Study on Calibration Algorithm of Embedded Touch Screen

Li Ping-yong [1, 2] and Li Zhong-wen [1, 2]
1. College of Information Science and Technology, Chengdu University, Chengdu, China
2. Key Laboratory of Pattern Recognition and Intelligent Information Processing, Institutions of Higher Education of Sichuan Province, Chengdu University, China
Email: lpy999@163.com

Abstract—**This paper firstly analyzes causes and classification of the errors generated between touch screen and liquid crystal display (LCD). For the deficiencies of existing embedded touch screen calibration algorithm, five-point calibration algorithm (also known as multi-point mean calibration algorithm) is proposed. Then, the mathematical theoretical basis of the algorithm is introduced. Integer arithmetic instead of floating-point operations which can get better computing performance is proposed. Integer arithmetic can adapt to the characteristics and requirements of embedded devices. Meanwhile, deviation correction method is proposed to enhance the practicability of algorithm. Finally, the algorithm is implemented by program. Experiments show that the algorithm is accurate and has actual value.**

Index Terms—**Touch Screen; LCD; Five-point Calibration Algorithm; Integer Arithmetic; Deviation Correction**

I. INTRODUCTION

The touch screen, which is characterized with excellent firmness and durability, quick reaction, small space consumption and easy operation mode, is now widely used in human-computer interface of embedded system. Currently this technology shows a tendency which develops in the direction of PC. The touch screen is an absolute coordinate system whose point can be selected directly, which is essentially different with a relative positioning system (such as a mouse). It can either serve as a display terminal for input or an input device. There are many kinds of touch screen which can be divided into resistive, capacitive, surface acoustic wave and infrared scanning etc [1]. The resistive touch screen is the most popular. Its principle is to obtain the coordinates of contact point by measuring the resistance value of the horizontal and vertical.

The resistive touch screen is generally constituted by 4 layer transparent film. Bottom-up are: glass substrate, the lower ITO conductive layer, the upper ITO conductive layer and the plastic layer [2]. Between the upper and lower ITO conductive layers separated by insulating transparent fine particles. When the touch screen surface without pressure, the upper and lower ITO conductive layer is in an open state. Once a point on the touch screen is pressed, the upper and lower layers will be on conductive. When a different point is pressed, the resistor value of the point to the output terminal is also different. Thus, system outputs voltage value (analog) of the point which is corresponding to the position. Then the point's XY coordinate values can be obtained after A / D conversion.

Touch screen processing data is a physical coordinate of point, which is obtained through the acquisition of the touch screen controller; while LCD display (LCD) processing object is pixels. For example, the resolution is usually said 240×320. In fact, it means that the width of each line is 240 pixels, height 320 pixels. Accuracy of the touch screen is determined by number of bits of A/D converter. Its number of bits is 8, 9, 10 or more. Generally, the higher the accuracy, the better the effect. However, such a device that touch-screen is installed on the surface of LCD does not necessarily require a very high precision A/D converter [3]. LCD has its own fixed resolution. Such as, for a 240×320 resolution LCD, a maximum of 1/320 precision is enough. So, it at most requires a 9-bit A / D converter. In fact, 1/320 precision requirement is probably not necessary for actual application. For example, button that often appears in application program interface is a large rectangular area containing a number of points. The distance between any two button is also great, so it is not necessary such a high accuracy. It can be seen that the touch screen and LCD are two different physical devices. A certain convert is required between the both. Secondly, when the touch screen was installed on the LCD, inevitably there are some errors, such as rotation, translation. This situation is resolved by calibration. Further, the resistive touch screen material itself is different. And as time goes on, parameters will change [4]. Therefore frequent calibration is also required. The above factors lead to deviation of coordinates of point between touch screen and LCD. In this case, the user may not properly operate user interface on the LCD via the touch screen. As such, the touch screen shall be calibrated before putting into operation.

