ISSN: 2634 - 8845

Journal of Earth and Environmental Sciences Research

SCIENTIFIC Research and Community

Review Article

Open Access

Circuit Design and Embedded Software Development of Automatic Water Seepage Meter

Junjie YUAN and Hui ZHANG*

School of Energy and Environmental Engineering, University of Science and Technology Beijing, Beijing, China

ABSTRACT

Permeability meter plays an important role in the measurement of road seepage performance. The data deviation is easy to produce because of incorrect reading by using the manual seepage meter. Thereby, the application program based on Windows CE operating system is developed, which includes test process, data processing and battery monitoring. And the manual control, automatic delay and industry standard operation processes can be achieved by adopting the timing and event-driven modes, furthermore, it has the function of calculating standard deviation, mean value and coefficient of variation, checking out the existing data. And the recorded data can be automatically appended to the list. The software which uses the recursive median algorithm to filter the original data can monitor the dynamic change of liquid level synchronously, and can also measure the initial and final liquid level height like the manual seepage meter. It can provide a smooth curve of the seepage process. Combined with observation and data analysis, it is found that the process curve is obviously related to the phenomena of water seepage, lateral infiltration, water leakage and bubble burst, so these phenomena of water seepage process of road can be recognized through algorithm development.

***Corresponding author**

Hui ZHANG, School of Energy and Environmental Engineering, University of Science and Technology Beijing, Beijing, China.

Received: June 27, 2023; **Accepted:** July 03, 2023; **Published:** July 10, 2023

Keywords: Liquid level pressure; Water permeability coefficient; Water penetration process; Automatic water seepage meter; Embedded software for water seepage meter

Introduction

Pavement Seepage Meter (hereinafter referred to as Seepage Meter) is an important test device for measuring pavement seepage coefficient and characterizing pavement seepage performance. The accuracy of relevant measurement results is related to the service life of the road and traffic safety. With the transition of China's highway industry from the construction period to the maintenance period, the rescue-based passive maintenance will change to the preventive maintenance. However, the traditional manual operation of the water seepage instrument lacks the sensing monitoring function required for the construction of the Internet of Things, and the collected large amounts of data are difficult to be applied by AI equipment, which cannot provide data support for highway monitoring, disease warning and maintenance decision-making in time. As the terminal node of the Internet of Things, the water seepage meter not only needs to have the ability to measure the seepage coefficient, but also needs to carry out preliminary analysis and judgment of the seepage process, and has the ability of intelligent detection.

At present, the water seepage meter uses the operation method of combining meter pinch timing with visual reading scale, and human factors have great influence on the measurement accuracy. Through the improvement of the water seepage instrument, the automatic detection and timing of 100 mL and 500 mL liquid levels can be realized by using photoelectric liquid level switch, single chip microcomputer controller and electromagnetic valve, and the automatic measurement of water seepage time can be completed to improve the accuracy of measurement [1,2]. For the asphalt pavement with large voids, the seepage coefficient is required to be greater than 3600 mL/min, the double layer pavement is up to 6000~8000 mL/min, and the liquid level drop time consumption is only 3~5 s, and the personnel observation error is large. The standard deviation measured by the electronic osmometer with weighing sensor, solenoid valve and microprocessor is one tenth of the manual measurement results, and the response time reaches 0.01 s [3,4]. The American Association for Materials and Testing has published a water seepage device that uses up and down float switches to trigger clocks to automatically record the corresponding time of two liquid levels. In order to improve the intelligent analysis and processing ability of data, Luo Wen-ting et al developed a water permeability coefficient measurement system combining four-core chip, 12-bit AD module, eType resistance type liquid level sensor and solenoid valve [5]. The Kalman filter was used to eliminate the influence of liquid level fluctuation, and the resolution reached 1 mm. The liquid level data per second could be displayed in real time, and the water permeability rate and evaluation results could be automatically calculated. The system was connected to the mobile phone through Bluetooth, and the on-board serial port module was used to communicate with the computer. The pavement water seepage meter developed by the Research Institute of Highway Science of the Ministry of Transport adopts machine vision technology to identify the liquid level in a non-contact way, so as to achieve the synchronization of volume measurement and timing, without any structural transformation of the existing water seepage device [6]. Chen Jun et al studied

the directional difference characteristics of water permeability of asphalt mixture and the law between connected pores by combining CT scanning and image processing methods [7].

A large number of studies have shown that the test process data can reflect the changes of structural and functional parameters such as infiltration, lateral infiltration, connected porosity, segregation, compaction, mineral gradation, freeze-thaw cycle performance and soil slope stability [8-15]. However, the photoelectric liquid level switch and the float switch are all for the initial and final state detection, and the water leakage, defoaming and side leakage phenomena in the test process cannot be identified. Weighing sensors and CT are limited to laboratory studies. Machine vision technology is significantly affected by many factors such as target height, magnification, focal length, and is expensive, so it is difficult to be applied to engineering practice. Only eType liquid level sensor can monitor the liquid level change in real time, but it is fixed by tape, and the position is easy to change, and the measurement consistency cannot be guaranteed.

Using the rich input and output control ability, high efficiency calculation ability and data acquisition ability of embedded Window CE system, taking digital pressure sensor as liquid level sensing element, combined with serial port and network communication function, a set of automatic measuring instrument for seepage coefficient based on embedded operating system is designed and developed.

Automatic Permeation Process

The structure size of the automatic water seepage instrument is made according to JJG104-2015 standard. When the liquid level changes 1 mm, the water volume changes about 2.00 mL and the pressure changes about 10 Pa. The digital pressure sensor PI with RS-485 semi-duplex communication is installed in the pipeline below the water tank through the G1/4 external thread. The original manual switch valve was replaced by a VX232GZ1E type solenoid valve with an interface diameter of G1/2 and a orifice diameter of 10 mm, which was powered by DC12V.

Figure 1: shows the six process steps decomposed according to industry standard JJG104-2015.

