We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Development of an Intelligent Bedsore Prevention System

Runjing Zhou
Inner Mongolia University
China

1. Introduction

Bedsore is a very common disease for long-term bedridden patients, and its main cause is that the human skins have been pressed too long in part areas, blood circulation obstruction lead to tissue necrosis. The conventional prevention measures include turning the body over at regular intervals by medical care personnel, using mattress and so on. Manpower nursing will increase the working strength of medical care personnel, and the common mattress only can be set to run periodically, it can not estimate the practical pressures of the different human body parts, as a result, if the time interval is set too long, maybe some parts can not be treated in time, and if the time interval is set too short, it will affect the rest of patient. A new method is put forward to improve the conventional bedsore prevention mattress, and the mattress is divided into areas to be controlled respectively. Several groups of stress sensor are used to measure the pressures signals in each area, and then the pressure signals are sampled to the computer. The LabVIEW virtual instrument is designed to analyze the measure pressure data, and estimate the state of every parts of the human body that have been continuously pressed. For the parts warned by the system, the LabVIEW instrument will send control signals to the corresponding pumps of the mattress.

2. System design and structure

The main cause of bedsore is that the part skin has been pressed too long, and the magnitude and duration of pressure are the crucial factors. According to the anatomical characteristics of human rachis, the pressures are bigger in four parts when the patient lies low, the four parts are head, scapular, sacrum and cruses. If these parts keep immobile for a long time, blood circulation of the corresponding parenchyma will be cumbered, and it will cause bedsore. According to amount of clinic experiments, in general, the pressure of sacrum is the biggest, that of the head and the scapular follows, and the pressure on cruses is the smallest among the four parts, therefore, the mattress can be divided into four areas, as shown in Fig. 1. Considering the position of the patient is not always in the center of the mattress, the numbers of stress sensor for practical system can be increased.

The whole system structure is shown in Fig. 2, pressure signals in each area of the mattress are conditioned and sampled to the computer, and then analyzed in the LabVIEW virtual instrument. Four pumps in different areas of the mattress are controlled by the signals from the virtual instrument.
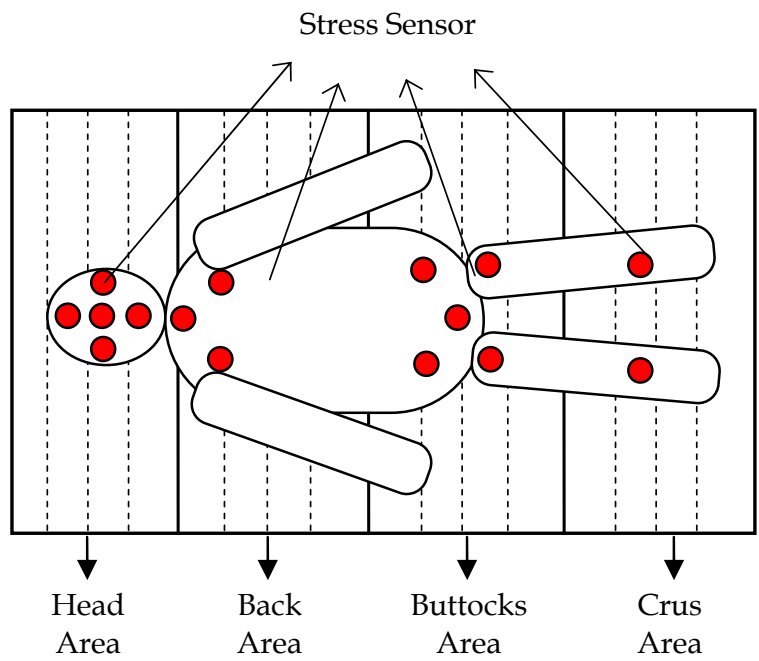
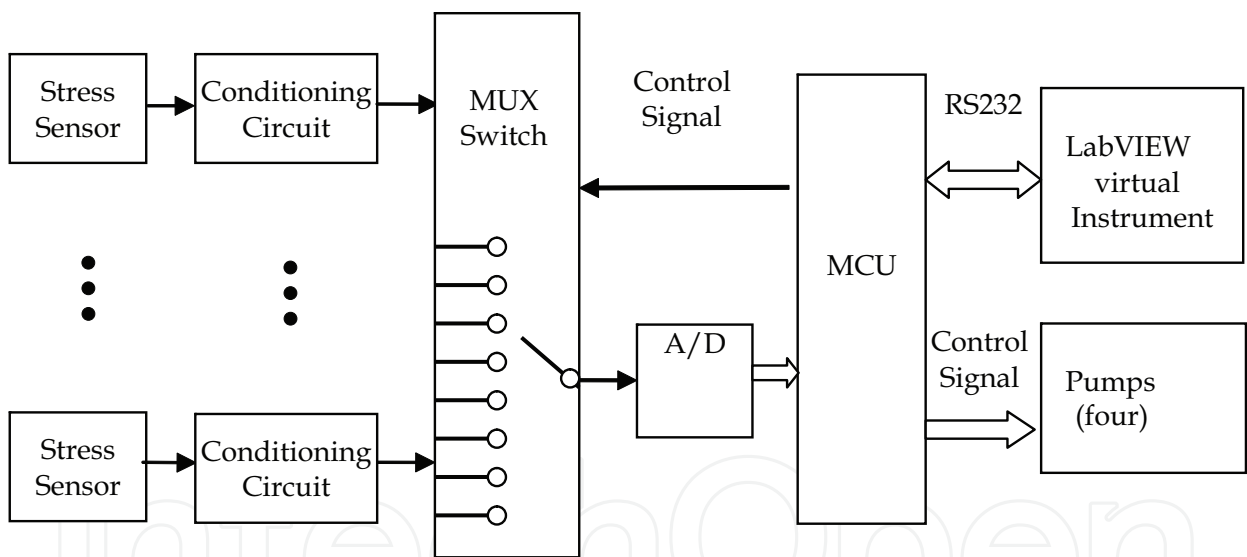Fig. 1. Mattress partition and sensors distributing

Fig. 2. System structure

3. Hard design

3.1 Pressure signal conditioning
3.1.1 Stress sensor selection

MPX2100 semiconductor pressure sensor manufactured by Motorola can convert pressure into a millivolt-level differential-mode voltage signal, the pressure sensor has good linearity, and the output voltage has a precise direct proportion with the pressure (Zhou, 2009). In addition, MPX2100 the compensation characteristics with temperature overcome the problem of the temperature drift of semiconductor pressure-sensitive semiconductor devices. So the system uses the MPX2100 pressure sensor, which can meet the system

requirement. The physical map and pins map are shown in Fig.3 and Fig.4. MPX2100 has the following characteristics:

- As using a laser trimming technology to make the output of the bridge zero drift become small, generally less than ± 1mV;
- Having high sensitivity: 40mV± 1.5mV;
- Thermosensitive sensor compose of a temperature compensation network, in the range of -40 °C and +125 °C having a good temperature compensation effect, thereby enhancing the accuracy of the sensor;
- A better linearity (±0.25% F.S);
- A wide temperature range (-40 °C ~ +125 °C);
- Allowable large overload (400%).

Fig. 3. MPX2100 physical map

MPX2100 has 4 pins. Pin 1 connects to ground and pin 3 connects to the working voltage. Moreover, the output between pin 2 and pin 4 is differential-mode voltage signal which is proportional to the pressure.
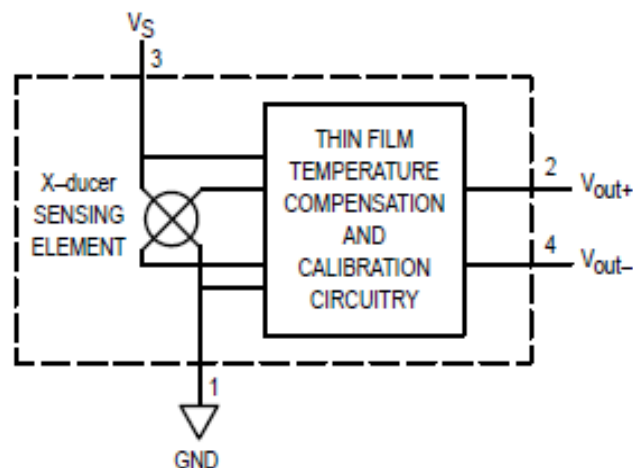

Fig. 4. MPX2100 pins map

MPX2100 is a piezoresistive pressure sensor, four equal resistors components are made by using diffusion technology on the silicon chip, which is constitute the Whiston bridge. Bridge have two power supply of voltage source power supply and constant current power supply. The principle of constant voltage source powered is shown in Fig.5.

When the pressure sensor under the pressure, a pair of arm resistance value increase ΔR, another pair of arm resistanc value decrease ΔR, resistance variation ΔR and stress is proportional, that is:

$$\Delta R = KP \qquad (1)$$

The bridge output voltage is:

$$Uo = E \, (\Delta R \, / \, R) = (EK \, / \, R) \, P \qquad (2)$$

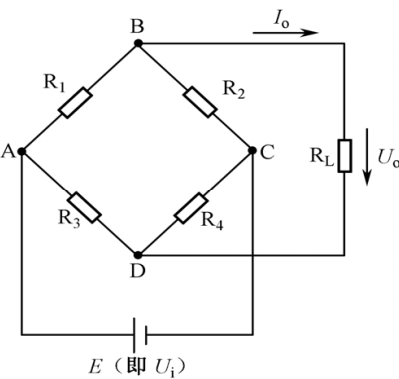The bridge output voltage is proportional to the pressure P.

Fig. 5. Whiston bridge

3.1.2 Amplification circuit

The main amplification circuit is shown in Fig.6. The amplifier has a strong common-mode rejection ratio. It consists of two-stage amplifier, and the first-level is compose of integrated amplifier A1 and A2, as using the same type of operational amplifier, it can further reduce the drift. Resistors R1, R2 and R3 are composed of the same-phase input parallel differential amplifier with very high input impedance. The second-level comsist of inverse ratio amplifier by amplifier A3 and four resistances of R4, R5, R6 and R7, it will change a double-end input into a single-end output. The resistance is set as follows:

$$R1 = R3 \qquad (3)$$

$$R4 = R5 \qquad (4)$$
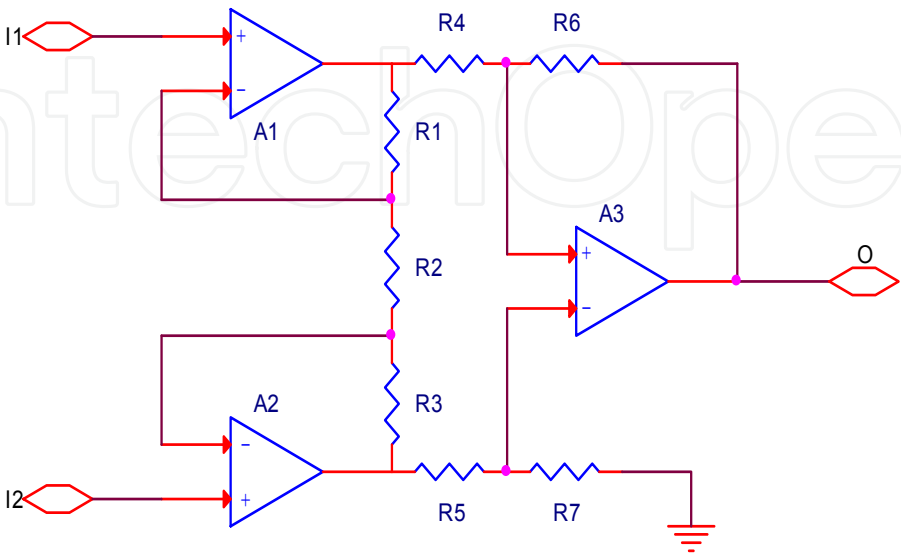
$$R6 = R7 \qquad (5)$$