II. ANALYSIS OF TOUCH SCREEN CALIBRATION ALGORITHM

A. Analysis of Errors Generated between Touch Screen and Liquid Crystal Display (LCD)

Although errors generated between touch screen and

doi:10.4304/jmm.9.4.605-610

liquid crystal display are caused by a number of factors, those errors can mainly be attributed to two factors: the first is that the touch screen and LCD are different physical elements, so resolution error will naturally arise, as shown in Figure 1(a). As shown in Figure 1(a), Figure 1(b) and Figure 1(c), point P(X, Y) is the coordinate of the touch panel coordinate system, while the point P corresponds to the point P' is the LCD coordinate system coordinate. As shown in Figure 1(a),

$X' = K_X \times X$, K_X is the ratio of the X-direction resolution between LCD and touch screen.

$Y' = K_Y \times Y$, K_Y is the ratio of the Y-direction resolution between LCD and touch screen.

Generally, resolution of touch screen and LCD are arrangement of rule. So, K_X and K_Y can be seen as a linear.

The second factor is that, mechanical errors will be generated after the touch screen is installed on the LCD, which includes displacement error and rotation error, as shown in Figure 1(b) and (c).

As shown in Figure 1(b),

$X' = X + \Delta X$, ΔX is the offset amount of touch screen in the X-direction.

$Y' = Y + \Delta Y$, ΔY is the offset amount of touch screen in the Y-direction.

As shown in Figure 1(c),

$X = R \times \cos\theta$, R is the distance of between the point $O(0,0)$ and the point P, and θ is offset angle that touch screen is relative to the X-axis positive direction.

$Y = R \times \sin\theta$,

$X' = R \times \cos(\theta - \Delta\theta)$, R is the distance of between the point $O(0,0)$ and the point P, and $\Delta\theta$ is the rotation offset amount of between touch screen and LCD.

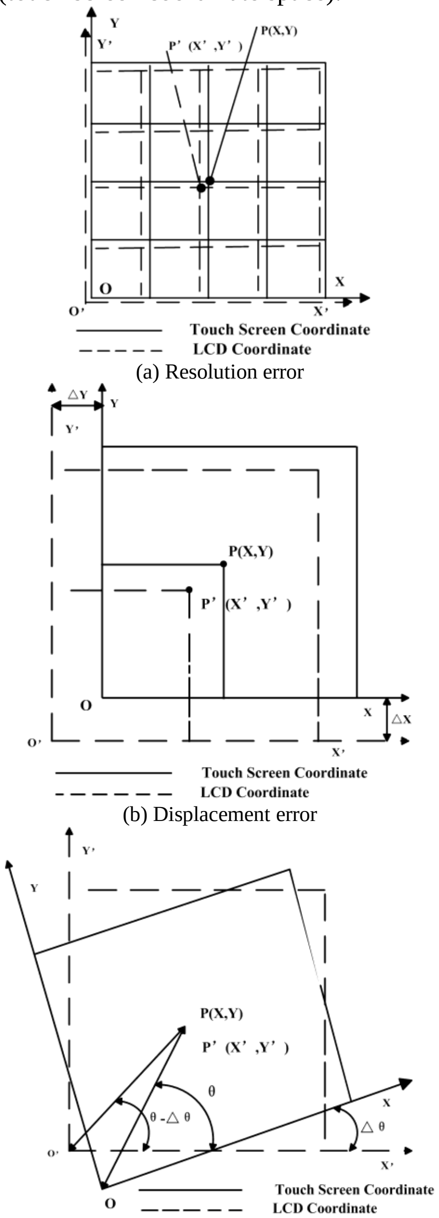
$Y' = R \times \sin(\theta - \Delta\theta)$

B. Mathematical Analysis of Calibration Algorithm

In order to solve the problem of inconsistency of coordinates of the same point on touch screen and LCD, a kind of linear calibration algorithm may be established between theoretical space and actual space. By thus doing, when we obtain the coordinates of a certain point by touch screen driver, the LCD coordinates can be obtained by means of linear calibration algorithm.

Touch screen calibration methods include: two-point calibration, three-point calibration and N-point calibration [5] and so on. The precision of the two-point calibration method is lower, while, it is higher [6] of the three-point calibration method and further improvement is quite possible. Generally speaking, the more points are collected, the higher the calibration precision will be. Nevertheless, too many points will be a cause redundancy, which is of little use for improvement of calibration precision but only increases the time for calculation. Therefore, the five-point calibration algorithm (also known as multi-point mean calibration algorithm) is adopted.