1. Water injection: After sealing the water meter through putty and road surface, the solenoid valve is closed, and the water is filled in the water tank.
2. Filling Open the solenoid valve, water filling to the gap between the road and the water seepage meter, the gas is discharged, and accompanied by water seepage process.
3. Calibration: close the solenoid valve, fill the water to 0 mL scale line, and calibrate the pressure measured by the pressure sensor through the upper computer software.
4. Pre-seepage: When the electromagnetic valve is opened, the water begins to infiltrate into the pavement. With the continuous infiltration of the pavement, the liquid level decreases at a certain rate.
5. Start: when the water level height reaches 100 mL scale line, the automatic seepage meter identifies the water level height through the pressure value feedback, and accurately records the starting time.

6A End: Water continuously seeps into the pavement. When water leakage, overflow, blistering, defoaming or other phenomena occur, the pressure monitoring system can sense the corresponding pressure changes in real time, and record the process data completely. When the water level cannot reach 500 mL scale in 3 min, the corresponding water level height is recorded.
6B End: When the water level reaches 500mL scale line, if the time is not more than 3 min, then end of the time.

After the test, the corresponding seepage coefficient is calculated according to the time difference and pressure difference between the starting time and the final time.

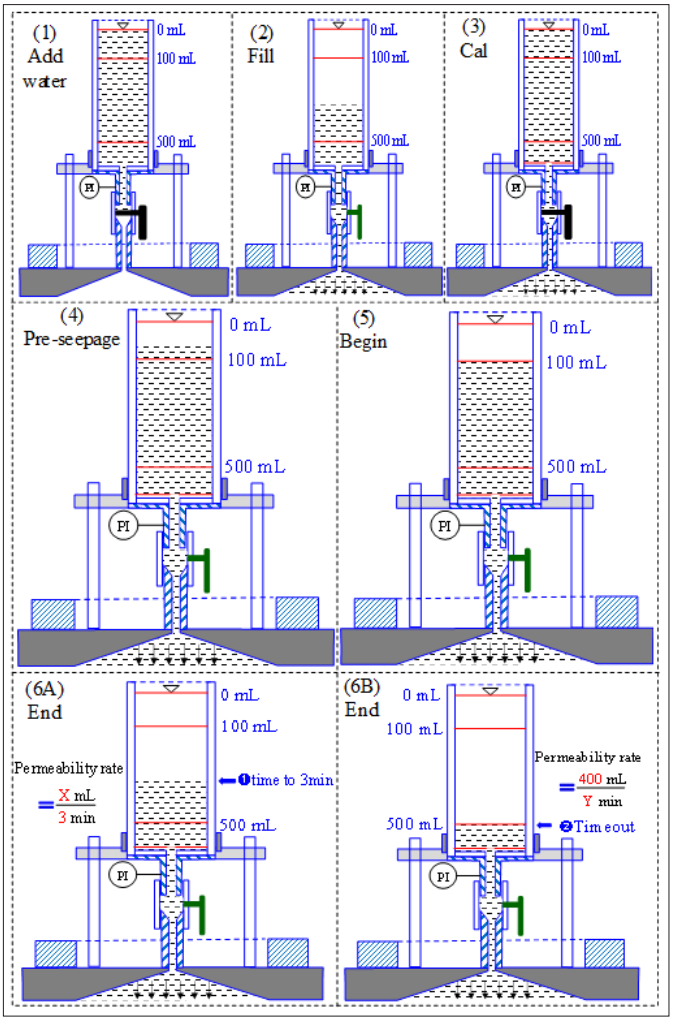

Figure 1: Schematic Diagram of the Process Steps

Control System Design

The control system consists of LJD-eWinV5-ST4 touch screen, circuit board, digital pressure sensor, solenoid valve and lithium battery pack. The touch screen provides man-machine interface, as shown in Figure: 2, through its RS-485 communication interface, AD interface, input and output I/O interface connected with the circuit board. The circuit board connects the various external devices together for photoelectric isolation and power amplification, driving the solenoid valve, as shown in Figure: 3.

