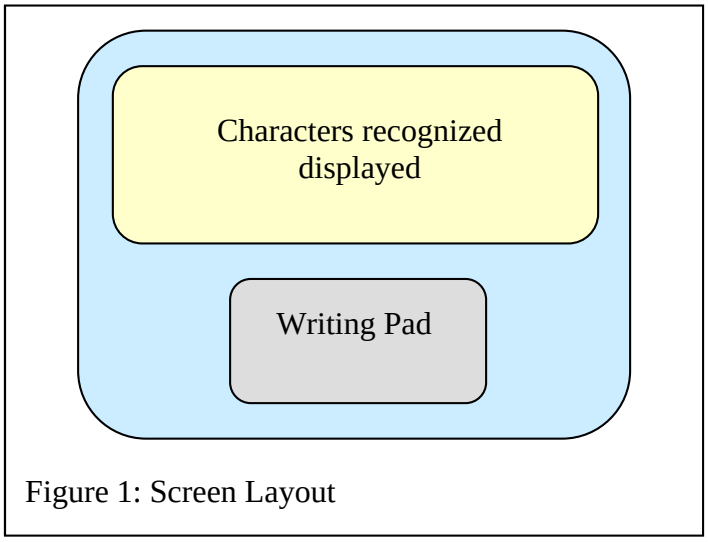

Figure 1: Screen Layout

The project will implement characters by storing the direction the mouse moves as it traces out a character. These directions are matched against a memory array that details the motion the mouse is expected to make when tracing out valid characters. Once a match is found between the input and the table, the character is displayed to the screen.

Modules

Mouse

The labkit interfaces with a PS/2 mouse. The mouse sends data (clocked on an internal mouse clock) to the labkit (mouse_data) that describe the movement of the mouse and any button clicks. Button clicks are encoded by single bits while movements in the x and y directions are 8-bit binary numbers. The directions of movement are controlled by direction bits and an overflow bit is asserted if the mouse is moved too quickly.

Since the data from the mouse only gives the change in x and y coordinates with respect to the previous measurement, the data needs to be translated into coordinates relative to the display. If reset is asserted, the mouse is repositioned on the middle of the

screen. The change in position data input from the mouse is then decoded to move the mouse around the screen. In addition, the left and right button signals are extracted from the data packet and output from the module. A block diagram detailing the input and output signals from the module are shown in Figure 2.

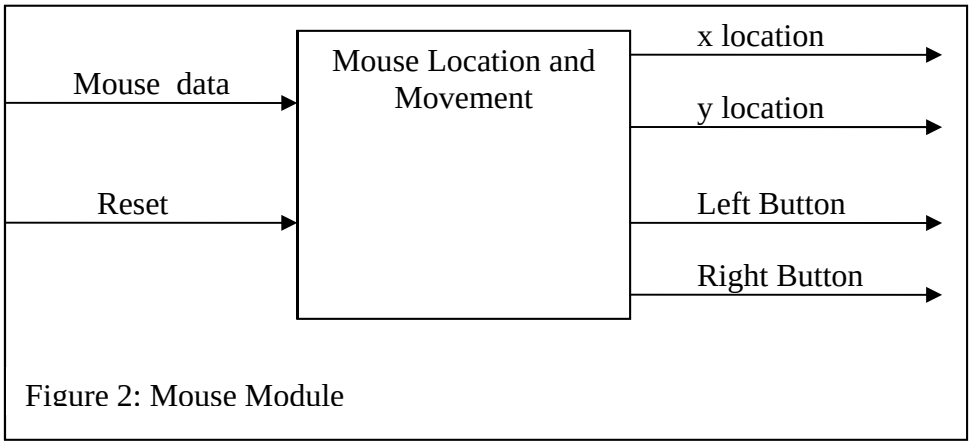

Figure 2: Mouse Module

Character Recognition

The decoded mouse input, along with the original mouse_data, is fed into the character recognition module. A memory component stores information about the movement of the mouse only if the position of the mouse falls within the area designated as the writing pad. Figure 3 outlines the flow of data within the module.