Select a rectangle in the touch screen coordinate space, and the center point (namely the fifth points) formed by the 4 vertexes and the 2 diagonals, which can segment the complete rectangle into 4 triangles, as shown in Figure 2, in which, dashed lines stand for theoretical triangle (LCD coordinate space), the solid line s stand for the actual triangle (touch screen coordinate space).



(a) Resolution error

(b) Displacement error

(c) Rotation error

Figure 1. Classifications of Errors Generated between Touch Screen and LCD

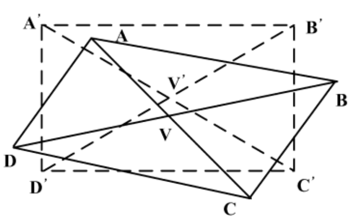


Figure 2. Linear Calibration Algorithm Model for Touch Screen

The three kinds of errors generated between touch screen and LCD are analyzed above, if the angle difference between LCD and touch screen is tiny, the error between the theoretical and actual space can be deemed as linear [7] and then correspondence of all points on touch screen and LCD can be established [8]. We assume that, the coordinates of all points on touch screen and LCD are completely corresponding, namely, coordinates of any point P_L in LCD is (X_L, Y_L), and coordinates of any point P_T in the touch screen is (X_T, Y_T), following lineal relations can be established:

$$\begin{aligned} X_L &= A \times X_T + B \times Y_T + C \\ Y_L &= D \times X_T + E \times Y_T + F \end{aligned} \tag{1}$$

where, A, B, C, D, E and F stand for coordinate conversion coefficient (also known as calibration coefficient). From the above formula, we can calculate the six coordinate conversion coefficients just by coordinates of 3 sample points on the touch screen and the corresponding coordinates in LCD.

Firstly, we choose a pair of triangles, for example, one actual triangle ADV (touch screen) and one corresponding theoretical triangle $A'D'V'(LCD)$, and substitute coordinates of the two triangles $A'D'V'(LCD)$, $D(X_{L1}, Y_{L1})$, $V(X_{L2}, Y_{L2})$; $A'(X_{T0}, Y_{T0})$, $D'(X_{T1}, Y_{T1})$, $V'(X_{T2}, Y_{T2})$) into formula (1-1). So, the following equations sets will be got:

$$\begin{aligned} X_{L0} &= A \times X_{T0} + B \times Y_{T0} + C \\ Y_{L0} &= D \times X_{T0} + E \times Y_{T0} + F \\ X_{L1} &= A \times X_{T1} + B \times Y_{T1} + C \\ Y_{L1} &= D \times X_{T1} + E \times Y_{T1} + F \\ X_{L2} &= A \times X_{T2} + B \times Y_{T2} + C \\ Y_{L2} &= D \times X_{T2} + E \times Y_{T2} + F \end{aligned}$$

The equations sets can be solved by Gauss elimination method. So, expression of these six coefficients can be obtained as follows:

$$\begin{aligned} A &= [(X_{L0} - X_{L2}) \times (Y_{T1} - Y_{T2}) - (X_{L1} - X_{L2}) \times (Y_{T0} - Y_{T2})] / K \\ B &= [(X_{T0} - X_{T2}) \times (X_{L1} - X_{L2}) - (X_{L0} - X_{L2}) \times (X_{T1} - X_{T2})] / K \\ C &= [Y_{T0} \times (X_{T2} \times X_{L1} - X_{T1} \times X_{L2}) + Y_{T1} \times (X_{T0} \times X_{L2} - X_{T2} \times X_{L0}) \\ &\quad + Y_{T2} \times (X_{T1} \times X_{L0} - X_{T0} \times X_{L1})] / K \\ D &= [(Y_{L0} - Y_{L2}) \times (Y_{T1} - Y_{T2}) - (Y_{L1} - Y_{L2}) \times (Y_{T0} - Y_{T2})] / K \\ E &= [(X_{T0} - X_{T2}) \times (Y_{L1} - Y_{L2}) - (Y_{L0} - Y_{L2}) \times (X_{T1} - X_{T2})] / K \\ F &= [Y_{T0} \times (X_{T2} \times Y_{L1} - X_{T1} \times Y_{L2}) + Y_{T1} \times (X_{T0} \times Y_{L2} - X_{T2} \times Y_{L0}) \\ &\quad + Y_{T2} \times (X_{T1} \times Y_{L0} - X_{T0} \times Y_{L1})] / K \end{aligned}$$