Fig. 6. Instrument amplifier circuit

According to the basic analysis method of operation circuit, the available output voltage is:

$$U_o = -\frac{R_6}{R_4}(1 + 2\frac{R_1}{R_2})(U_{I1} - U_{I2}) \tag{6}$$

To facilitate adjustment, the circuit increase a part of percentage of amplifier. At the same time, making the the output signal of the instrument amplifier circuit reverse, which is shown in Fig.7. RW is zero adjustment resistance.
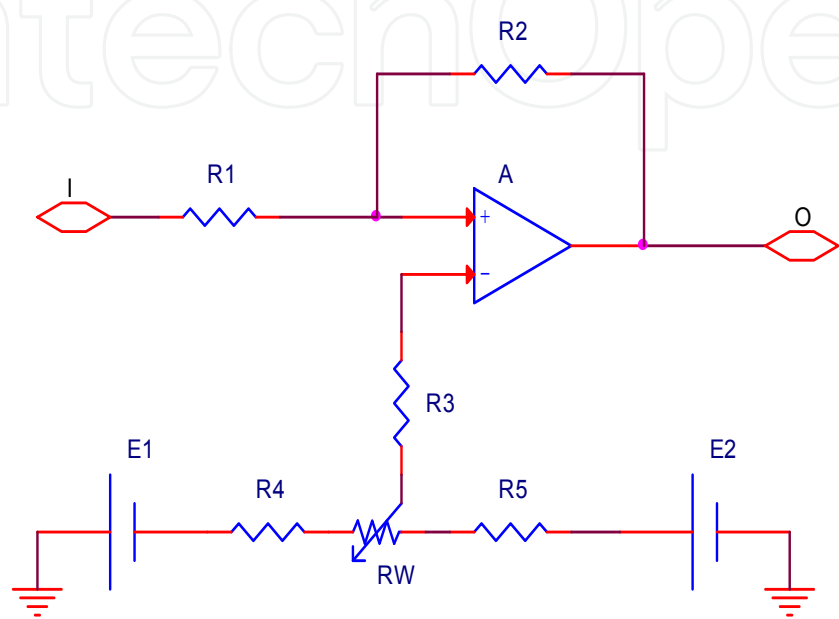
Fig. 7. Rate amplifier circuit

### 3.1.3 Integrated circuit design
Therefore, pressure measurement circuit is completed based on MPX2100. Integrated circuit is shown as follows. Pressure signal transmitted the signal to the multi-selector switch through IN1 port. Another 14-channel signal sampling circuit is the same with the Fig.8.

In the Fig.8, RW1 is the gain adjustment potentiometer; RW2 is zero adjustment potentiometer of the amplifier circuit. Amplifier circuit choose OP07CP as an amplifier, because it is an operation amplifier with low noise and a low offset voltage. In addition, the diode VD1, VD2 may play a protective effect on the circuit.

3.2 Control circuit
This system contain so much sensors, if sampling all sensor signals at the same time, the hardware design will be more complex. Multi-channel selection switch are timing of gating and cycle-sampled for the signal of every channel. For long-term bedridden patients, the pressure signal in different part of the human body changes relatively slow, so sampling period can be set slightly longer. Generally setting to one minute, it can meet the system requirement.

3.2.1 Microcontroller selection
So selecting AT89S52 microcontroller to design system not only meets the requirement of design but also can on-line program on the circuit board and taking convenient program debugging.

Fig. 8. Pressure measurement circuit
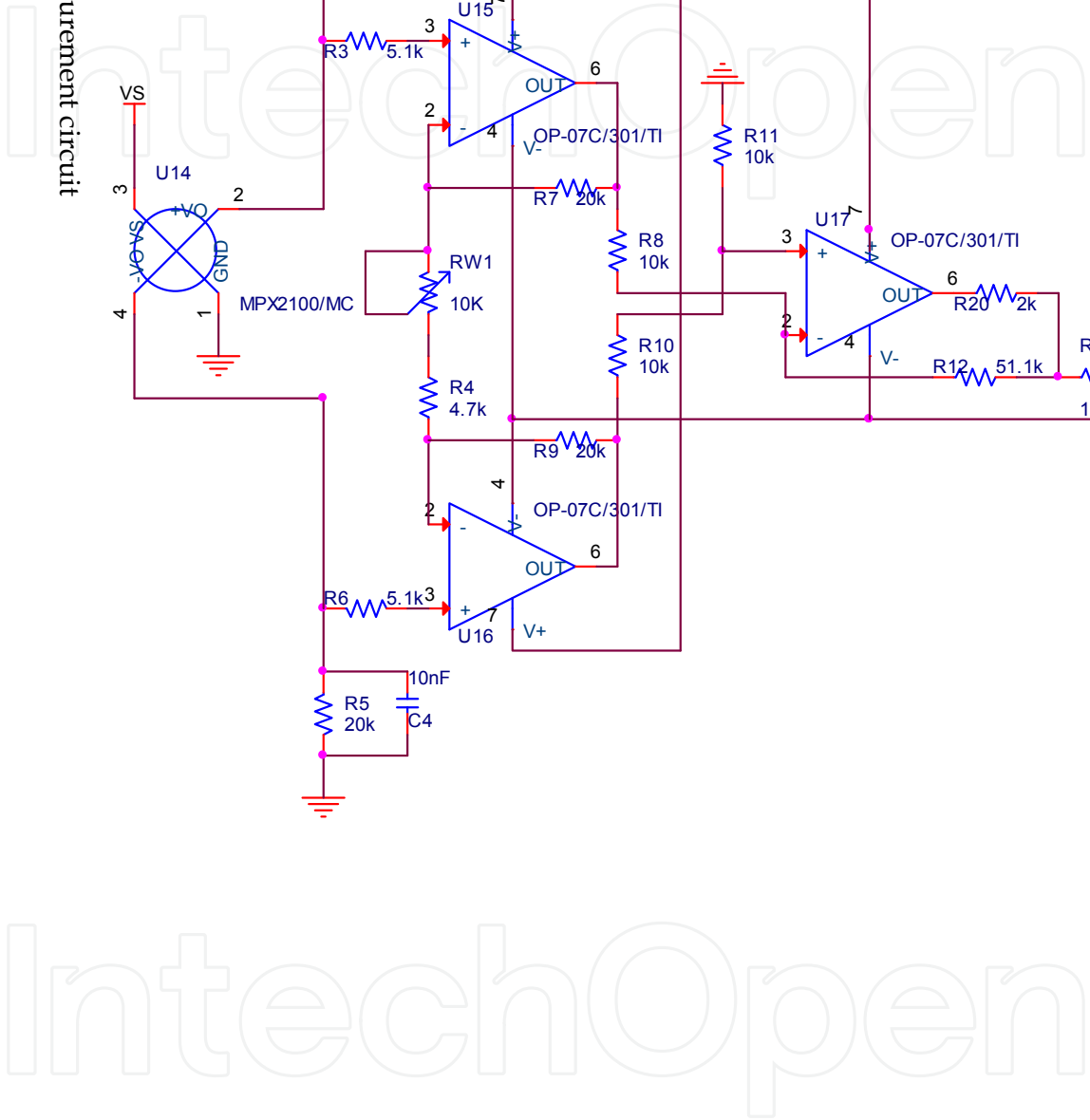
3.2.2 A/D converter selection

It is necessary to analyse sampled press signal in this design, taking the detail information of the signal, so sampled signal need to have higher resolution. Considering the performance-price ratio of hardware design, 12-bit A/D converter is selected to take a sample of data. The main function parameters of MAX1240 are as follows:

- A single supply(+2.7V ~+3.6V);
- 12-bit resolution;
- Low power consumption: 37Mw(73 KSPS),5μW (standby working mode);
- Internal provide track/hold circuit;
- Internal provide conversion clock.

The pin distribution of MAX1240 is shown in Fig.9, MAX1240 interface timing sequence is shown in Fig. 10.
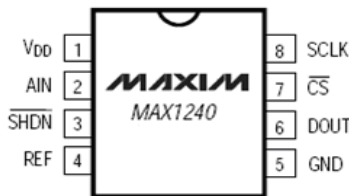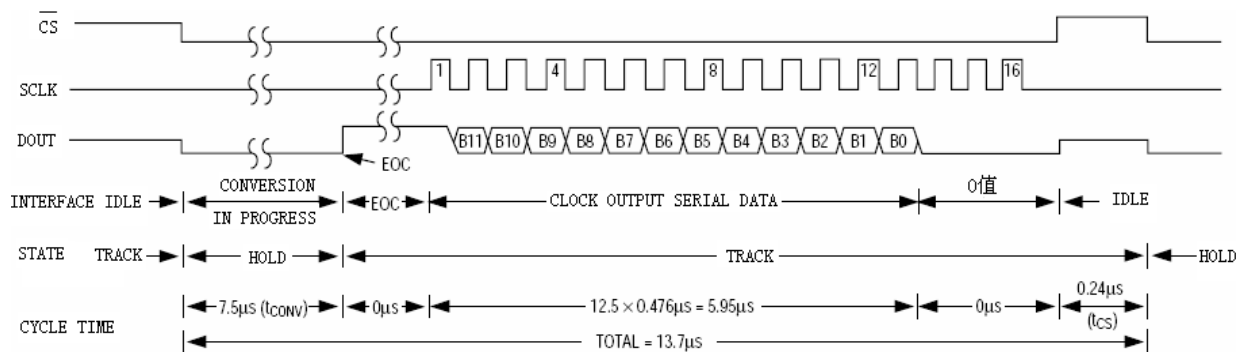

Fig. 9. MAX1240 pin distribution

Fig. 10. MAX1240 working timing sequence

3.2.3 Serial communication circuit

Serial communication line is simple and reducing design cost, so it is used more widely to transfer data in the close distance with less demanding speed. Serial communication can be divided into two basic transmission forms of asynchronous communication and synchronization communication. Asynchronous transmission is characterized by the data discontinuous transmission on the transmission line, but the two communication parties must agree the character format and baud rate at first. Faster than asynchronous transmission, the hardware of synchronous transmission is more complex and also has a very strict requirement for the phase consistency of the synchronize clock pulse signal.

Serial communication has established a number of the consistent concepts and standards to make the computer, telephone and other communication devices communicate with each other. The concepts and standards conclude:

Transmission rate: transmission rate refers to the transmitting the bit number per second, so transmission rate is also called baud rate. Most CRT devices can work according to any baud rate of the range from 110 to 9600.

RS-232-C standard: RS-232-C standard make a regulation for two aspects, that is signal level standard and control signal standard. RS-232-C use the logic level of negative logic regulation, signal level is not compatible with the normal TTL level. RS-232-C standard use EIA level, in which the high level is +3V to +15V and the low level is -3V to -15V. But the high level of the standard TTL level is +2.4V to +5V and the low level is 0 to 0.4V. If transforming the two levels, the specialized level conversion chip is needed. At present, MAX3232 is usually used as level conversion chip. The main function parameter is as follows:

- 3.0~5.5V power supply;
- 300μA low supply current;
- Only connect 0.1μF capacitance

MCS_51 microcontroller has a full duplex serial port. Full duplex serial communication only need a output line TXD and a input line RXD. Serial communication mainly has two technical problems. The one is data transmission, the other is data conversion. Data transmission mainly solves some problems about standard, format and working mode in the transmission. Data conversion is the purpose to complete the conversion of string and parallel. MCS-51 serial port reception and sending both can work in the query or interrupt way with the use of very flexible. 51 microcontroller serial programming involved some major special function register, which is serial data buffer SBUF, serial port control register SCON, special function register PCON and interrupt allowable register IE.

3.2.4 The whole circuit of microcontroller

A/D converter power supply voltage is 3.3V, and taking supply voltage as the external reference voltage (Hu, 2003). SPI timing sequence can be programmed by the MCU software. MAX1240 sampling rate is much larger than the transmission rate, so the sampling rate of the whole hardware sampling system is decided by the transmission rate of serial port.