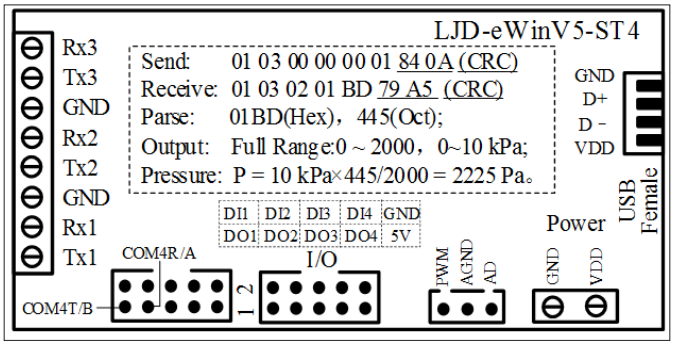

Figure 2: Interface Diagram of the Touch Screen with External Components

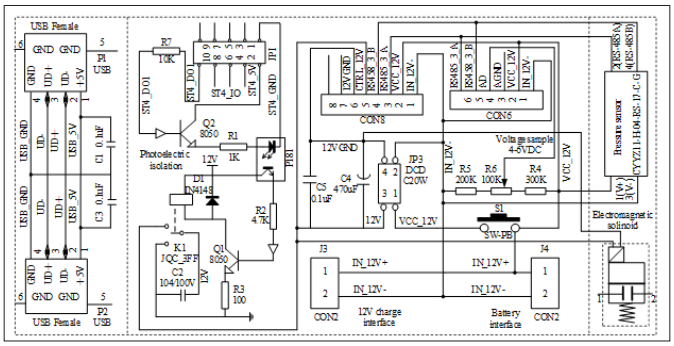

Figure 3: Circuit Diagram of Control System for Automatic Water Seepage Meter

Real-Time Liquid Level Sensing

The measuring range of the pressure sensor is 0~10 kPa, and the digital output corresponds to 0~2000. Each digital corresponds to the identifiable pressure of 5 Pa, which is equivalent to the height of 0.5 mm water column. According to the inner diameter of the water cylinder, the volume can be calculated to be about 1 mL. The program can send instructions to the sensor at regular intervals:
01 03 00 00 00 01 84 0A
It represents 00 01 words (equivalent to 2 bytes) at the beginning of 00 00 address of the address pressure sensor 01 (03 function code), and the cyclic redundancy check code is 84 0A (low bytes are ahead, and high bytes are behind).
The receiving command format is as follows:
01 03 02 01 BD 79 A5
Indicates reading the address pressure sensor No. 01 (03 function code) to get 02 bytes of data, the value is 01 BD, cyclic redundancy check code is 79 A5 (low bytes in front, high bytes in back).
The obtained data 01 BD is hexadecimal and then converted into decimal is:
$0\times16^3+1\times16^2+11(B)\times16^1+13(D)\times16^0=445$.
According to the corresponding relationship between the output range of digital
pressure sensor 0~2000 and the pressure range of 0~10 kPa, the corresponding pressure can be obtained as follows:
0 + 445 × (10000 - 0) / (2000 - 0) = 2225 Pa
The refresh frequency of the pressure sensor is 5 Hz, and according to the Shannon theorem, the reading frequency is more than twice that of 5 Hz. Therefore, it is appropriate to set the reading instruction time interval of 100 ms in the touch screen software.

Optoelectronic Isolation Control

The touch screen has four digital output DO interfaces, which connect the JP1 interface of the circuit board with the I/O interface of the touch screen. When the DO1 is high, the triode potential is increased through the ST4_DO1 pin on the circuit board, so that the P181 photocoupler glows, the constant open contact point of the JQC_3FF relay is closed, the 12V power supply is connected, and the external solenoid valve is powered. When DO1 is low, the optocoupler lights are out, the relays are often disconnected, and the solenoid valves are disconnected. Due to the coil inside the solenoid valve, inductance will be produced in the process of power-on and power-off, which affects the digital signal of the pressure sensor. In order to reduce the stimulation of the electromagnetic pulse, the isolation module DCD is designed. The solenoid valve is directly supplied by the external battery, and the voltage after DCD isolation is supplied by the pressure sensor and the touch screen. In this way, the digital circuit and the analog circuit are separated to reduce the interference on the data acquisition signal.

Battery Voltage Acquisition

The whole system is powered by three 26800 Li batteries, the voltage range is 7.8~12.9V, 7000mA•h. The battery consists of a closed loop through a 300 kΩ current limiting resistor, a 200 kΩ voltage pull-up resistor and a 100 kΩ variable resistor. The anode of the battery is connected with the resistance of 200kΩ, and the output node of the variable resistance R6 is connected with the AD pin of the touch screen to realize the acquisition of the analog voltage. Touch screen A/D is 12 bit, its range is 0~4096. Through the actual measurement, the corresponding linear relationship between the touch screen AD value and the voltage value is obtained as follows:
AD value = 263 × voltage + 6.596

7.8V voltage corresponding AD value is 2058, 12.9 V voltage corresponding AD value is 3399. 2058–3399 were divided into five equal parts, which were represented by different color blocks in the software interface, namely, red(2058–2325), yellow(2326–2593), yellow-green(2594–2861), light green(2862–3129) and dark green(3130–3399).

Application Software Design

System Function

The pavement seepage coefficient measurement software operates the seepage meter through the control system. Its function is shown in Figure: 4, which mainly includes three program modules : test process, data processing and battery management.

The main program creates menus, battery management classes and attribute pages in WM_CREATE messages. The attribute page class includes two subclasses: test process and data processing. In the notification message of the attribute page class, the display and hiding of the two subclasses are processed according to the active focus to realize page switching.

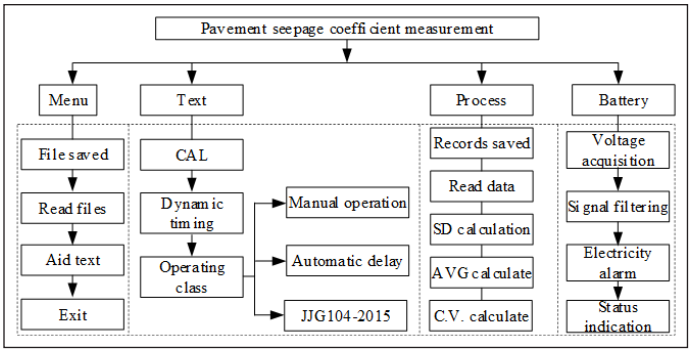

Figure 4: Functional Diagram of the Application Software

Test Process

Test process class is mainly responsible for data acquisition, solenoid valve switch, timing display, event processing and other operations. This kind of process processing function first draws each control of the interface, initializes parameter variables, and updates the color of the control. Then, the input messages of users are detected circularly, and the three types of operation: namely manual operation, JJG104-2015 and automatic delay are judged. Finally, the test process is automatically executed by the delay time, 100 mL scale line recognition event and 500 mL scale line recognition event.

Manual Operation

The manual operation process is completely controlled by manually clicking on the opening valve, closing valve, starting test and stopping test buttons on the touch screen, as shown in Figure: 5. The program sends switching instructions to the solenoid valve to eliminate the water leakage caused by manual switching valve. At the same time, the timing operation is accomplished and is more accurate than manual pinch. When the test is completed, the start and stop time and the corresponding water level pressure are automatically recorded for calculation and analysis.