The six coefficients share a common denominator of K, whose value is as follows:

$$K = (X_{T0} - X_{T2}) \times (Y_{T1} - Y_{T2}) - (X_{T1} - X_{T2}) \times (Y_{T0} - Y_{T2})$$

Coordinate conversion coefficients of the rest three pairs of triangles can be obtained in a similar way, the four groups of conversion coefficients (four groups of transformation matrix) can be obtained. If actual coordinates of a point on the touch screen is given, 4 theoretical values in LCD can be obtained individually using the four groups of conversion coefficients, and then the 4 theoretical values may be averaged to obtain the only theoretical value.

C. Integer Implementation for Five-Point Calibration Algorithm

Based on the mathematical analysis above, touch screen calibration algorithm may be implemented by programming. In general, implementation of floating-point is more precise. However, a large number of floating-point operations are applied to embedded devices will lead to performance loss to a certain extent. It is not necessary to obtain a very precise calibration for touch screen. [9] For example, the user taps the screen on a certain point, whose theoretical coordinates are (20, 20), and the actual coordinates are (21, 21), such situation will not lead to a serious problem.

Based on the above analysis, there is simply no need to use floating-point operations which cause extra expenditure in touch screen calibration algorithm implementation for embedded devices, so, integer arithmetic is adopted for the intended purpose, which will yield better computing performance [10].

Compared to floating-point operations, integer arithmetic which is used for implementation of above algorithm possesses another 2 advantages:

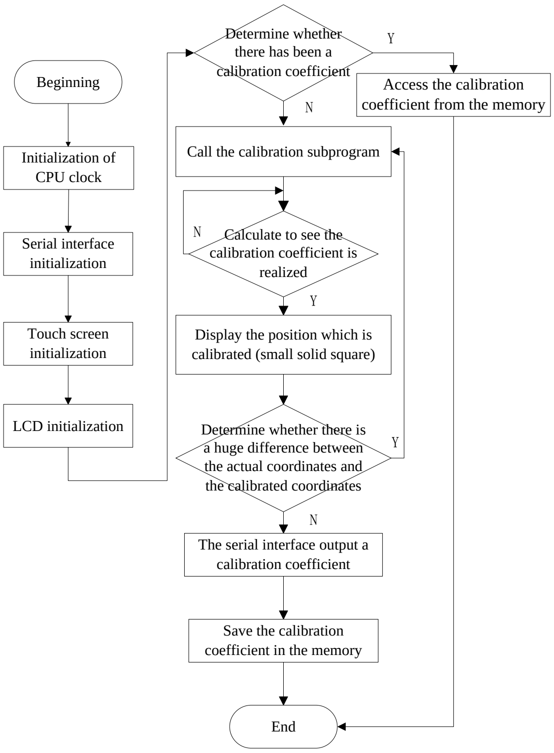
1) After 4 integer arrays are adopted for storing the four groups of coordinate conversion coefficients, half of the memory space is saved;

2) When the Gaussian elimination method is used for determining whether there is a solution, the denominator shall be compared with certain real number which is particularly low for floating-point implementation, while, we only need to determine whether the denominator is zero for Integer implementation

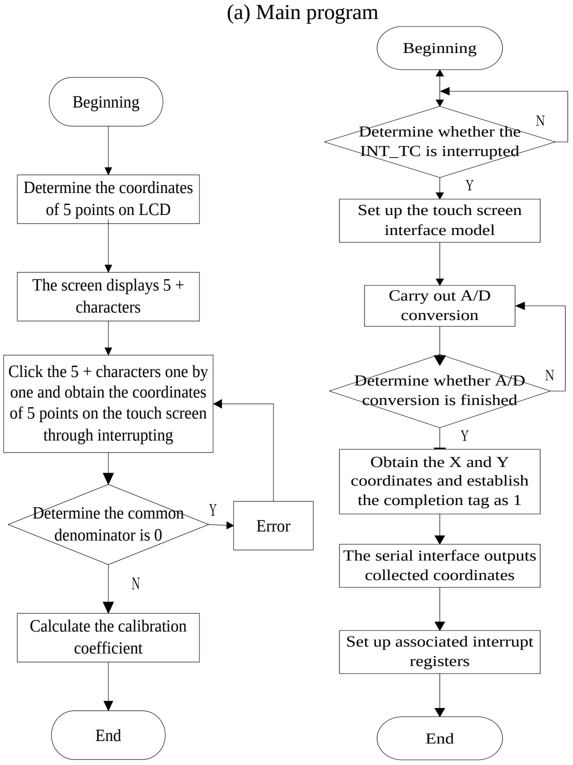
In order to further reduce the precision loss, we can left shift the calibration coefficient for n times (for example, 10 times) to realize the increase by 2n times and definitely, the final result shall be right shifted for narrowing by the same times.