Baud rate of both communication sides is necessary ensure of making serial communication normal. Serial port select working mode 1, baud rate B is determined by the overflow rate S of timer T1 and it can use the following formula to express:

$$B = \frac{2^{SMOD} \cdot S}{32} \tag{7}$$

$$S = \frac{f_o}{12(2^n - X)} \tag{8}$$

Where X is the calculation initial value of timer T1, n is the number bits of the timer T1 (select working mode 2, n=8), the value of serial baud rate multiplier bit SMOD is 0. In this design, we should select the right size of the crystal oscillator to make the baud rate of system the same with the baud rate of setting in the PC. The baud rate generated by microcontroller is equal to 9600, so we select the 14.7456MHz crystal oscillator as the microcontroller crystal oscillator.

Where X is the calculation of initial value of timer T1, n the number of bits for the timer T1 (timer selection work 2, n = 8), the serial baud rate multiplier bit SMOD values to 0. We should choose the right size for the single chip crystal oscillator, the baud rate generated by

the system and set the baud rate the same PC. This article hopes generated baud rate is equal to 9600 MCU, so select the 14.7456MHz crystal oscillator as a single chip.

3.2.5 Software design

The main tasks of the microcontroller which need to complete are data sending and data reception. The flow chart is shown in Fig.11, Data sending can send the data that the AD samples to the computer through the serial port. Data reception can receive the pressure signal processing results through the computer serial port, and make the corresponding control to the pump. The whole circuit of microcontroller is shown in Fig.12.
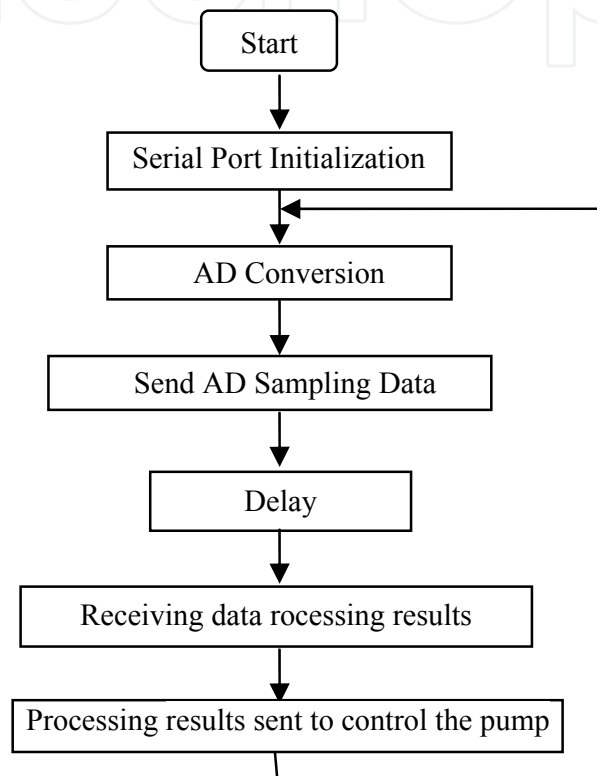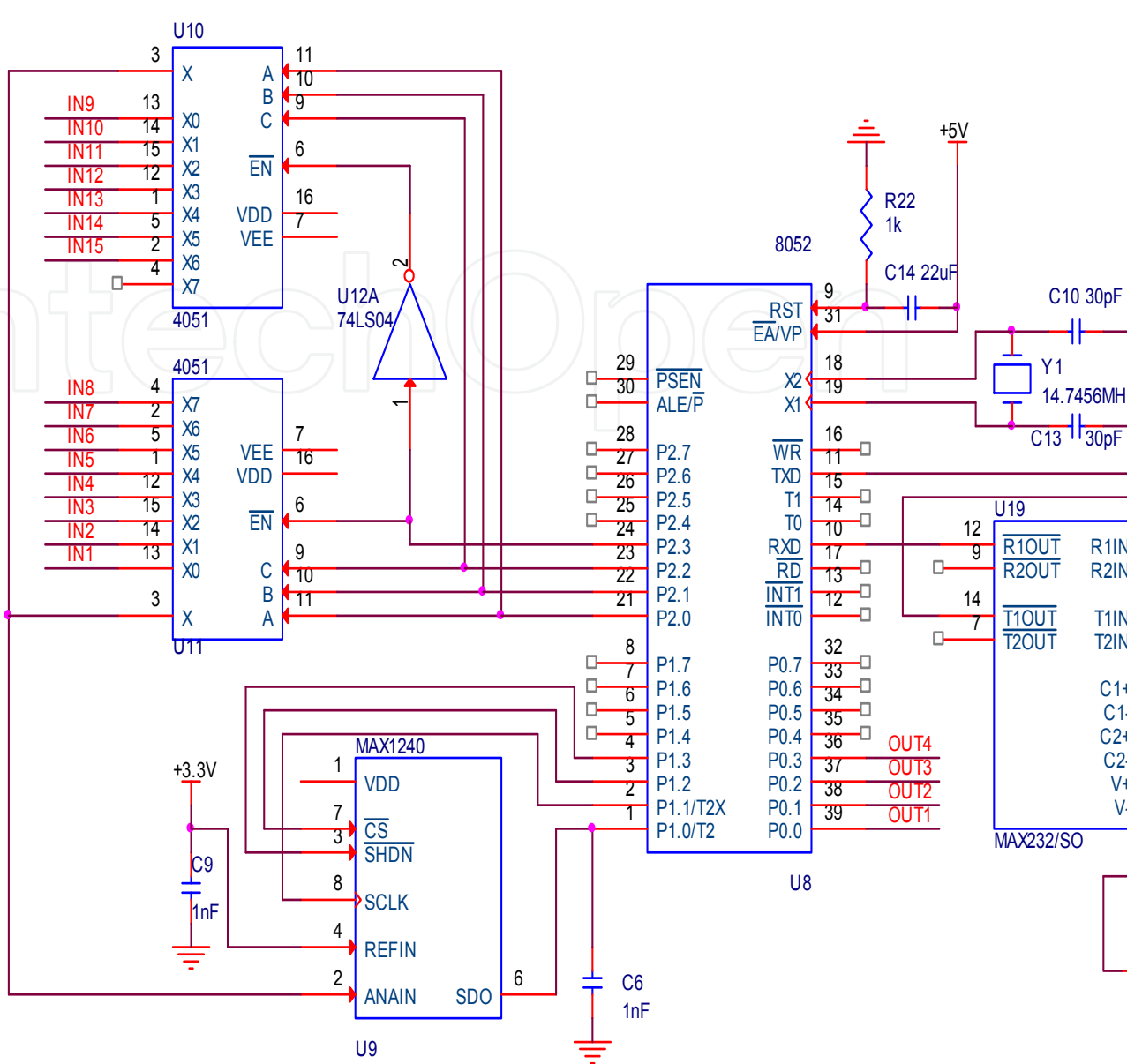
```
                    ┌─────────┐
                    │  Start  │
                    └─────────┘
                         │
                         ▼
            ┌──────────────────────────┐
            │ Serial Port Initialization│
            └──────────────────────────┘
                         │
                         ▼
              ┌────────────────────┐
              │   AD Conversion    │◄──────┐
              └────────────────────┘       │
                         │                 │
                         ▼                 │
            ┌──────────────────────┐       │
            │ Send AD Sampling Data│       │
            └──────────────────────┘       │
                         │                 │
                         ▼                 │
                 ┌──────────────┐          │
                 │    Delay     │          │
                 └──────────────┘          │
                         │                 │
                         ▼                 │
        ┌──────────────────────────────┐  │
        │ Receiving data rocessing results│ │
        └──────────────────────────────┘  │
                         │                 │
                         ▼                 │
    ┌────────────────────────────────────┐│
    │ Processing results sent to control the pump ││
    └────────────────────────────────────┘│
                         └─────────────────┘
```

Fig. 11. Flow chart

Source Code:
```c
#include <reg51.h>
#include<INTRINS.H>
#define uint unsigned int
#define uchar unsigned char
#define SomeNOP(); {}
uchar ch;
uchar cha;
unsigned char word1[16]={"Cha:  Val= "};   // Display Cache
bit read_flag= 0 ;
sbit dout=P1^0;
sbit sclk=P1^1;
sbit cs = P1^2;
sbit shdn=P1^3;
```

Fig. 12. MCU and its peripheral circuits

U10

4051
4051

U11

U12A
74LS04

MAX1240

U9

8052

U8

U19

MAX232/SO

IN9 IN10 IN11 IN12 IN13 IN14 IN15
IN8 IN7 IN6 IN5 IN4 IN3 IN2 IN1

X X0 X1 X2 X3 X4 X5 X6 X7
A B C EN VDD VEE
VEE VDD

+3.3V
C9 1nF
C6 1nF

VDD CS SHDN SCLK REFIN ANAIN SDO

PSEN ALE/P P2.7 P2.6 P2.5 P2.4 P2.3 P2.2 P2.1 P2.0
P1.7 P1.6 P1.5 P1.4 P1.3 P1.2 P1.1/T2X P1.0/T2
RST EA/VP X2 X1 WR TXD T1 T0 RXD RD INT1 INT0
P0.7 P0.6 P0.5 P0.4 P0.3 P0.2 P0.1 P0.0

OUT4 OUT3 OUT2 OUT1

R22 1k
C14 22uF
C10 30pF
Y1 14.7456MH
C13 30pF
+5V

R1OUT R2OUT T1OUT T2OUT
R1IN R2IN T1IN T2IN
C1+ C1- C2+ C2- V+ V-

```
/***********************1ms Delay Functions ********************/
delay1ms(uint t)
{
        uint i,j;
         for(i=0;i<t;i++)
          for(j=0;j<120;j++);
}
/******************* MAX1240 conversion process *******************/
void adconversion(void)
{ uint rice=0;
  uchar i;
float rice1=0;
 sclk=0;
 cs=0;
_nop_();
if(dout==0)
{_nop_();
_nop_();
_nop_();
_nop_();
_nop_();
_nop_();
_nop_();

if(dout==1)
        {
         for(i=0;i<12;i++)
         { sclk=1;
          _nop_();
          sclk=0;

    if(dout==1)
        {rice=(rice<<1)|0x0001;}
         else
         rice=rice<<1;
          }
          cs=1;}
        }
word1[4]=(cha/10)+48;
word1[5]=(cha%10)+48;
rice1=(float)rice*1.220703125;//(2.5*rice*1000)/0xfff;
rice=(uint)rice1;
word1[11]=(rice%10000)/1000+48;
word1[12]='.';
word1[13]=(rice%1000)/100+48;
word1[14]=(rice%100)/10+48;
word1[15]=rice%10+48;
}
```

```
/*************** Serial port initialization process***************/
void UART_init()
        {
                TMOD=0x20;                      // timer 1  mode 2 : 8-Bit reload
                PCON=0x00;                      // Baud rate is not doubled
                SCON=0x50;                      // mode 1 bit UART, enable receiver
                TH1=0xFC;                       // reload value 9600 baud
                IE = 0x90;          //Enable Serial Interrupt
                TR1=1;                  //timer1 run
        }
 /************* Serial communication function (send)*********************/
void putchar(unsigned char data1)  // Characters send function
        {
         SBUF = data1;
        while(TI == 0);
        TI = 0;
        }

void putstring(unsigned char *dat)   // string Send function
        {
         while(*dat != '\0')
        {
         putchar(*dat);
         dat++;
         delay1ms(5);
        }
        }
void Serialsent()                  // Send a string of specified bits
        {
        unsigned char c;
        for(c=11;c<16;c++)
                        {
                        SBUF=word1[c];
                        while(!TI);
                        TI=0;
                        delay1ms(10);
                        }
        }
/************* Serial communication function (receiver)*****************/
// Serial port receive interrupt function
void serial () interrupt 4 using 3
{
   if (RI)
        {
                RI=0;
                ch=SBUF;
```

```
                read_flag= 1 ;
                P0=ch;
                delay1ms(1000);
            }
}

/***********************Main function *************************/
void main()
{
        UART_init();                          // Serial port initialization
            do
            {
            P2=cha;
            adconversion();
            delay1ms(100);
            putstring("\r\n") ;
            Serialsent();
            delay1ms(50);
            cha++ ;
            }
        while(cha<15);

        putstring("\r\n");
        putstring("-----------\r\n");
        delay1ms(1000);
if (RI)    // If have received interrupt, get results and send it to P0
 {                  RI=0;
                ch=SBUF;
                read_flag= 1 ;
                P0=ch;
                delay1ms(1000);
            }
}
```

3.3 Pump control circuit

Pump control signal first use an inverter 74LS04 and a regional control circuit is shown in Fig.13. Then the other three regions control circuit is the same with the first pump control circuit.