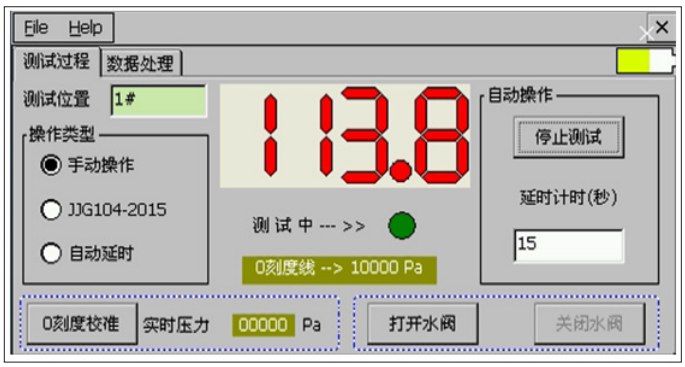

Figure 5: Operation Interface of Process Flow for Manual Operation

JJG104-2015 Industry Standards

The JJG104-2015 process adopts the industry standard specification. The record water level is started by 100 mL scale line, and stopped by 500 mL scale line or 3min decline time. The program reads the pressure value in real time for screening and judgment. It can be seen from the above code that after the JJG104 - 2015 mode is selected, the 'Start Test' button is clicked. At this time, the WaitFor_100 mL_WaterLevel thread will be started. If the water level cannot drop to 100 mL scale within 3 min. Then, the message that is directly terminated will be sent to the main program:

Send Message (hwndRecvWin, WM_TEST_START, (WPARAM)1, (LPARAM)1);

Represents that the liquid level does not reach the specified scale line. If the water level normally passes through the 100 mL scale line, the WaitFor_500 mL_WaterLevel thread is started. The operation interface is shown in Figure: 6.

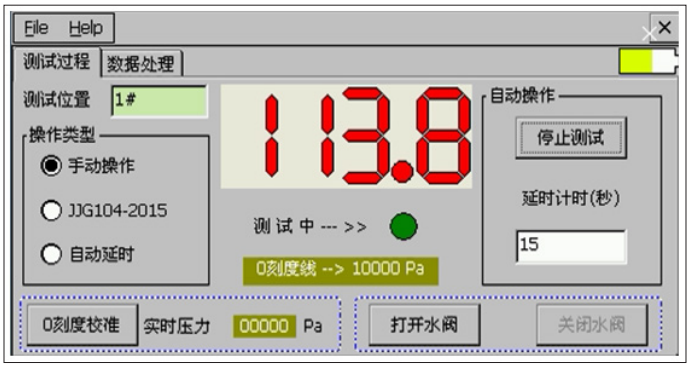

Figure 6: Operation interface of process flow for standard specification

Automatic Delay

Automatic delay process refers to skipping the pre-seepage step after filling the water and filling the lower space after opening the valve. The delay time can be set in the 'delay time (second)', and the test step can be started after the water level is stable. The 'start test' button label is displayed as 'countdown time...' and presents a state of prohibition. The switch of the solenoid valve and the end of the test process are automatically controlled by the delay thread, as shown in Figure 7.

Figure 7: Operation Interface of Process Flow for Automatic Delay

Data Processing

The application software based on embedded operating system has powerful data processing ability, which is more flexible and convenient than the single chip microcomputer control system based on microprocessor. Using its efficient computing ability and visual man-machine interface interaction ability, the measurement data are displayed, deleted, appended, calculated, stored and consulted, and the measurement results are monitored and judged in time. The data runs in the interface program, as shown in Figure: 8. The data processing window first creates the interface buttons, static text, and list controls, and then processes the button messages in turn. 'Save data ' stores the new test data as a file named by the current time, and the data transferred from the file will not be stored and processed. 'Statistical Computation' can perform statistical calculations for test data already in the list or for data entered from the file. The 'empty record' deletes the data transferred from the file or stored data. If the data is not stored, the pop-up window prompts the user to save the data first. 'Additional records' add backwards on the basis of existing data records automatically. If the data is statistically calculated, the results of statistical calculation are automatically deleted and then added. 'Open Files' to invoke stored data files, and if there is unsaved data in the list box, prompt the user to 'save data', then empty the original data automatically, and invoke file data to the list box.

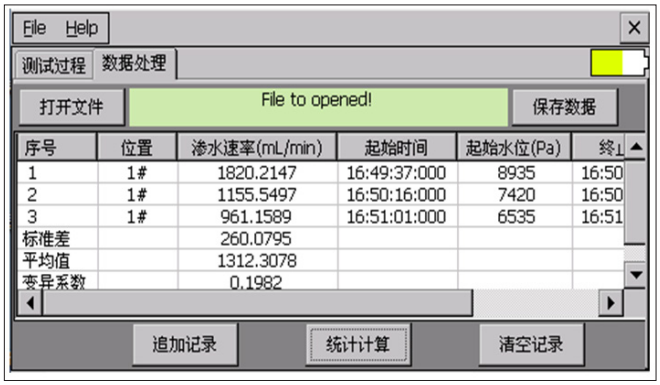

Figure 8: Data Recording and Analysis Interface for the Test

Battery Management

The battery management module periodically reads the battery voltage, maps it to shape and color, and prompts users through changes in shape width and color, as shown in Figure: 9. The main function of BatteryStatusProc starts the clock in the WM_CREATE message. The WM_TIMER message is generated every 1000 ms, and the battery voltage is read from the A/D interface of the touch screen. Due to the large fluctuation of the data, the threshold and anti-trembling filtering must be carried out. Then, the values are segmented and mapped, and the battery voltage display area is refreshed through the WM_PAINT message. The detailed code of each function is shown in Figure: 10.