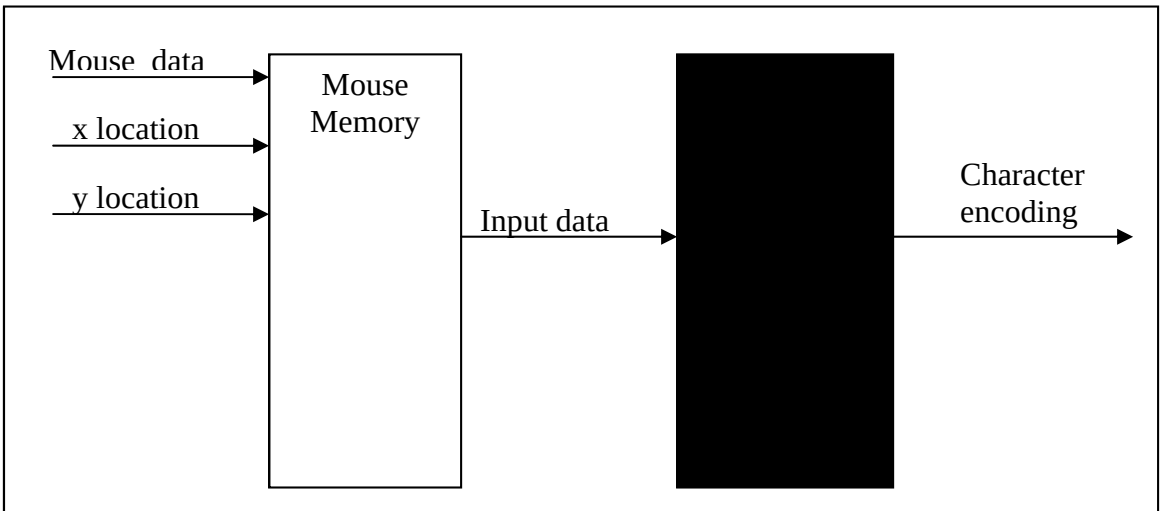

Figure 3: Character Recognition

The information stored about the movement of the mouse is matched against an existing memory of recognizable characters. A more detailed discussion of the strategy used to identify characters is located below. The module outputs information necessary to display the character found to be the best match to the drawn character to the display module.

Display Module

The display is responsible for refreshing the contents of the writing pad and the recognized character block. A memory block stores the contents of the screen and is interfaced with a module that generates the display on the computer screen.

The memory block is changed by modules that control the character recognized and the path of the mouse on the writing path (displayed while the mouse is moving). The position tracker module remembers the position that the current character is displayed and increments to the next position once the next character is recognized. A reset button clears the characters displayed and realigns the module to the left hand side of the display. The mouse movement module instantaneously displays the movement of the mouse. Inputs to the memory module from the mouse movement module only affect the display within the writing pad block.