3.4 Power supply module
3.4.1 Analysis of system power demand

In this design, the pressure signal conditioning circuit requires two groups of +5V and -5V analog power, the microcontroller requires 5V power supply, the operational amplifier of conditioning circuit requires +12V and -12V. A/D converter and MAX3232 need 3.3V digital power supply, while pump requires 12V power. To suppression the interference generated by digital circuit to the analog signal, the digital ground and the analog ground need to be isolated.
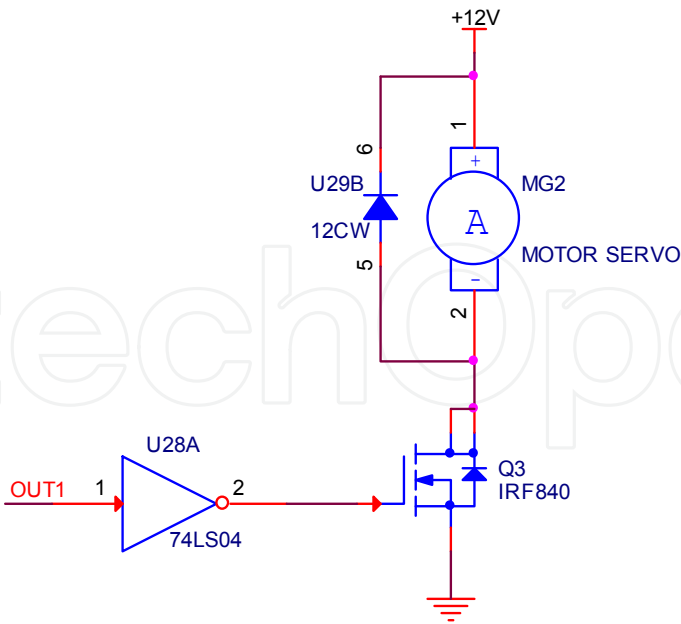
Fig. 13. Pump control circuit

According to the system design requirements and evaluating various factors, initially setting the basic parameters of the system power is as follows:

- Input Voltage: 12V AC voltage;
- Output voltage: +12 V,-12V, +5 V,-5V, +3.3 V;
- Output current: the major components required normal currents were several to tens of milliampere, so that the power supply current typically is less than 100mA.

3.4.2 Design of digital power

In order to reduce the power supply of system and improve power efficiency and reduce heat of the power system, low dropout regulator should be chosen. This design choice RT9163 series regulator to provide the required voltage, the main features of the chip are as follows:

- LDO: maximum voltage 1.4V;
- Fast transient response;
- ± 2% of total output regulation;
- 0.1% line regulation;
- 0.1% load regulation;
- Output Current: 0.5A;

+5V digital and analog power supply circuit is shown in Fig.14(a), output voltage of three-terminal regulator RT9163 is used as the +5V digital power supply, then the inductance L1 and capacitor Ct8 constitute of the LC low-pass filter, cutoff frequency is:

$$f_c = \frac{1}{2\pi\sqrt{L_1 \cdot C_2}} \approx 400Hz \tag{9}$$

Digital voltage filters high frequency noise by the LC filter, which can be used as analog power VCC. +5 V digital power supply followed by a three-terminal regulator RT9163 of 3.3V output is shown in Fig.14(b) and available +3.3 V digital power supply can be got.

(a) +5 V digital power supply   (b) +3.3 V digital power supply
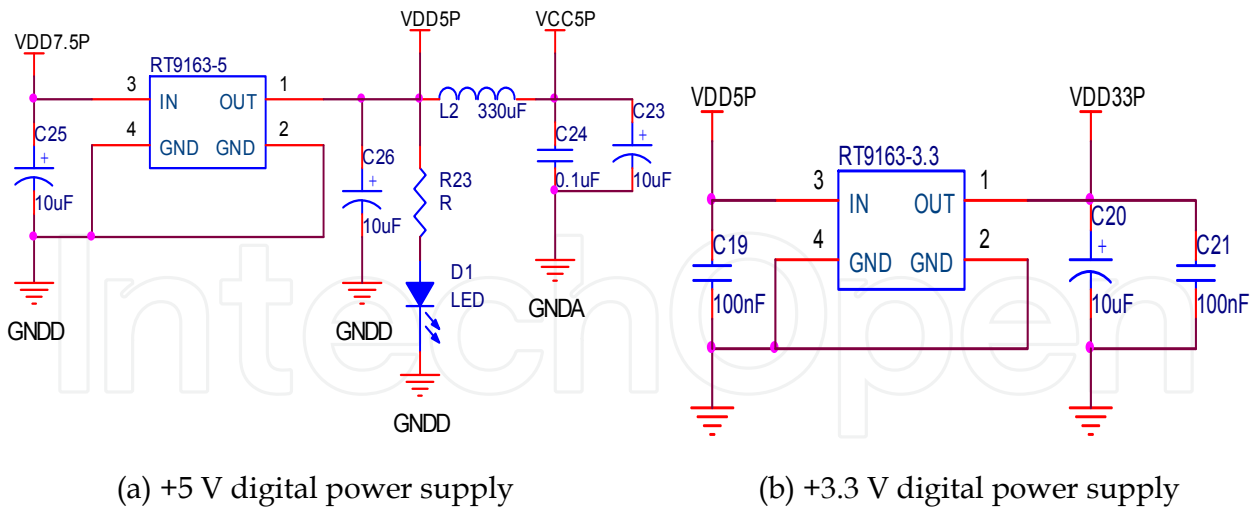
Fig. 14. Digital power supply

3.4.3 Analog power supply

Analog power supply made the multi-level regulation of the three-terminal regulator block, and large-capacity electrolytic capacitors and high-frequency small capacitors make output to meet the voltage requirements. The circuit is shown in Fig.15.
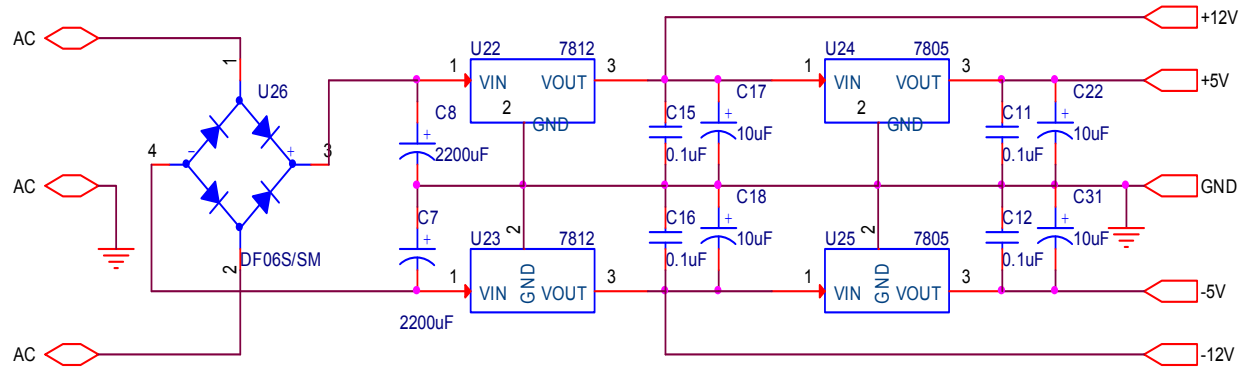
Fig. 15. Analog power circuit

4. LabVIEW virtual instrument design

The main thought of LabVIEW virtual instrument design is: LabVIEW virtual instrument recieves the pressure data of each sensor which sent from microcontroller by serial port, then the pressure data is transformed, arranged and filtered. At last, it is divided into four groups and each group of data represents the press state of head area, back area, buttocks area and crus area. The sensor signals in each area of mattress are shown in real time in different waveform chart, and then be saved in the corresponding file named by the user. Then LabVIEW calls the data saved in the excel file and analyzes them, if the point bearing the most in certain part does not change during the setting time, the system will recognize that the skin in this part has been pressed for a long time and it will easily cause bedsore, then the LabVIEW virtual instrument sends control signals to microcontroller and begins to alarm, microcontroller controls pump to massage this part of skin, its flow chart is shown in Fig.16.
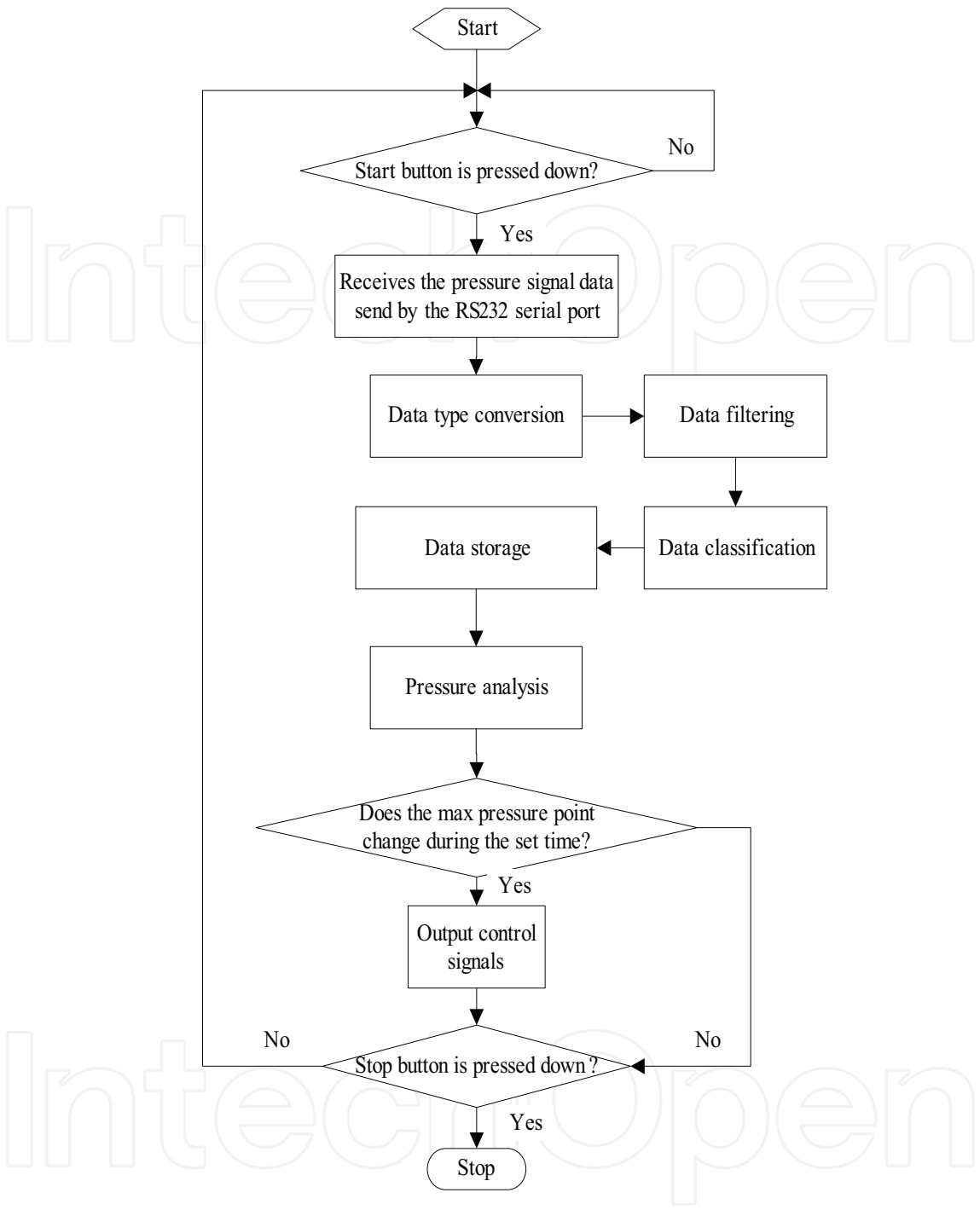
Fig. 16. LabVIEW main program flow chart

First, open National Instruments LabVIEW 2009 software, then choose "Empty Project" in the left side of the "Getting Started" window to build a new project, at last name the project with "Design of a Intelligent Bedsore Prevention and Treatment System".

