#include <WiFi.h>

#include <WiFiClient.h>

#include <WiFiServer.h>

#include <WiFiUdp.h>

#include <SPI.h>

#include <EEPROM.h>

#include <common.h>

#include <Keypad.h>

const byte ROWS = 4; //four rows

const byte COLS = 4; //four columns

//defining the symbols on the buttons of the keypads.

char keys[ROWS][COLS] = {

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

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

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

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

};

byte rowPins[ROWS] = {9, 8, 7, 6}; //{11, 10, 9, 8};//pins connected to the row pinouts of the keypad

byte colPins[COLS] = {5, 4, 3, 2}; //{7, 6, 5, 4}; //pins connected to the column pinouts of the keypad

//initialize an instance of class NewKeypad

Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS);

//MODE\_CHANGE e\_mode\_change = INTIAL;

MODE\_TYPE current\_mode = NOTHING;

MODE\_TYPE prev\_mode = NOTHING;

TANK\_STAUTS tank\_mode = TANK\_STAUTS\_OK;

unsigned int integerValue = 0;

float float\_dispaly\_qty = 0.0;

float float\_tank\_qty = 210.00;

//struct MACHINE\_DESC machine\_desc;

float f\_mode\_1 = 0;

float f\_mode\_2 = 0;

void setup() {

Serial.begin(9600);

pinMode(led, OUTPUT);

f\_mode\_1 = 900; // read form sd card

f\_mode\_2 = 100; // read form sd card

// if(1){

if (readTankStatus(20.0)) { /// DANGerous code start

//displayMessage("you can press A / B / C / D to continue");

}

else {

current\_mode = FILLING\_REQ;

tank\_mode = EMPTY\_TANK;

} /// DANGerous code end

// Serial.println(" <<<<<<<<<<<<<<<<<<<< you can press A / B / C / D to continue >>>>>>>>>>>> ");

// Add an event listener for this keypad

// keypad.setHoldTime(2\*1000);

keypad.addEventListener(keypadEvent);

}

void loop() {

//scan the pins of the keypad

char key = keypad.getKey();

//blink the LED

digitalWrite(led, HIGH);

delay(100);

digitalWrite(led, LOW);

delay(100);

//displayMessage(" TANK level sensor ");

//delay(2\*1000);

if (tank\_mode == TANK\_STAUTS\_OK) { /// DANGerous code start

if (current\_mode != NOTHING) {

// displayMessage("LOOP 2 PASSED");

getKeyPadInput();

}

}

else if (tank\_mode == EMPTY\_TANK) {

displayMessage("FILLING\_REQ IS REQUIRED ");

} /// DANGerous code end

//if (mode\_change == true) showModeValue();

//

// 1. read tank statuS , 2. read abort status to stop filling 3.

//

}

// Taking care of some special events.

void keypadEvent(KeypadEvent key) {

// Serial.println("keypadEvent()---> called ");

if (tank\_mode == TANK\_STAUTS\_OK) {

switch (keypad.getState()) {

case PRESSED:

// mode\_key = key;

if (key == 'A') {

//when 'A' is pressed allow the user to input a key.

current\_mode = REGULAR\_MODE;

}

else if (key == 'B') {

current\_mode = FAST\_MODE\_1;

}

else if (key == 'C') {

current\_mode = FAST\_MODE\_2;

}

else if (key == '#') {

current\_mode = CONTINUE;

}

else if (key == '\*') {

current\_mode = CANCEL;

}

break;

} // switch end

}

}

void getKeyPadInput() {

integerValue = 0;

char incomingByte;

if (current\_mode == REGULAR\_MODE) {

integerValue = 0;

integerValue = getKeypadInput();

float\_dispaly\_qty = (float)integerValue / 1000.000;

// displayMessage("The Entered value is ");

// displayFloatMessage(float\_dispaly\_qty);

// Serial.println(" -- MLS PRESS # to continue [or] prees \* to cancel ");

//fillBottle(); // 06/10/2016

current\_mode = NOTHING;

prev\_mode = current\_mode;

} // REGULAR\_MODE END

else if (current\_mode == FAST\_MODE\_1) {

// Serial.println(" <<<<<<<<<<<<<<<<<<<< FAST\_MODE\_1 >>>>>>>>>>>> ");

current\_mode = NOTHING;

prev\_mode = current\_mode;

float\_dispaly\_qty = f\_mode\_1 / 1000;

// Serial.println(" <<<<<<<<<<<<<<<<<<<< PRESS # to continue [or] prees \* to cancel >>>>>>>>>>>> ");

}

else if (current\_mode == FAST\_MODE\_2) {

// Serial.println(" <<<<<<<<<<<<<<<<<<<< FAST\_MODE\_2 >>>>>>>>>>>> ");

current\_mode = NOTHING;

prev\_mode = current\_mode;

float\_dispaly\_qty = f\_mode\_2 / 1000;

// Serial.println(" <<<<<<<<<<<<<<<<<<<< \n PRESS # to continue [or] prees \* to cancel >>>>>>>>>>>> ");

}

else if (current\_mode == FAST\_MODE\_3) {

// Serial.println("\n FAST\_MODE\_3 \n");

current\_mode = NOTHING;

prev\_mode = current\_mode;