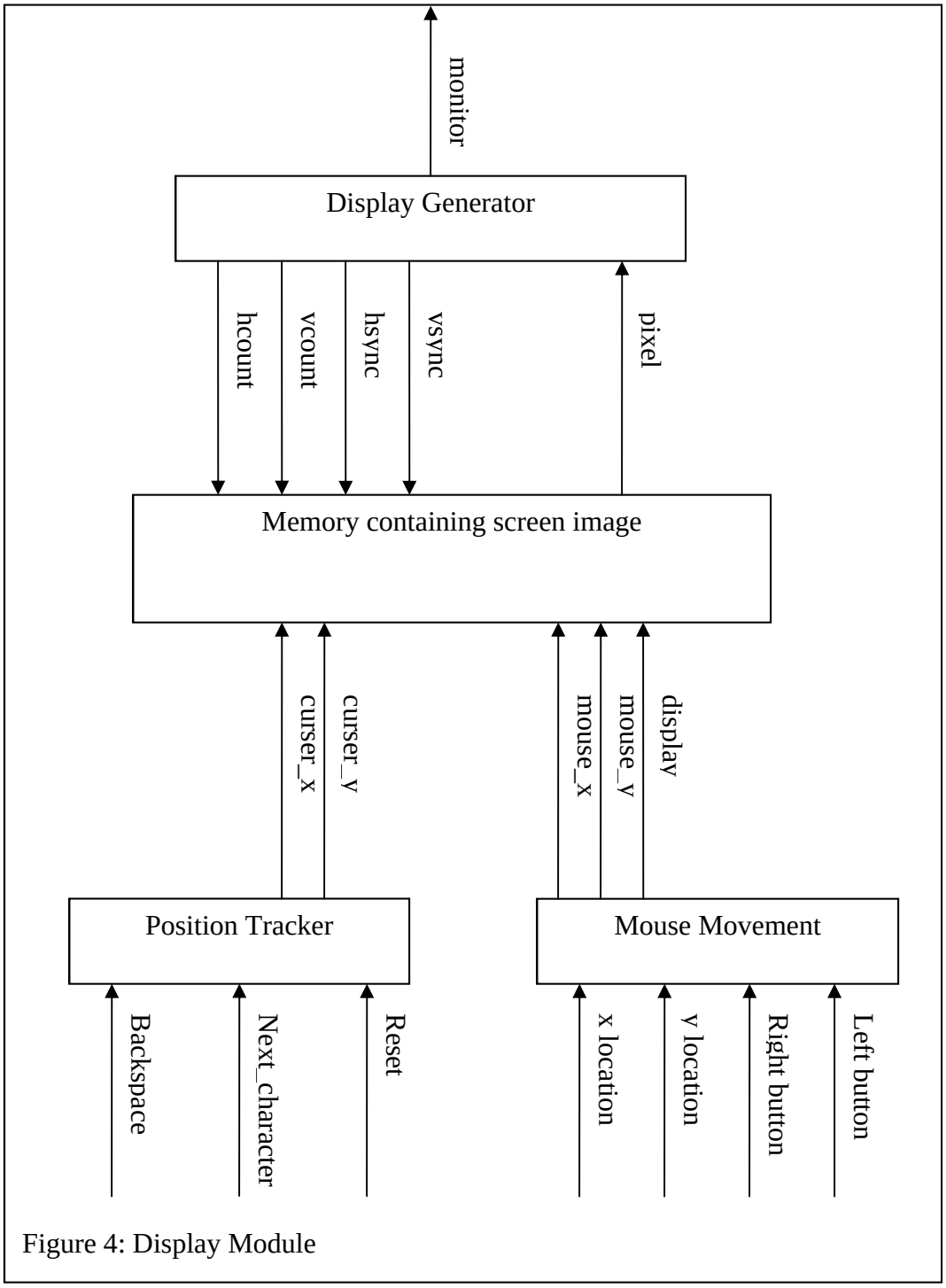

Figure 4: Display Module

Character Recognition Strategy

The process of recognizing characters is based on the strategy used by the Palm Graffiti software. The software expects that every valid character is drawn using a predetermined set of strokes and recognizes character based on the direction the mouse is moving.

The implemented strategy implements the Graffiti software strategy by dividing the writing pad into eight blocks. Within each block, the direction that the mouse moves is categorized into eight directions.

The software remembers which direction the mouse is moving in each block and searches down a tree to find a possible valid character match. Each node of the tree leads to another node based on the one of eight possible directions the mouse is determined to be moving toward. In addition, the tree also begins the search based on which block in the writing pad the mouse was located when starting to sketch the character.

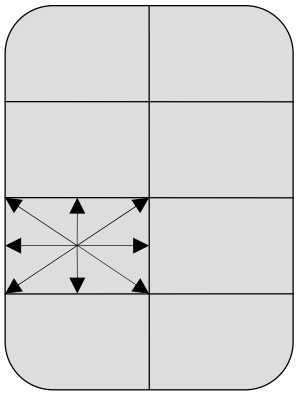

Figure 4: Writing Pad

Other possible strategies include storing the direction the mouse is moving at small time intervals (without begin constrained to the eight individual boxes) and comparing the overall direction to the tree. In addition, a character can be recognized by comparing the positions on the writing pad that the mouse touches with expected positions that a character should touch.