D. Deviation Correction Method for Five-Point Calibration Algorithm

When the above calibration algorithm is used, calibration process is generally: tap the cross character in the touch screen after it is displayed to obtain the actual coordinates by touch screen driver [11]. If the point that the operator taps is rather far away form the cross character due to maloperation or other unexpected conditions, there will be a great difference between the obtained calibration coefficient and the actual one. LCD coordinates are far from accurate with such coefficient and even the user couldn’t use the touch screen normally. In order to solve the problem, coordinate conversion coefficients (or calibration coefficients) which are obtained for calibration can be used to calculate the LCD coordinates of one point on the touch screen after calibration is finished, if the calculated LCD coordinates can basically match the actual ones, the group of coordinate conversion coefficients may be stored for future calibration. And if the calculated coordinates deviate significantly from the actual ones, re-calibration is needed.



(a) Main program



(b) Calibration subprogram (c) Interrupt service subprogram

Figure 3. Program flow chart

III. PROGRAM IMPLEMENTATION OF CALIBRATION ALGORITHM

The procedures of touch screen calibration process are: firstly, display 5 points with fixed coordinates and then tap the 5 points one after another, then obtain the coordinates via touch screen driver and calculate the calibration coefficient by the calibration subprogram [12].

The complete calibration process is divided into six parts: initialization (including initialization of CPU clock, serial interface initialization, and touch screen initialization, LCD initialization), screen display, access of coordinates (by interrupt processing program), calculation of calibration coefficient, deviation correction, storage of calibration coefficient' which are composed of one main program and two subprograms.

Program flow is as shown in Figure 3.

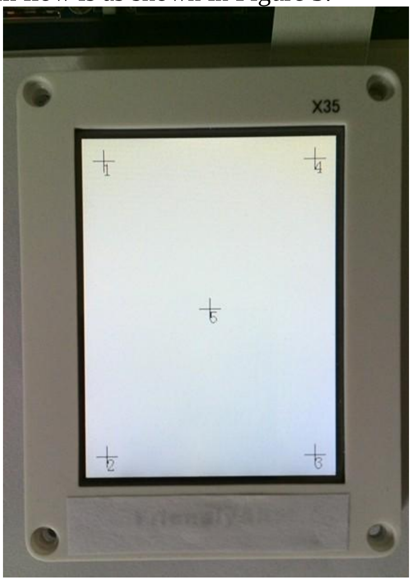


Figure 4. Five calibration points displayed LCD

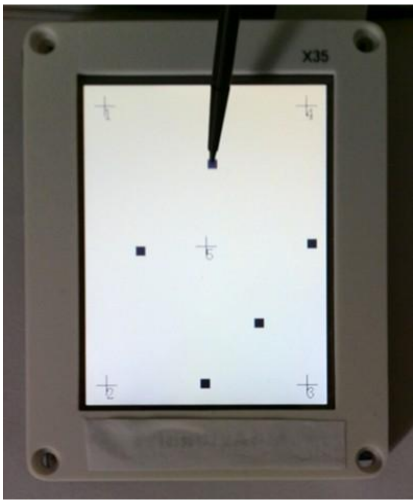


Figure 5. Deviation correction

IV. ALGORITHM TEST

A. Test Environment

The test is carried out with S3C2440A ARM9 development board, external 4-wire resistive touch screen and 240×320 pixels 3.5 inch TFT true color LCD screen (vertical screen). The development board is connected with the PC through the serial interface and the PC is used to carry out on-line debugging for the program via the JTAG interface in the development board and ADS1.2 integrated development environment is used.