}

else if (current\_mode == CONTINUE) {

if (float\_dispaly\_qty != 0.00) {

if (readTankStatus(float\_dispaly\_qty))

fillBottle(float\_dispaly\_qty);

}

else {

// Serial.println(" <<<<<<<<<<<<<<<<<<<< Please enter more than zero value >>>>>>>>>>>> ");

}

if (tank\_mode != EMPTY\_TANK)

// Serial.println("\n PREES # FOR CONTINUE \n [or] \* FOR CANCEL ");

//incomingByte = keypad.waitForKey();

current\_mode = NOTHING;

}

else if (current\_mode == CANCEL) {

// Serial.println("\n CANCEL \n");

current\_mode = NOTHING;

// digitalWrite(led, HIGH);

// Serial.println(" <<<<<<<<<<<<<<<<<<<< you can press A / B / C / D to continue >>>>>>>>>>>> ");

// incomingByte = keypad.waitForKey();

}

}

void fillBottle(float f\_qty)

{

if (f\_qty != 0.00) {

for (int i = 0; i < 10 ; i++) {

Serial.print("\*\*\*\* ");

// Serial.print(f\_qty);

// Serial.print("\*\*\*\* ");

// Serial.print(float\_tank\_qty);

float\_tank\_qty = float\_tank\_qty - 10.00;

// Serial.print("\n");

delay(500);

}

// displayMessage("Up dateed SD Card");

}

}

bool readTankStatus(double float\_dispaly\_qty)

{

bool bool\_tank\_status = false;

int int\_dispaly = int(float\_dispaly\_qty);

int int\_tank\_qty = int(float\_tank\_qty); // WE GET INFO FROM SENSOR

if (int\_tank\_qty > (int\_dispaly + 10) ) {

bool\_tank\_status = true;

}

else {

// displayMessage(" TANK IS EMPTY PLEASE FILL THE TANK BEFORE THIS OPERATION ");

// current\_mode = FILLING\_REQ;

tank\_mode = EMPTY\_TANK;

bool\_tank\_status = false;

}

return bool\_tank\_status;

}

void displayMessage(char \*strMessage) {

// Serial.print("<<<<< ");

// Serial.print(strMessage);

// Serial.println(" >>>>>");

}

void displayFloatMessage(float f\_val) {

//erial.print("<<<<< ");

// Serial.print(f\_val);

//Serial.println(" >>>>>");

}

int getKeypadInput()

{

char incomingByte;

int integerValue = 0;

// TERG : REGULAR\_MODE START

// displayMessage(" please enter the value below 10000 ML or 10 ltrs end with 'D' ");

while (1) { // force into a loop until 'n' is received

incomingByte = keypad.waitForKey();

//

// look for enter and zero checking

//

if (incomingByte == 'D') { // TERG : look for enter and zero checking

if (integerValue != 0)

break;

else {

displayMessage("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Error :: Please enter more tham 0 value \*\*\*\*\*\*\*\*\*\*\*\* ");

}

} // TERG : look for enter and zero checking

//

// CHECK FOR NUMERICS ONLY ,

//

else if (incomingByte != 'A' && incomingByte != 'B' && incomingByte != 'C' && incomingByte != '\*' && incomingByte != '#') {

// Serial.print(incomingByte);

if (incomingByte == -1) continue; // if no characters are in the buffer read() returns -1

integerValue \*= 10; // shift left 1 decimal place

// convert ASCII to integer, add, and shift left 1 decimal place

integerValue = ((incomingByte - 48) + integerValue);

if (integerValue > 10000) {

displayMessage(" Value Exceeded .... please enter the value below 10000 ML or 10 ltrs ");

integerValue = 0;

}

}

} // WHILE (1) END

return integerValue;

}

Int incrementKey(int modevalue , int incValue){

Return (modevalue+ incValue);

}

int decrementKey(int modevalue , int decValue){

Return (modevalue+ decValue);

}

// Taking care of some special events.

void keypadEvent(KeypadEvent key) {

// Serial.println("keypadEvent()---> called ");

if (tank\_mode == TANK\_STAUTS\_OK) {

switch (keypad.getState()) {

case PRESSED:

// mode\_key = key;

if (key == '+') {

//when 'A' is pressed allow the user to input a key.

current\_mode = INC\_MODE;

reqValue = incrementKey(display\_value, inc\_dec\_value);

LCD.display(reqValue);

}

else if (key == '-') {

current\_mode = DEC\_MODE;

reqValue = decrementKey (display\_value, inc\_dec\_value);

LCD.display(reqValue);

}

else if (key == '1') { // 10ML

current\_mode = FAST\_MODE\_1;

modevalue = 10;

inc\_dec\_value = 10;

}

else if (key == '2') { // 100ml

current\_mode = FAST\_MODE\_2;

modevalue = 100;

inc\_dec\_value = 10;

}

else if (key == '3') {// 500ml

current\_mode = FAST\_MODE\_3;

modevalue = 500;

inc\_dec\_value = 100;

}

else if (key == 'ENTER') {//

current\_mode = CONTINUE;

IF(READTANK\_STATUS())

FILL\_BOTTLE();

}

break;

} // switch end

}

}

void fillBottle(float f\_qty)

{

reqQty = false;

While(reqQty == true){

If(reqQty >= reqValue){

.. Stop solenoidvalve

.. update file

..

}

}