4.1 Data receiving and storage

Data receiving and storage progam is composed with four parts: read serial port, data transformation and arrangment, data filtering and area saving, its concrete program block diagram is shown in Fig.17.
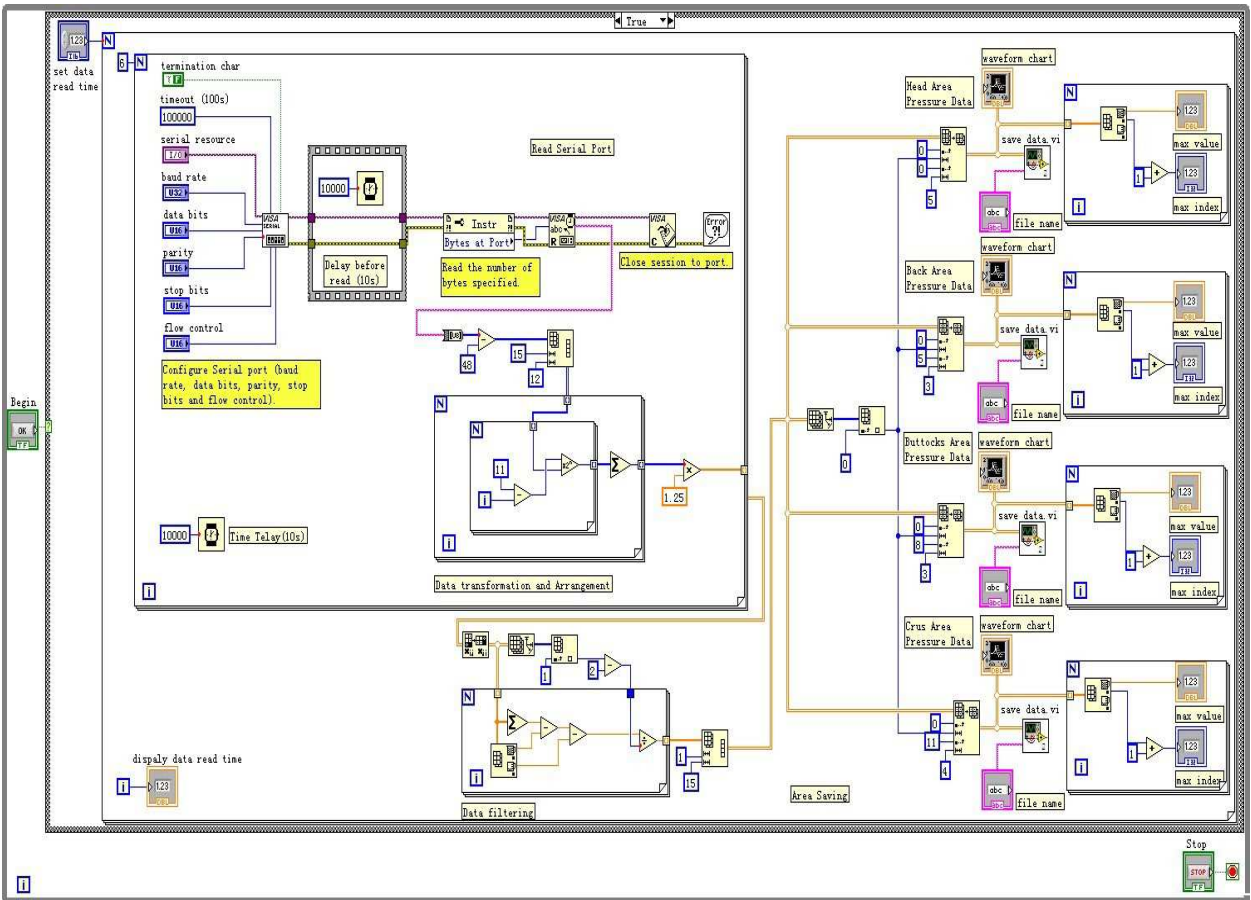
Fig. 17. Data read and storage program block diagram
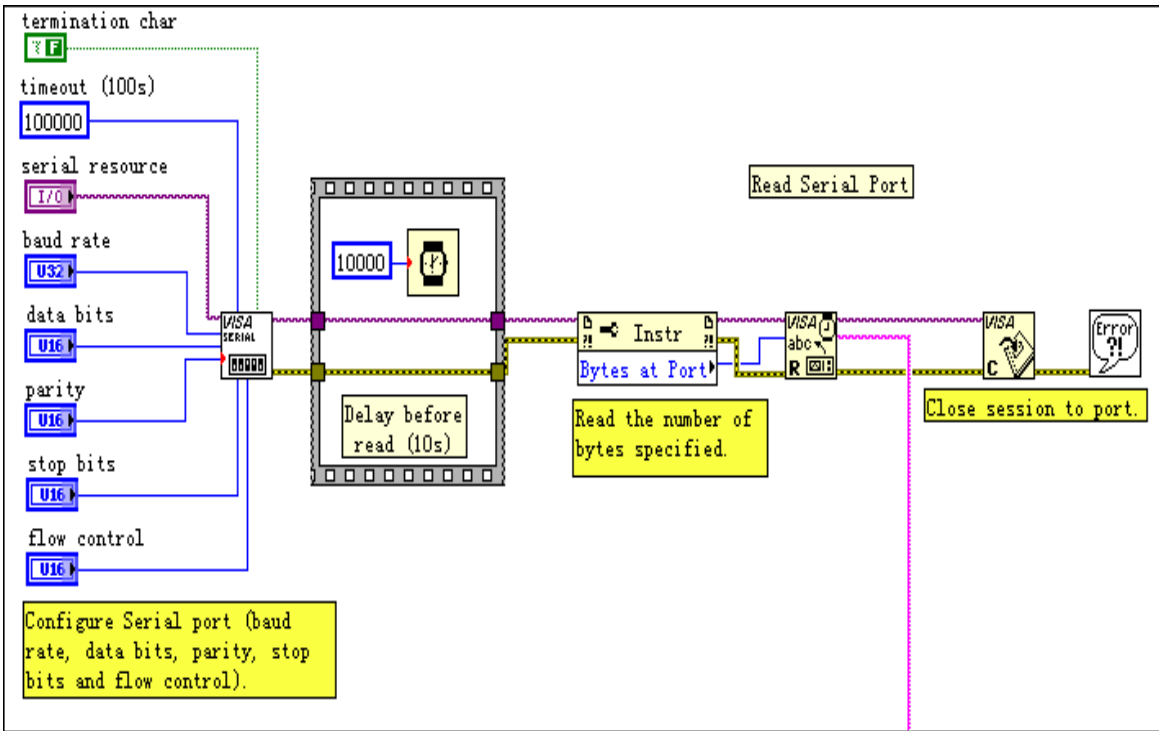
Fig. 18. Read serial port program block diagram

Concrete steps are as follows (Chen, 2007):

1. Build a new program in this project and name it with "data read and storage.vi".
2. Read serial port: the data communications are based on the RS232 serial port protocol. The serial port should be initialized before the data communications, the settings of serial port in the software and the hardware should be consistent, and its block diagram is shown in Fig.18. VISA Configure Serial Port VI is used to initializes the serial port, the parameters usually want to be configured are: baud rates, data bits, stop bits, flow control and so on. VISA Read Function is used to read the specified number of bytes from the serial port and return the data in read buffer, these functions are in the Instrument I/O|Serial Palette of the Functions Palette.
3. Data transformation and arrangement: in the LabVIEW virtual instrument, the type of data read from the serial port is binary string, so it should be converted to binary integer firstly, we use String To Byte Array Function to convert a string into an array of unsigned bytes, but each byte in the array has the ASCII value of the corresponding character in the string, so its value should decreases 48. All the converted sample data composes a binary one-dimension array with a length of 180n(12x15xn), we should transform it to a denary two-dimension array with fifteen columns and n rows, and the data in the same column is the sampled data of one sensor. Reshape Array Function is used to change a one-dimension array to a two-dimension array, after use this function, a one-dimension array with a length of 180 is changed to a two-dimension array with 12 columns and 15 rows. Then we should convert the binary data to the denary, change the binary data to decimal data in each row accoring to the binary to decimal conversion formula. After series of data transformation and arrangement, the data is transformed into a denary two-dimension array with n rows and 15 columns. But the data is not the pressure data we need at last, it is the data after A/D conversion, the relation between this data and the real pressure data is linear, so the data should be multiplied by a factor, the value of the factor is related to measurement circuit and A/D conversion circuit. Data transformation and arragement block diagram is shown in Fig.19.
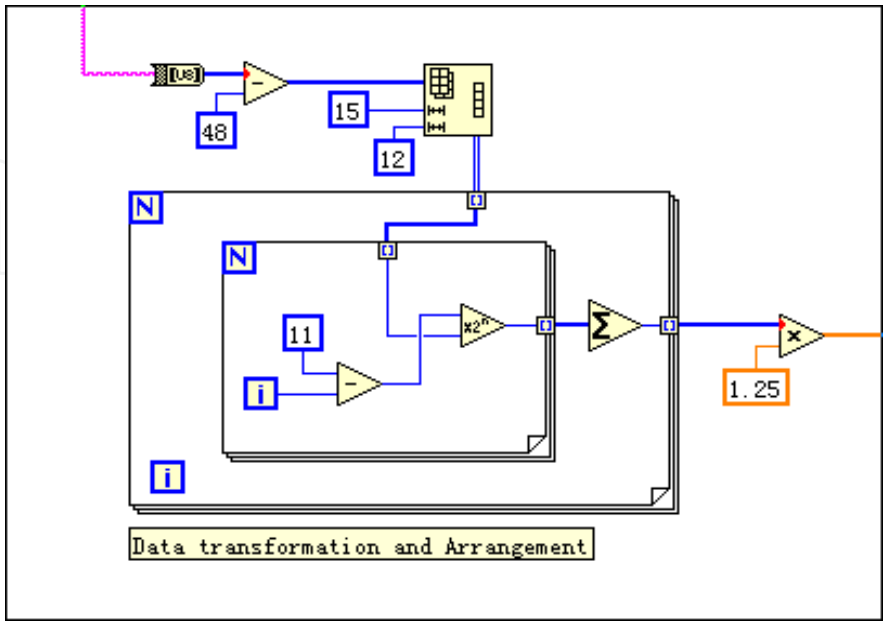