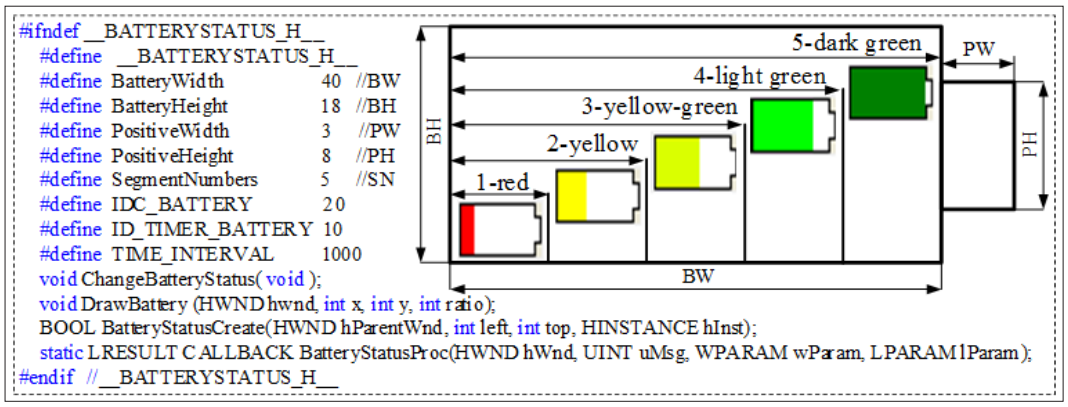

Figure 9: Map of battery voltage to colors and shapes

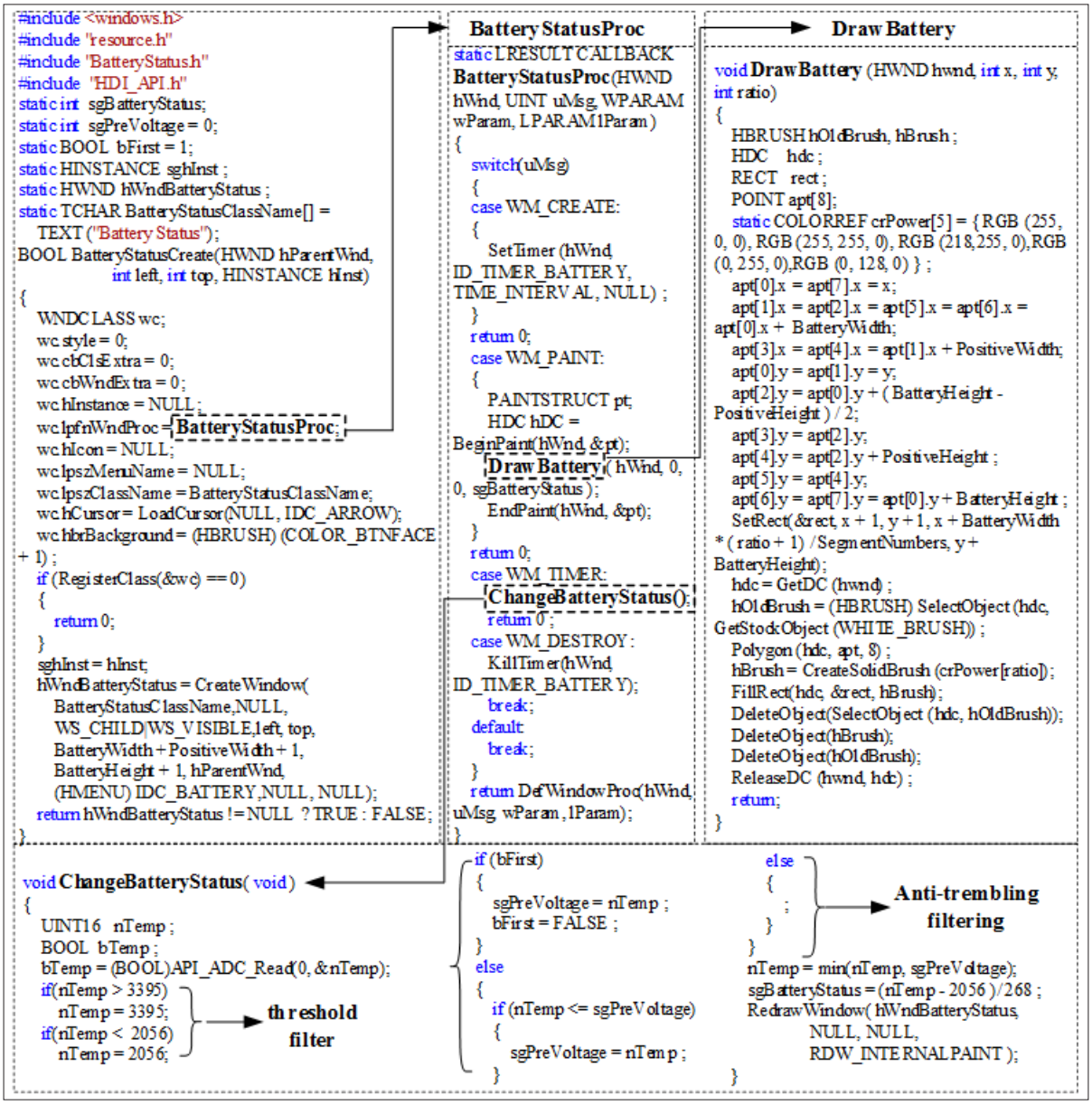

Figure 10: Logical Diagram of Functions for Monitoring Battery Voltage

Data Analysis

In addition to the control function, the software can automatically complete the measurement of seepage coefficient, and also provide the process measurement data. According to the observed phenomena and data, users can comprehensively compare and analyze them to obtain the detailed process of pavement seepage, thus forming a wide range of feature recognition algorithms and establishing the mechanism model of pavement seepage process. From the current measured data, the following features can be identified.

Infiltration

The infiltration is a normal permeation process, and the pressure process curve is shown in Figure: 11. There are a large number of burrs in the original data curve, indicating that the sensor and liquid level are disturbed by external factors in the acquisition process, such as vehicle passing, personnel walking, ground vibration, liquid level fluctuation, etc. The impact of data fluctuations will be follow-up analyzed, and the recursive median algorithm is used to filter the data. The burrs are removed and the smooth curve is obtained. The seepage coefficient was calculated to be 11.914mL/min from the near-linear segment corresponding to the seepage process.

