```
//Libraries
#include <LiquidCrystal.h>        //LCD library
#include <SoftwareSerial.h>       //SoftwareSerial library
#include <OneWire.h>
#include <DallasTemperature.h>
#include <SPI.h>                  //SPI library for SD card
#include <SD.h>                   //SD card library

//Serial ports
#define orprx 2                         //define what pin orp rx is going to be
#define orptx 3                         //define what pin orp Tx is going to be
SoftwareSerial orpserial(orprx, orptx); //define the ORP soft serial port
#define phrx 14                         //define what pin pH rx is going to be
#define phtx 15                         //define what pin pH Tx is going to be
SoftwareSerial phserial(phrx, phtx);    //define the pH soft serial port

//Temperature probe setup
#define ONE_WIRE_BUS 19                  // Data wire is plugged into pin 19 on the Arduino
OneWire oneWire(ONE_WIRE_BUS);           // Setup a oneWire instance to communicate with any OneWire devices
DallasTemperature sensors(&oneWire);     // Pass our oneWire reference to Dallas Temperature.
DeviceAddress insideThermometer = { 0x28, 0xB4, 0x6B, 0xC8, 0x04, 0x00, 0x00, 0x1F };     // Assign the addresses of your 1-Wire temp sensors.

//define ORP variables
char orp_data[20];                    //20 byte character array to hold ORP data
char orp_computerdata[20];            //20 byte character array to hold incoming data from a pc
byte orp_received_from_computer=0;    //we need to know how many character have been received.
byte orp_received_from_sensor=0;      //we need to know how many character have been received.
byte orp_startup=0;                   //used to make sure the arduino takes over control of the ORP Circuit properly.
float ORP=0;                          //used to hold a floating point number that is the ORP
byte orp_string_received=0;           //used to identify when we have received a string from the ORP circuit

//define pH variables
char ph_data[20];                    //20 byte character array to hold incoming pH
char ph_computerdata[20];            //20 byte character array to hold incoming data from a pc
//byte pc_debug=0;                   //if you would like to debug the pH Circuit through the serial monitor(pc/mac/other). if not set this to 0.
byte ph_received_from_computer=0;    //we need to know how many characters have been received from computer
byte ph_received_from_sensor=0;      //we need to know how many characters have been received from pH sensor
byte ph_startup=0;                   //used to make sure the arduino takes over control of the pH Circuit properly.
float ph=0;                          //used to hold a floating point number that is the pH.
byte ph_string_received=0;           //used to identify when we have received a string from the pH circuit.

//LCD set up
LiquidCrystal lcd(8, 9, 4, 5, 6, 7); // select the pins used on the LCD panel

void setup(){
     Serial.begin(38400);         //enable the hardware serial port
     orpserial.begin(38400);     //enable the software serial port
     phserial.begin(38400);      //enable the software serial port
     sensors.begin();            //start up temp probe library
     sensors.setResolution(insideThermometer, 10);       // set the temp probe resolution to 10 bit
     lcd.begin(16, 2);           // start the lcd library
     SD.begin(16);
     pinMode(10, OUTPUT);
     }


void loop() {
 sensors.requestTemperatures();         //read Temp probe
  printTemperature(insideThermometer);

  orpserial.listen();
  delay(100);
  if(orpserial.available() > 0){           //if we see that the ORP Circuit has sent a character.
     orp_received_from_sensor=orpserial.readBytesUntil(13,orp_data,20); //we read the data sent from ORP Circuit untill we see a <CR>. We also count how many character have been recived.
     orp_data[orp_received_from_sensor]=0; //we add a 0 to the spot in the array just after the last character we recived. This will stop us from transmiting incorrect data that may have been left in the buffer.
     orp_string_received=1;                //a flag used when the arduino is controlling the ORP Circuit to let us know that a complete string has been received.
  }
  phserial.listen();
  delay(100);
  if(phserial.available() > 0){          //if we see that the pH Circuit has sent a character.
     ph_received_from_sensor=phserial.readBytesUntil(13,ph_data,20); //we read the data sent from ph Circuit untill we see a <CR>. We also count how many character have been recived.
     ph_data[ph_received_from_sensor]=0; //we add a 0 to the spot in the array just after the last character we recived. This will stop us from transmiting incorrect data that may have been left in the buffer.
     ph_string_received=1;               //a flag used when the arduino is controlling the pH Circuit to let us know
     that a complete string has been received.
  }
}
void printTemperature(DeviceAddress deviceAddress)
{
    int decPlaces = 0;     // set temp decimal places to 0
  float tempC = sensors.getTempC(deviceAddress);
  if (tempC == -127.00) {
    lcd.print("Error getting temperature");
  } else {
     lcd.setCursor(0,0);   //set position on lcd for pH
     lcd.print("pH:");
     lcd.print(ph, 1);     //send pH to lcd
     lcd.setCursor(7,0);   //set position on lcd for ORP
     lcd.print("ORP:");
     lcd.print(ORP, 0);    //send ORP to lcd
     lcd.setCursor(0,1);   //set position on lcd for Temp
     lcd.print("Temp:");
     lcd.print("C ");
     lcd.print(tempC,decPlaces);     //display Temp in celsius
     lcd.print(" F ");
     lcd.print(DallasTemperature::toFahrenheit(tempC),decPlaces);  //convert celsius to farenheit
     delay(10000);          //we will take a reading ever 10000ms


orpserial.print("R\r");                     //send it the command to take a single reading.
   if(orp_string_received==1){              //did we get data back from the ORP Circuit?
     ORP=atof(orp_data);                    //convert orp_data string to ORP float
     if(ORP>800){Serial.println("high\r");} //This is the proof that it has been converted into a string.
     if(ORP<800){Serial.println("low\r");}  //This is the proof that it has been converted into a string.
     orp_string_received=0;}                //reset the string received flag.

phserial.print("R\r");                      //send it the command to take a single reading.
   if(ph_string_received==1){               //did we get data back from the ph Circuit?
     ph=atof(ph_data);                      //convert ph_data string to ph float
     if(ph>=7.5){Serial.println("high\r");} //This is the proof that it has been converted into a string.
     if(ph<7.5){Serial.println("low\r");}   //This is the proof that it has been converted into a string.
     ph_string_received=0;}                 //reset the string received flag.
  }

 long currentTime = millis();                                // Get the current time in ms (time since program start)
 File dataFile = SD.open("datalog.txt", FILE_WRITE);         //open the file
  if (dataFile) {                                            // if the file is available, write to it:
      dataFile.println(currentTime);                         // logs the time in milliseconds since the program started
      dataFile.print(",");                                   //inserts a comma
      dataFile.println(ph);                                  //logs the pH
      dataFile.print(",");                                   //inserts a comma
      dataFile.println(ORP);                                 //logs the ORP
      dataFile.print(",");                                   //inserts a comma
      dataFile.println(tempC);                               //logs the temperature in degrees C
      dataFile.print("\r");                                  //inserts a return character
      dataFile.close();
}
}
```