Fig. 19. Data transformation and Arrangement program block diagram

4. Data filtering: data transmission errors and data loss is likely to occur during the RS232 serial communications, so the data need to be filtered. The main thought of data filtering is: LabVIEW virtual instrument recieves a group of data which sent by the RS232 serial port every 10 seconds, so it will receive 6 groups of data in a minute. Removing the maximum and minimum values of the 6 groups, the average value of the remaining 4 groups is thought as the pressure data in one minute, its block diagram is shown in Fig.20.
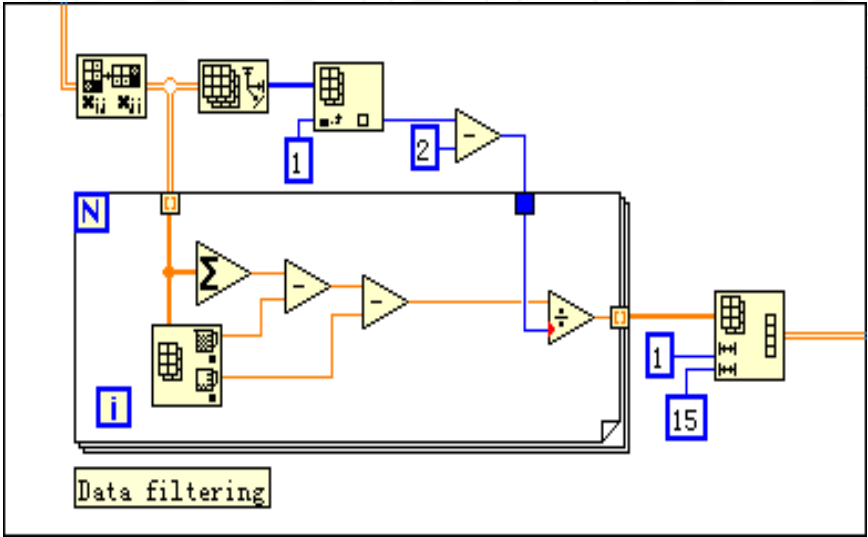
Fig. 20. Data filtering program block diagram

5. Area Saving: because the pressure analysis in each area is only relative to the measured pressure signals in the areas, the sensor signals can be saved by areas to different files. The column number of the array in each file is the same to the number of sensors in the corresponding area. That is to say, the two-dimension array will be divided into four new arrays with the row number invariable. We use Array Subset Function to divide the array, after using this function, 1~5 columns, 6~8 columns, 9~11 columns and 12~15 columns of the array will form 4 new arrays with the row number invariable. This 4 new arrays are pressure data of sensors in head area, back area, buttocks area and crus area. The sensor signals in each area of the mattress can be shown in real time in different waveform chart. Array Max & Min Function is used to show the maximal pressure of the area and the sensor number is under the maximal pressure, the program block diagram is shown in Fig.21.

4.2 Data storage

In Fig.21, the 'save data. vi' block is a LabVIEW subroutine, its function is to save the new two-dimension array of each area, which block diagram is shown in Fig.22. The pressure signals in each area will be saved into four different excel files. Write To Spreadsheet File VI is used to save the data into excel files, the intraday two-dimension array in each area can be save by entering the corresponding file name in the front panel of the subroutine, save path is "d:\file\(date)(file name).xls".

Concrete steps are as follows:

1. Build a new program in the project, and name it with "data storage.vi";
2. Draw program block diagram is shown in Fig.22;

3. Edit icon and connector. Double click the icon in the upper right corner and edit it, right-click the icon pane in the front panel and choose "Show Connector" in the shortcut menu. And then connect the controls in the front panel with the terminal of the connector. One input terminal should be connected to the array which needs to be saved, the other input terminal should be connected to the file name which wants to be saved.
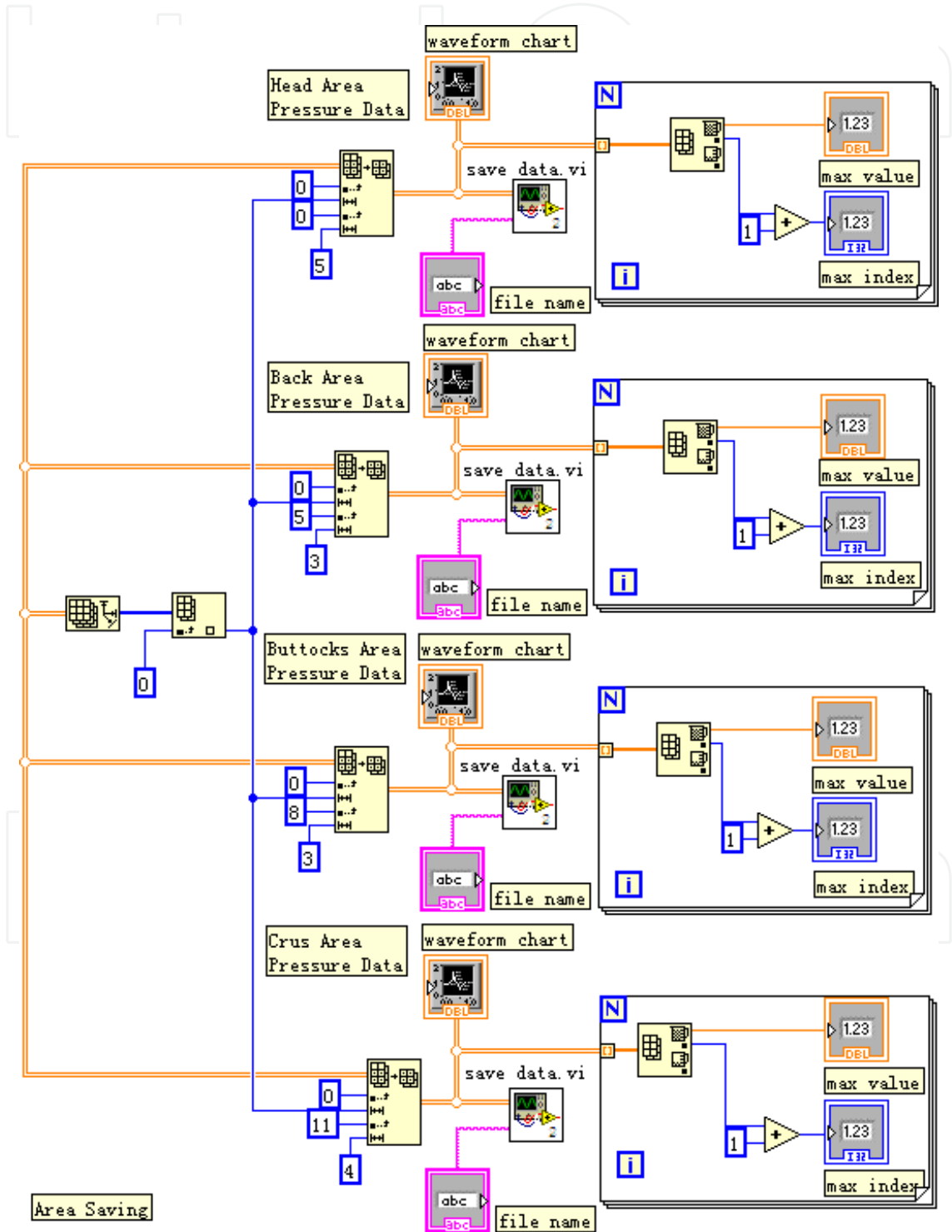4. Save this VI file, so we can use it next time.

Fig. 21. Area saving program block diagram
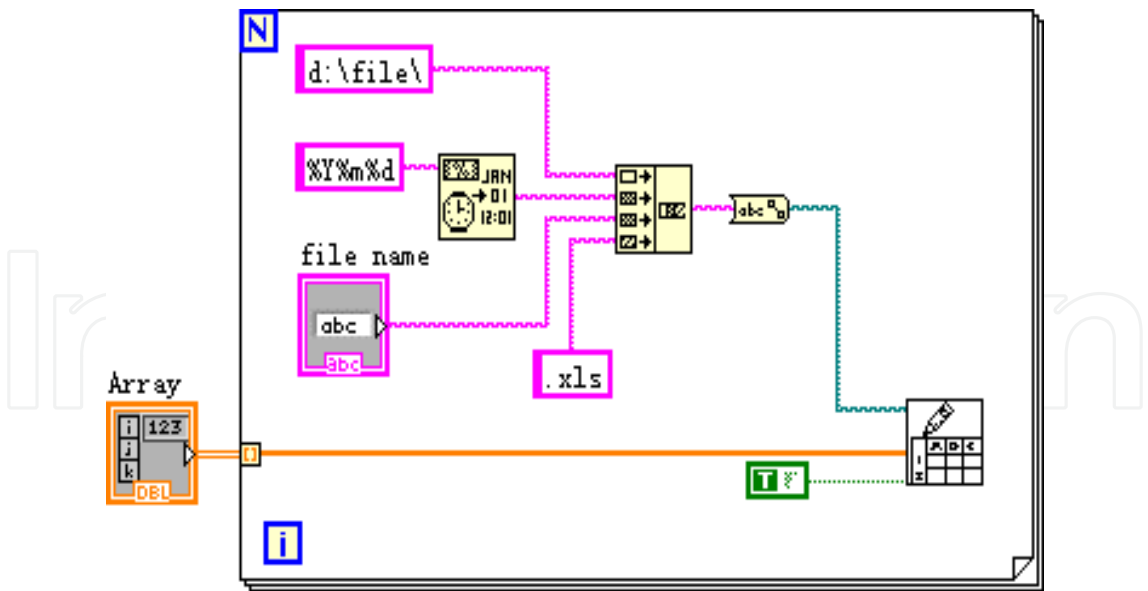
Fig. 22. Data storage block

4.3 Pressure analysis in each area

The press state of each area is got by comparing and analyzing the saved pressure data. The pressure analysis of the single area can be taken as an independent LabVIEW subroutine, the concrete program block diagram is shown in Fig.23.
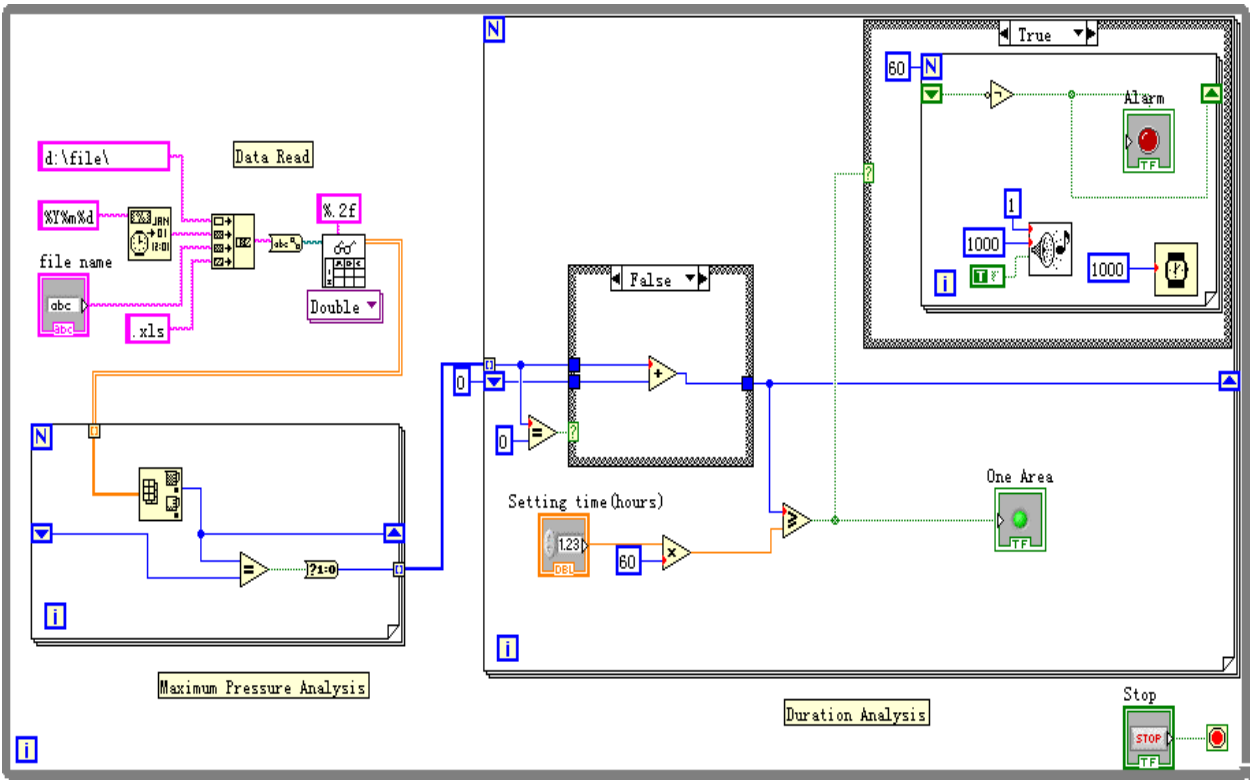
Fig. 23. Pressure analysis program block diagram

The point under the biggest pressure in the certain area of mattress can be got by comparing the corresponding data in this area. The maximum value in a row is the maximal pressure of the area at the certain time and the position of the maximal value can show the sensor which is under the maximal pressure. If the point bearing the most in certain part does not change during the set time, the system will recognize that the skin in this part has been pressed for a long time and need to be massaged. Otherwise, the system will recognize that the main bearing point in this area has been changed and this area do not need to be massage during the time (Zhou, 2008).