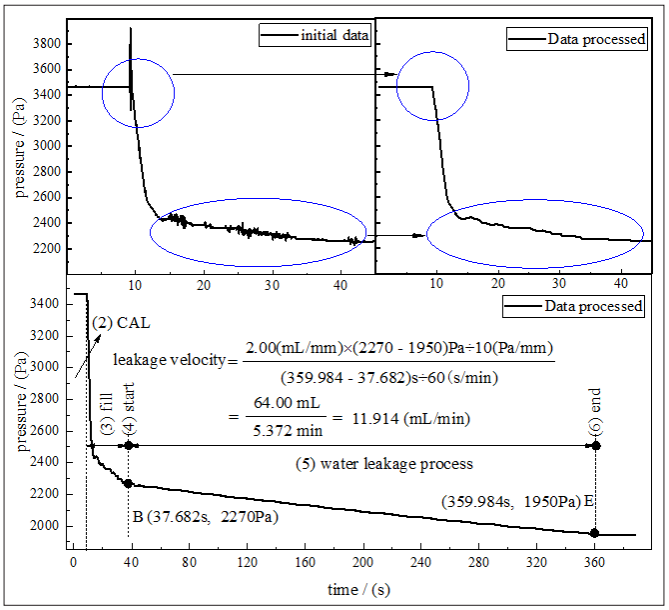

Figure 11: Curve of Normal Seepage Process

Water Leakage

The water leakage process shows a big change in the curve, and the value of seepage coefficient is abnormal, as shown in Figure: 12, which is 627.084 mL/min. Subsequent values suddenly smaller, indicating that the water has leaked out. The leakage process can be identified by analyzing the slope of the curve. If the first derivative of the curve is calculated, there will be a jump of two platforms. The corresponding derivative of the water leakage process is large, and the corresponding derivative of the subsequent process is small. There is a large fault between them.

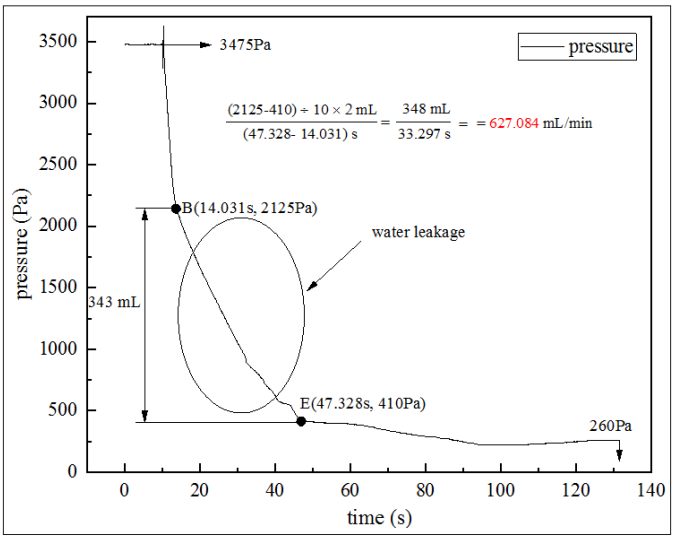

Figure 12: Curve of Water Leakage Process

Defoaming

The defoaming process is mainly caused by the burst of internal bubbles. Due to the wetting between the road surface and the water during the test process, some small bubbles are produced. These bubbles grow up gradually with the decrease of water level pressure, and finally burst. The pressure value in the corresponding process curve decreases suddenly and the gradient is small. If not corrected, it is considered to be normal seepage. From Figure: 13, it can be seen that the seepage coefficient obtained without considering bubbles is 38.152 mL/min, and the seepage coefficient obtained after deducting bubbles is 28.250 mL/min.

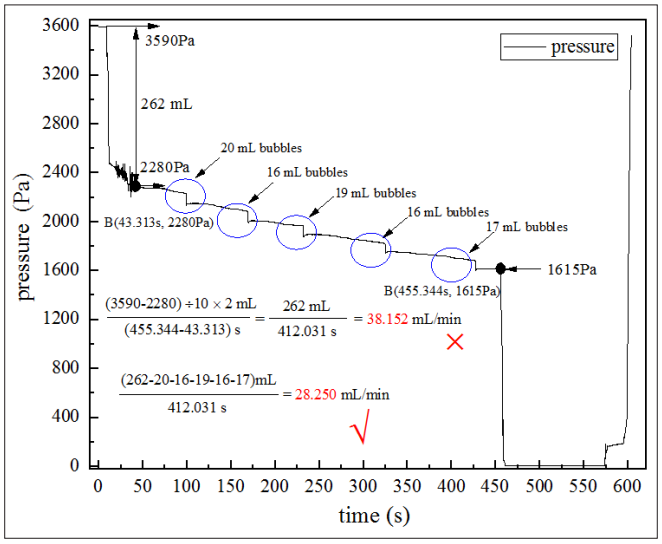

Figure 13: Curve of Bubble Burst Process

Conclusions

The measurement process is fully automated and the measurement accuracy is improved by using the embedded touch screen based on Windows CE combined with the industry standard specification of seepage coefficient measurement. The data terminal processing and analysis are completed by using the ability of the operating system and human-computer interaction ability. The main innovations are as follows:

1. A set of automatic measurement system for seepage coefficient is developed by combining the digital sensor and embedded system and using the advantages of touch screen communication interface.
2. The time and event threads are used to realize the automatic measurement and data recording of the software, and a complete set of test software is formed.
3. The process data reflect the test phenomenon, and a large number of data analysis is conducive to the intelligent identification of infiltration, water leakage and defoaming.

In order to improve the screening of test process characteristics, it is necessary to analyze a large number of data and establish a reliable theoretical model, which will be further studied.

Acknowledgements: No fundings were support for this study.

Conflicts of interest: The authors declare that they have no competing financial interests.

Appendix:

In order to clearly analyze the correlation between events, the detailed code is as follows :

```
#ifndef __CONTROLUI_H__  //ControlUI.h
#define  __CONTROLUI_H__
#define  IDC_TIME_DISPLAY   21
#define  IDC_EDT_DELAYTIME  1022
#define  PRESSURE_OF_100ML  501
#define  PRESSURE_OF_500ML  2506
#define  TIMER_CONTROLUI_DELAY  100
#define  WM_TIME_STOP       (WM_USER+31)
#define  WM_TEST_START      (WM_USER+32)
#define  WM_TEST_END        (WM_USER+33)
static  DWORD  WaitFor_TimeOut_AutoOperation(PVOID
pParam);
static DWORD WaitFor_100mL_WaterLevel(PVOID pParam);
static DWORD WaitFor_500mL_WaterLevel(PVOID pParam);
```

```
static enum  OperationType {ManualOperation, JJG104To2015,
                AutoOperation }  otStatus = ManualOperation ;
  typedef  struct  tagDataWaterP
  {
    DWORD  dwRecvData;
    int   nDataPressure;
    int   nDataWaterHeight
  }DATA_WATER_P;
  typedef struct tagTestParams
  {
  int     nPresZero ;
  int     nOperateState ;
  } TEST_PARAMS ;
  typedef  struct tagRecvParam
  {
    HWND   hwndRecv ;
    DWORD  dwDelayTime ;
  }RECV_PARAM;
  typedef  struct tagRecv100mL
  {
    HWND   hwndRecv ;
    DWORD  dwElapseTime ;
  }RECV_100mL;
  typedef  struct tagRecv500mL
  {
    HWND   hwndRecv ;
    DWORD  dwElapseTime ;
  }RECV_500mL;
#endif //__CONTROLUI_H__
#include  <windows.h>   //ControlUI.cpp
#include  <tchar.h>
#include  "ControlUI.h"
HANDLE  hCom;
static BYTE byTxBuf[8] = {0x01, 0x03, 0x00, 0x00, 0x00, 0x01,
0x84, 0x0A};
static BYTE byRxBuf[7] = {0};
static DATA_WATER_P  dwpWaterPressure = {0};
static TEST_PARAMS tpTestParams = { 10000 , ManualOperation
};
static   HANDLE   hEvent_100mL = NULL ;
static   HANDLE   hEvent_500mL = NULL ;
static  BOOL  bBtnStartTestFlag = FALSE ;
……
LRESULT CALLBACK ControlUIPageProc(HWND, UINT
message, WPARAM, LPARAM lParam)
{
  ……
  switch(message)
  {
  ……
  case WM_CREATE:
    {
      hEvent_100mL = CreateEvent(NULL, FALSE, FALSE,
TEXT("100mL"));
    if (NULL == hEvent_100mL)    return   0 ;
      hEvent_500mL = CreateEvent(NULL, FALSE, FALSE,
TEXT("500mL"));
    if (NULL == hEvent_500mL)    return   0 ;
    ……
    }
    return 0 ;
  case WM_TIMER:
    {
    ……
      int    nTmpPressure ;
    if( bDataTransfer(hCom, byTxBuf, 8, byRxBuf, 7, TIMER_
CONTROLUI_DELAY))
        {
          dwpWaterPressure.dwRecvData = byRxBuf[3]<<8 |
byRxBuf[4];
        dwpWaterPressure.nDataPressure = dwpWaterPressure.
dwRecvData * 5;
        nTmpPressure = dwpWaterPressure.nDataPressure ;
      if(nTmpPressure <= (tpTestParams.nPresZero - PRESSURE_
OF_100ML))
          SetEvent(hEvent_100mL);
      if(nTmpPressure <= (tpTestParams.nPresZero - PRESSURE_
OF_500ML))
          SetEvent(hEvent_500mL);
        ……
        }
      }
      return 0;
  case WM_COMMAND:
    {
    int  wmId = LOWORD(wParam);
    switch (wmId)
    {
      ……
      case IDC_BTN_BEGIN_TEST:
        {
        HWND   hwndShowTime, hwndDelayTime ;
        TCHAR   strTemp[32] ;
        ……
        bBtnStartTestFlag = !bBtnStartTestFlag ;
          hwndDelayTime = GetDlgItem(hwnd, IDC_EDT_
DELAYTIME) ;
          hwndShowTime = GetDlgItem(hwnd, IDC_TIME_
DISPLAY) ;
        if(bBtnStartTestFlag)
          {
            switch(otStatus)
            {
              ……
              case JJG104To2015:
                {
                RECV_100mL *  pRecv100mL;
                HANDLE    hWaitFor100mL ;
                ……
                SetValveState(1);
                    pRecv100mL = (RECV_100mL *)
malloc(sizeof(RECV_100mL));
                if(NULL == pRecv100mL)         return  0;
                pRecv100mL->hwndRecv  = hwnd ;
                pRecv100mL->dwElapseTime  = 0 ;
                ResetEvent(hEvent_100mL);
              hWaitFor100mL = (HANDLE)CreateThread(NULL,
0, WaitFor_100mL_WaterLevel, (PVOID)pRecv100mL, 0,
NULL);
                CloseHandle(hWaitFor100mL);
                bBtnStartTestFlag = !bBtnStartTestFlag ;
                }
                break;
              case AutoOperation:
                {
                int   nCountDown ;
                RECV_PARAM  *  pRecv_Param ;
                HANDLE   hAutoOperation ;
```

```
                    SetValveState(1);
                    ……
                GetWindowText(hwndDelayTime, strTemp, 32) ;
                    nCountDown = 1000 * _ttoi(strTemp) ;
                        pRecv_Param = (RECV_PARAM *)
malloc(sizeof(RECV_PARAM));
                    if(NULL == pRecv_Param)    return  0;
                    pRecv_Param->hwndRecv = hwnd ;
                pRecv_Param->dwDelayTime = nCountDown ;
              hAutoOperation = (HANDLE)CreateThread(NULL,
0, WaitFor_TimeOut_AutoOperation, (PVOID)pRecv_Param, 0,
NULL);
                    CloseHandle(hAutoOperation);
                    bBtnStartTestFlag = !bBtnStartTestFlag ;
                    }
                    break;
                  default:
                    break;
               }
            }
          else
            {
            SendMessage(hwndShowTime, WM_TIME_STOP,
(WPARAM)NULL,  (LPARAM)NULL );
          SendMessage(hwnd, WM_TEST_END, (WPARAM)0,
(LPARAM)(GetTickCount() - dwWaterlevelBeginTime));
             InvalidateRect(hwnd, NULL, TRUE);
            }
          }
        default:
          break;
    }
    return 0;
  ……
  default:
    break;
  }
  return DefWindowProc (hwnd, message, wParam, lParam) ;
}
static DWORD WaitFor_TimeOut_AutoOperation(PVOID
pParam)
{
  HWND  hwndTimeDisplay ;
   HWND   hwndRecvWin = ((RECV_PARAM *)pParam)-
>hwndRecv ;
   DWORD  dwDelayTime = ((RECV_PARAM *)pParam)-
>dwDelayTime ;
  free(pParam);
 hwndTimeDisplay = GetDlgItem(hwndRecvWin, IDC_TIME_
DISPLAY);
  Sleep(dwDelayTime);
     SendMessage(hwndRecvWin,  WM_TEST_START,
(WPARAM)2, (LPARAM)NULL);
  Sleep(180000);
 SendMessage(hwndRecvWin, WM_TEST_END, (WPARAM)2,
(LPARAM)(GetTickCount() - dwWaterlevelBeginTime));
    SendMessage(hwndTimeDisplay,  WM_TIME_STOP,
(WPARAM)NULL, (LPARAM)NULL);
  return  0;
}
static DWORD WaitFor_500mL_WaterLevel(PVOID pParam)
{
  int nRetValue;
  HWND  hwndTimeDisplay ;
  DWORD  dwTimeStart, dwTimeElapse ;
   HWND   hwndRecvWin = ((RECV_500mL *)pParam)-
>hwndRecv ;
   DWORD  dwElapseTime = ((RECV_500mL *)pParam)-
>dwElapseTime ;
  free(pParam);
 hwndTimeDisplay = GetDlgItem(hwndRecvWin, IDC_TIME_
DISPLAY);
  dwTimeStart = GetTickCount();
  nRetValue = WaitForSingleObject(hEvent_500mL, 180000);
  dwTimeElapse = GetTickCount() - dwTimeStart ;
  if((WAIT_OBJECT_0 == nRetValue)||(WAIT_TIMEOUT ==
nRetValue))
  {
        SendMessage(hwndRecvWin,  WM_TEST_END,
(WPARAM)1, (LPARAM)dwTimeElapse);
      SendMessage(hwndTimeDisplay, WM_TIME_STOP,
(WPARAM)NULL, (LPARAM)NULL);
  }
  return  0;
}
static DWORD WaitFor_100mL_WaterLevel(PVOID pParam)
{
  int  nRetValue ;
   HWND   hwndRecvWin = ((RECV_100mL *)pParam)-
>hwndRecv ;
   DWORD  dwElapseTime = ((RECV_100mL *)pParam)-
>dwElapseTime ;
  free(pParam);
  nRetValue = WaitForSingleObject(hEvent_100mL, 180000);
  if(WAIT_OBJECT_0 == nRetValue)
  {
        SendMessage(hwndRecvWin,  WM_TEST_START,
(WPARAM)1, (LPARAM)0);
  }
  else  if(WAIT_TIMEOUT == nRetValue)
  {
        SendMessage(hwndRecvWin,  WM_TEST_START,
(WPARAM)1, (LPARAM)1);
  }
  return  0;
}
```

