Class: EE3501

Assignment: Lab 4

Professor: Craig A. Chin

Student: Goran Novakovic

Lab Results:

(Code Start)

#include "mbed.h"

DigitalOut led1(LED1);

DigitalOut ROW1(PA\_6);

DigitalOut ROW2(PA\_7);

DigitalOut ROW3(PB\_6);

DigitalOut ROW4(PC\_7);

DigitalIn COL1(PA\_9, PullUp);

DigitalIn COL2(PA\_8, PullUp);

DigitalIn COL3(PB\_10, PullUp);

DigitalIn COL4(PB\_4, PullUp); //Connecting the inputs and outputs of the keypad to board

int const numRows = 4;

int const numCols = 4;

bool cols[4];

char keys[numRows][numCols] = {

{'1','2','3','A'},

{'4','5','6','B'},

{'7','8','9','C'},

{'\*','0','#','D'},

}; //Grid of characters for output when system is running the code following keypad

int colPress = 0;

void readCols()

{

cols[0] = COL1.read();

cols[1] = COL2.read();

cols[2] = COL3.read();

cols[3] = COL4.read();

}

bool checkPress()

{

for(int i=0; i<numCols; i++){

if(!cols[i]){

colPress = i;

return true;

}

}

return false;

}

char scan()

{

int currentRow = 0;

while(1){

ROW1 = 1;

ROW2 = 1;

ROW3 = 1;

ROW4 = 1;

wait\_ms(10);

currentRow = 1;

ROW1 = 0;

wait\_ms(5);

readCols();

if(checkPress()){

return keys[currentRow-1][colPress];

}

wait\_ms(5);

ROW1 = 1;

currentRow = 2;

ROW2 = 0;

wait\_ms(5);

readCols();

if(checkPress()){

return keys[currentRow-1][colPress];

}

wait\_ms(5);

ROW2 = 1;

currentRow = 3;

ROW3 = 0;

wait\_ms(5);

readCols();

if(checkPress()){

return keys[currentRow-1][colPress];

}

wait\_ms(5);

ROW3 = 1;

currentRow = 4;

ROW4 = 0;

wait\_ms(5);

readCols();

if(checkPress()){

return keys[currentRow-1][colPress];

}

wait\_ms(5);

ROW4 = 1;

}

} //All above is the system of how the inputs of button pressing are read

int main()

{

char input, previousInput;

char match[5] = {'#','1','2','0','2'}; //Backwards because that’s how the input will be read

char priorCpy[5]; //Works as a password as only looking at last 5 inputs.

char priorInput[5]; //Char functions here are the five input password needed

int yesCount = 0;

printf("Enter code.\n\r");

while(1){

input = scan();

if(input != previousInput){

for(int i=0; i<5; i++){

priorCpy[i] = priorInput[i];

printf("%c", priorCpy[i]);

}

printf("\r\n");

priorInput[0] = input;

for(int i=1; i<5; i++){

priorInput[i] = priorCpy[i-1];

}

}

yesCount = 0;

for(int i=0; i<5; i++){

if(priorInput[i] == match[i]){

yesCount++; //Essentially ‘going’ through each character checking if its part of combo

}

}

if(yesCount == 5){

led1 = 1;

}

if(input == '\*'){

led1 = 0;

}

previousInput = input;

printf("input = %c\r\n", input); //Buggy with the button constantly pressed multiple outputs

}

}

(Code End)

A picture containing shape

Description automatically generated

Figure 1: Putty output and password #2021

Discussion:

1) Refer to the webpage below and state the 4 input pin modes of a DigitalIn object?

PullUp – A resistor is connected to high.

PullDown – A resistor is connected to low.

PullNone – No connection to either pullup or pulldown

OpenDrain – A transistor connects to low

2) Describe the two methods for setting the value of a DigitalOut object.

DigitalOut (PinName pin): gpio() Which creates a DigitalOut object connected to a specific pin.

DigitalOut (PinName pin, int value): gpio() Which creates a DigitalOut object connected to a specific pin and set with an output value of 0 or 1.

3) Describe how the code is able to detect individual keys when in a 4x4 matrix.

The code can detect individual keys by utilizing a matrix style algorithm which is basically deciding the point at which two signals cross paths. It can be thought of like a Cartesian plane, if you know the x and y value, then you know the point in 2D space.