The maximum duration under pressure in each part of human body is different, because the maximum pressure in each part is different. In addition, the weight of patient is always different, thus the maximum duration should be set according to the characteristic of the patient and the specific part. If the sensor signals are set to be sampled every 10 seconds and stored every 1 minute, the maximum duration is set to be x hours. Then when the maximum values in all the seriate 60x rows are in the same column, it can be concluded that there is no move in this area during the x hours. The LabVIEW instrument send relevant signals to the microcontroller according to the analysis result, then the pump of certain area will be controlled to run (Yang, 2007).

Concrete steps are as follows:

1. Build a new program in the project, and name it with "pressure analysis.vi".
2. Data read: its block diagram is shown in Fig.24. Read From Spreadsheet File VI is used to read the saved pressure data files, and then output a double-precision two dimension array, the file path is "d:\file\(time)(file name).xls".
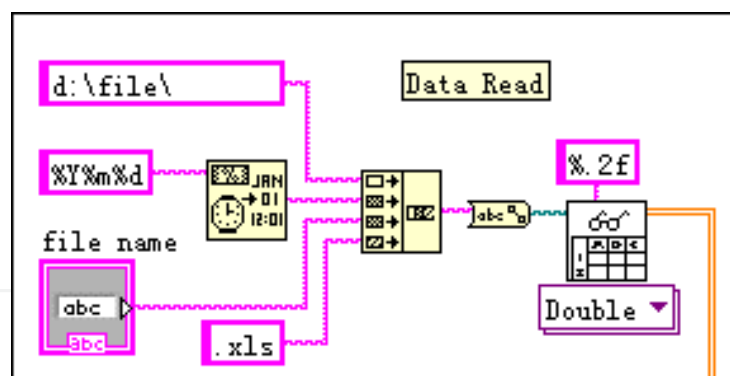

Fig. 24. Data Read program block diagram

3. Maximum pressure analysis: The array of pressure data can be read row by row, Array Max & Min Function is used to find the position of the maximal value in this row, then compare it with the position of maximal value in last row, if they are same, it will output Boolean variable '1', its program block diagram is shown in Fig.25.
4. Duration Analysis: its program block diagram is shown in Fig.26, if the position of the maximal value in this row is same to the position of the maximal value in last row, the value of register variable will increase 1. If the value of the register variable is larger than the setting time, it will output display signals and control signals, and it will begin to alarm.
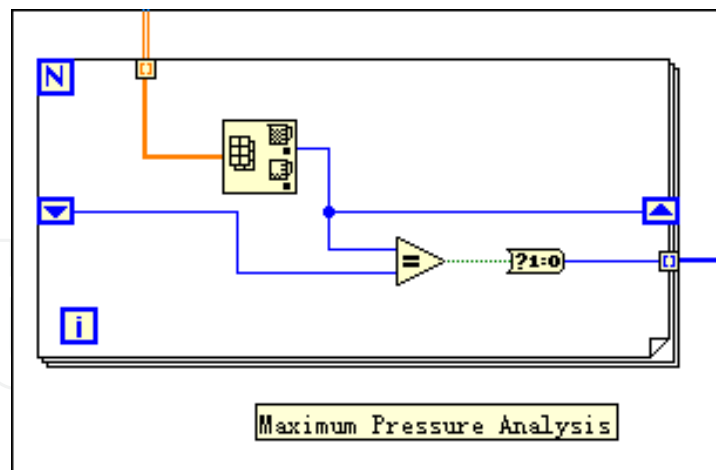
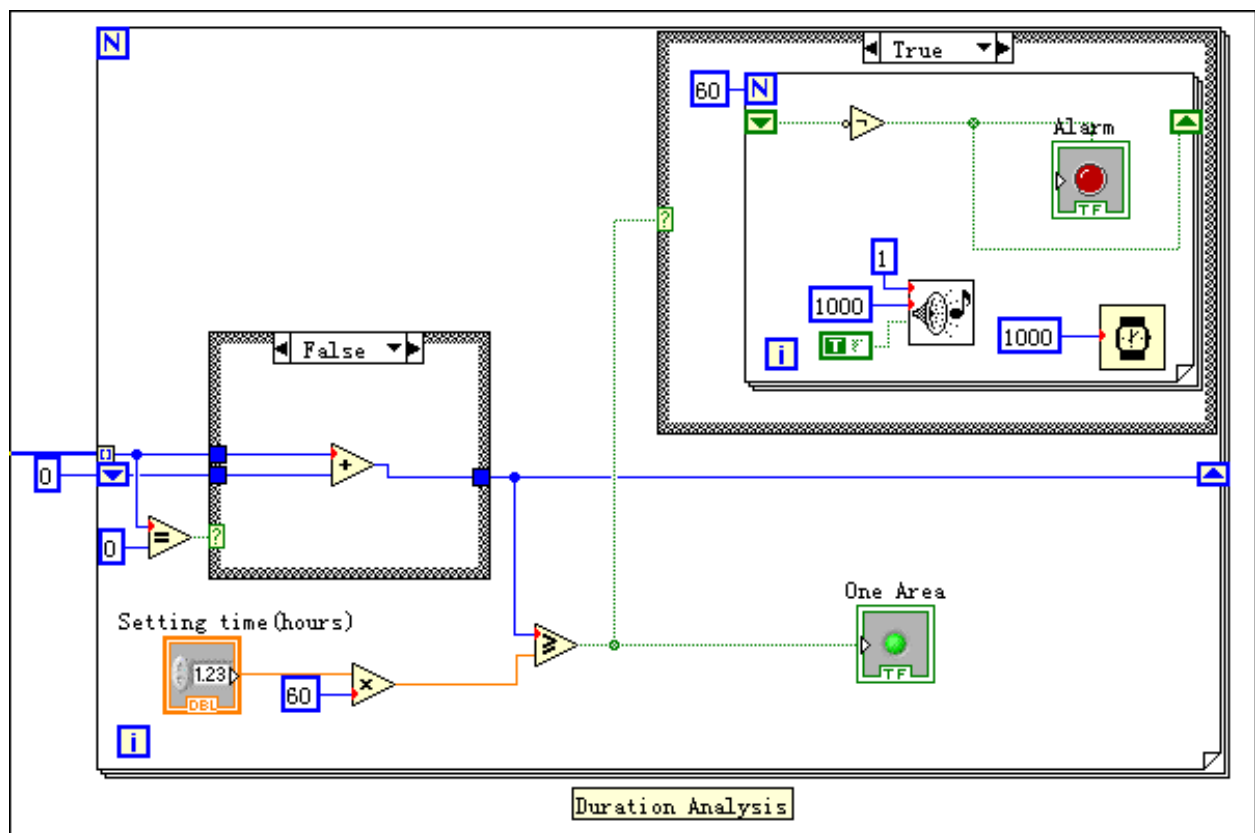Fig. 25. Maximum Pressure Analysis block diagram

Fig. 26. Duration analysis program block diagram

4.4 Control signals sending

We will send a 4 bytes string to microcontroller according to the result of pressure analysis in each area, each byte of the string indicates wheather the pump in this area begin to run, for example, a string of "0011" indicates that the pumps in buttocks area and crus area will begin to run, its diagram is shown in Fig.27.
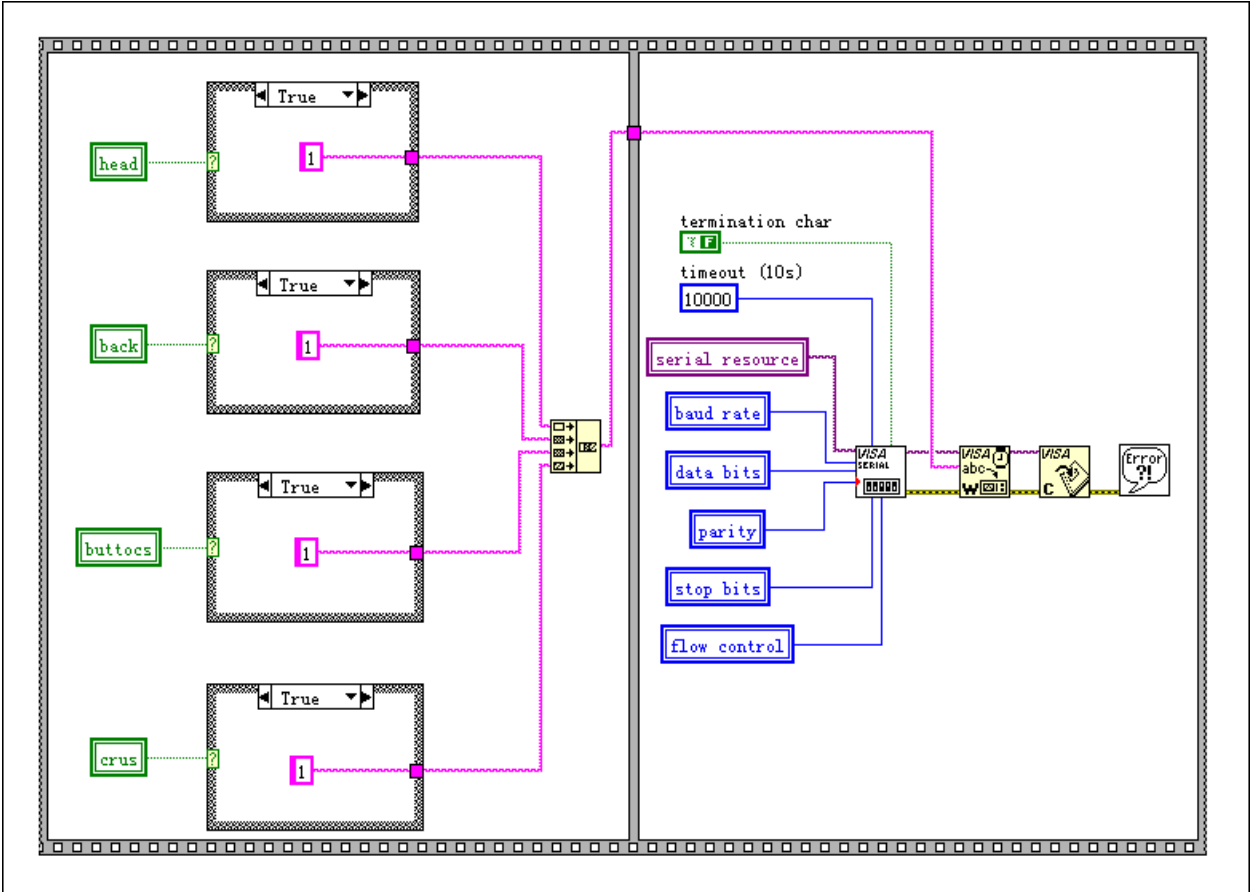
Fig. 27. Control signals sending block diagram

4.5 Instrument interface and function verification

The designed instrument interface is shown in Fig.28.The sensor signals in each area of the mattress are shown in real time in different waveform chart, and then be saved in the corresponding file named by the user. Saved pressure data file of different area is shown in Fig.29. The maximum duration under pressure of each area can be set by doctor. The alarming is visual and the status of different areas can be shown.

The system is set as in Fig.28, the signals in the waveform chart of buttocks area shows that the point bearing the most in this area has been changed during the past two hours, so there is no alarm. Alarm in the head area and the back area occurred after 3 hours, because the signals in these areas changed slowly during the time. Although the point bearing the most in crus area does not change after 3 hours, but its duration is less than the setting time (4 hours), so there is no alarm in this area either, its waveform chart is shown in Fig.30.