References

1. Rosenberry Donald O, José Manuel Nieto López, Richard M T Webb, Sascha Müller (2020) Variable Seepage Meter Efficiency in High-Permeability Settings[J]. Water 12: 3267-3267.
2. Donald O Rosenberry, Carlos Duque, David R (2020) LeeScience - Earth Science; Studies from U.S. Geological Survey (USGS) Yield New Data on Earth Science (History and Evolution of Seepage Meters for Quantifying Flow Between Groundwater and Surface Water: Part 2-marine Settings and Submarine Groundwater Discharge) [J]. Science Letter https://pubs.er.usgs.gov/publication/70216725.
3. Lee Changyong, Kim Wonbin Jeen SungWook (2020) Measurement of Flux at Sediment–Water Interface Using a Seepage Meter under Controlled Flow Conditions [J] Water 12: 3071-3071.
4. American Society for Testing Materials (2005) Standard Test Method for Measuring Pavement Texture Drainage using an Outflow Meter[S]. US: ASTM, 2005: 1-4.
5. Shaw R.D, Prepas EE (1990) Groundwater-lake interactions: I. Accuracy of seepage meter estimates of lake seepage[J]

119: 105-120.
6. Palash Debnath, Abhijit Mukherjee (2016) Quantification of tidally-influenced seasonal groundwater discharge to the Bay of Bengal by seepage meter study[J]. Journal of Hydrology 537: 106-116.
7. BM Mwashote, WC Burnett, J Chanton, IR Santos, N Dimova, et al. (2009) Calibration and use of continuous heat-type automated seepage meters for submarine groundwater discharge measurements[J]. Estuarine, Coastal and Shelf Science 87: 1-10.
8. Solder John E, Troy E Gilmore, David P Genereux, D Kip Solomon (2016) A Tube Seepage Meter for In Situ Measurement of Seepage Rate and Groundwater Sampling [J]. Ground water 54: 588-595.
9. Lee Jeongwoo, Seon Geum Chun, Myeong-Jae Yi, Nam Won Kim, Il-Moon Chung, et al. (2015) Measurements of Streambed Hydraulic Conductivity Using Drive-point Piezometers and Seepage Meters in the Upper Reaches of Anseong Stream[J]. The Journal of Engineering Geology 25: 413-420.
10. Russoniello Christopher J, Michael Holly A (2015) Investigation of Seepage Meter Measurements in Steady Flow and Wave Conditions.[J]. Ground water 53: 959-966.
11. Dirk Koopmans, Peter Berg (2011) An alternative to traditional seepage meters: Dye displacement[J] 47.
12. JE Cable, Burnett WC, Chanton JP, Corbett DR, Cable PH (1997) Field Evaluation of Seepage Meters in the Coastal Marine Environment[J]. Estuarine, Coastal and Shelf Science 45: 367-375.
13. Donald O Rosenberry, Martin A Briggs, Geoffrey Delin, Danielle K Hare (2016) Combined use of thermal methods and seepage meters to efficiently locate, quantify, and monitor focused groundwater discharge to a sand-bed stream[J]. Water Resources Research 52: 4486-4503.
14. Rosenberry Donald O, José Manuel Nieto López, Richard MT Webb, Sascha Müller (2020) Variable Seepage Meter Efficiency in High-Permeability Settings[J]. Water 12: 3267-3267.
15. Russoniello Christopher J, Michael Holly A (2015) Investigation of Seepage Meter Measurements in Steady Flow and Wave Conditions[J]. Ground water 53: 959-966.