5. Conclusion

By analyzing the multi-channel sensor signals, the human body areas that need to be massaged can be recognized automatically. The system is designed considering the practicality, and it can make the patients having a better rest, so that the purpose of the bedsore prevention has been achieved. For the function of LabVIEW virtual instrument is easy to extend, we can perfect the system gradually in the clinic experiments.
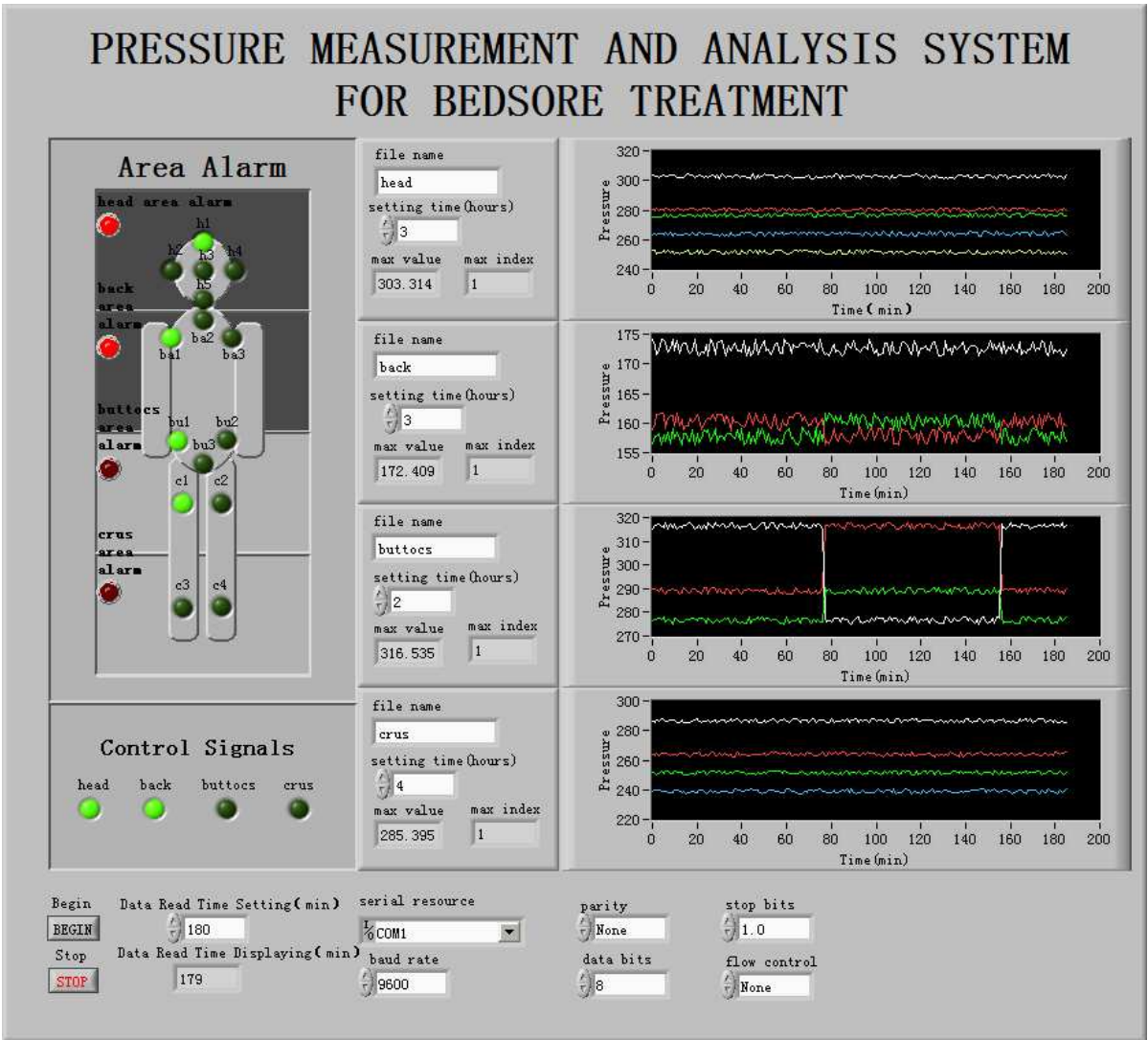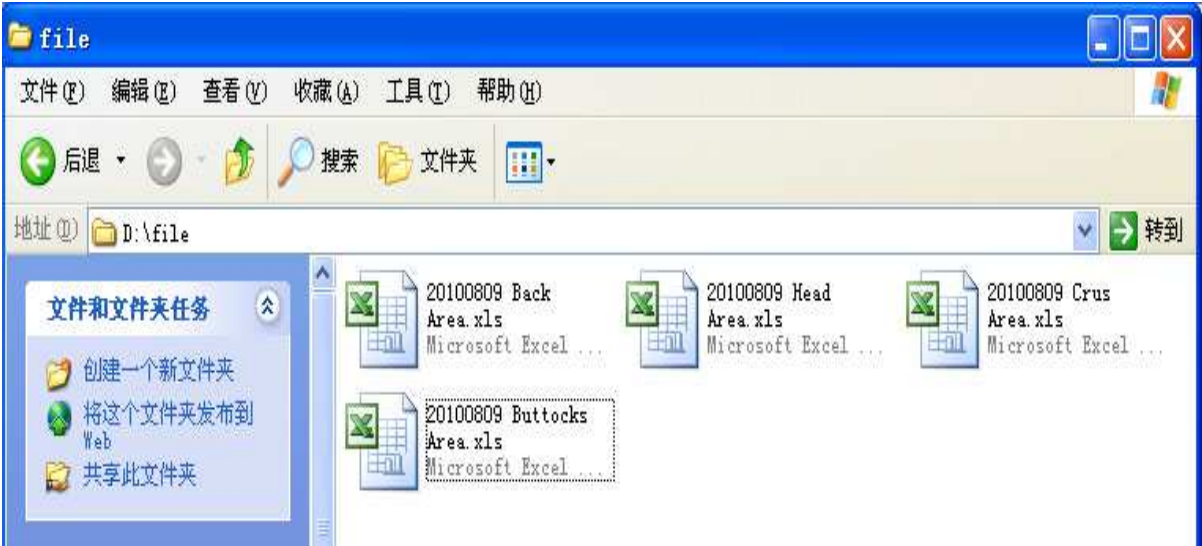
Fig. 28. Instrument interface
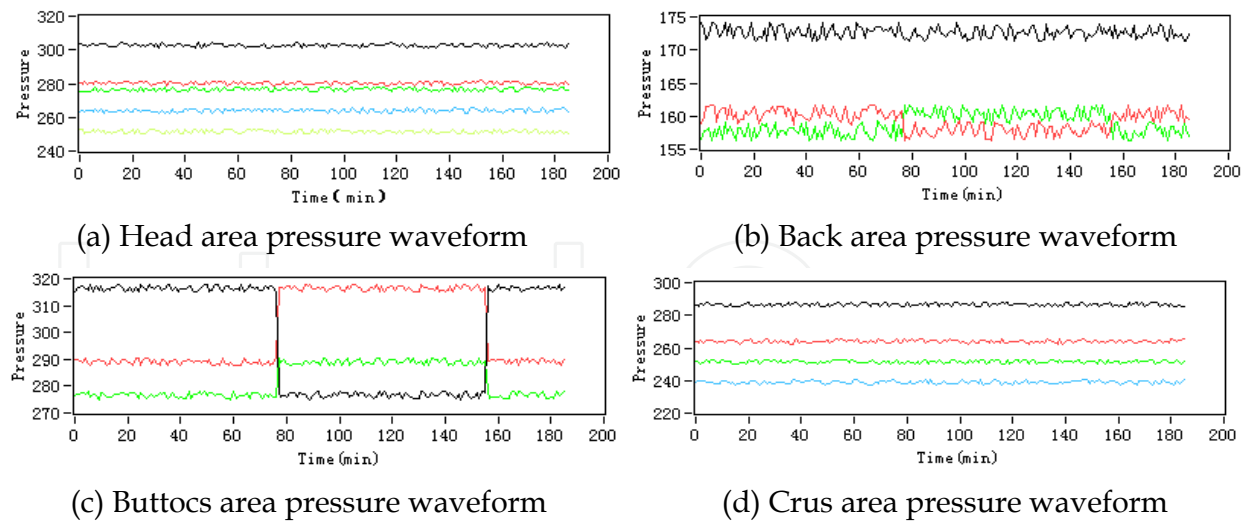

Fig. 29. Saved pressure data file of different area

(a) Head area pressure waveform

(b) Back area pressure waveform

(c) Buttocs area pressure waveform

(d) Crus area pressure waveform

Fig. 30. Waveform chart of pressure signals in each area

6. References

Chen Xihui. & Zhang Yinhong. (2007). *LabVIEW 8.20 Programming Design from Introduction to Master*, Tsinghua University Press, 978-7-302-15230-9, Beijing

Hu Hancai. (2003). *Microcontroller Theory and Interface Technology*, Tsinghua University Press, 7-302-07737-1, Beijing

Yang Leping et al. (2007). *LabVIEW Advanced Programming Design*, Tsinghua University Press, 978-7-302-15674-1, Beijing

Zhou Runjing & Hao Xiaoxia. (2008). *Multisim&LabVIEW Virtual Instrument Design Technology*, Buaapress, 978-7-81124-319-2, Beijing

Zhou Runjing & Hao Xiaoxia. (2009). *Sensor and Detection Technology*, Publishing House of Electronics Industry, 978-7-121-08530-7, Beijing
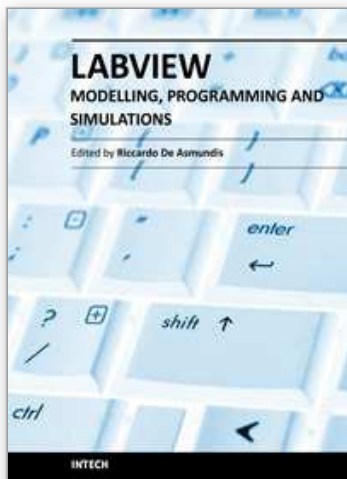
Modeling, Programming and Simulations Using LabVIEW™ Software

Edited by Dr Riccardo De Asmundis

Born originally as a software for instrumentation control, LabVIEW became quickly a very powerful programming language, having some characteristics which made it unique: simplicity in creating very effective User Interfaces and the G programming mode. While the former allows for the design of very professional control panels and whole applications, complete with features for distributing and installing them, the latter represents an innovative way of programming: the graphical representation of the code. The surprising aspect is that such a way of conceiving algorithms is extremely similar to the SADT method (Structured Analysis and Design Technique) introduced by Douglas T. Ross and SofTech, Inc. (USA) in 1969 from an original idea by MIT, and extensively used by the US Air Force for their projects. LabVIEW enables programming by implementing directly the equivalent of an SADT "actigram". Apart from this academic aspect, LabVIEW can be used in a variety of forms, creating projects that can spread over an enormous field of applications: from control and monitoring software to data treatment and archiving; from modeling to instrument control; from real time programming to advanced analysis tools with very powerful mathematical algorithms ready to use; from full integration with native hardware (by National Instruments) to an easy implementation of drivers for third party hardware. In this book a collection of applications covering a wide range of possibilities is presented. We go from simple or distributed control software to modeling done in LabVIEW; from very specific applications to usage in the educational environment.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Runjing Zhou (2011). Development of an Intelligent Bedsore Prevention System, Modeling, Programming and Simulations Using LabVIEW™ Software, Dr Riccardo De Asmundis (Ed.), ISBN: 978-953-307-521-1, InTech, Available from: http://www.intechopen.com/books/modeling-programming-and-simulations-using-labview-software/development-of-an-intelligent-bedsore-prevention-system

INTECH
open science | open minds

www.intechopen.com

Fax: +385 (51) 686 166
www.intechopen.com

Fax: +86-21-62489821