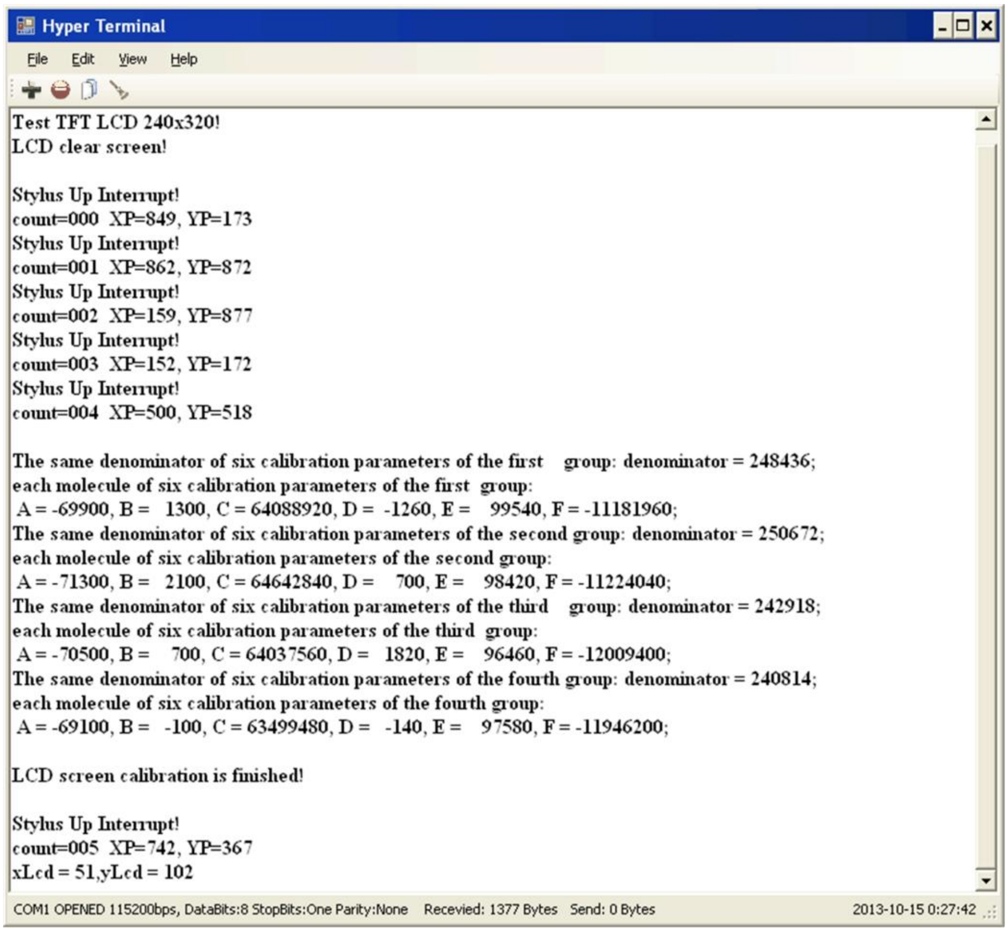


Figure 6. Calibration coefficients

B. Experimental Process

The 4 points among the selected 5 calibration points are located at the four corners of the screen, for the poor linearity of the touch screen edge, some distance shall be left between the 4 points and the screen edge. Space between those points shall be wide enough to reduce the scaling error as much as possible. According to the height and width (240×320) of the screen, 5 calibration points shall be selected, whose coordinates are (20, 20), (20, 300), (220, 300), (220, 20) and (120, 160) by the sequence of left top→left bottom→right bottom→right top→the center and display "+" and the serial number shall be displayed in the 5 points, as shown in Figure 4. Then, according to the order of left top→left bottom→right bottom→right top→the center, by using the touch pen, click 5 "+" symbols as shown in figure 4.

The CPU clock's initialization is that the desired clock frequency is set. S3C2440A external crystal frequency is 12 MHz. After frequency multiplication, FCLK frequency which is provided for CPU clock signal is 400 MHz, and HCLK frequency which is used for AHB (Advanced High-performance Bus) is 100 MHz, and PCLK frequency which is used for APB (Advanced Peripheral Bus) is 50 MHz.

Initialization of the touch screen includes setting A/D conversion frequency, setting touch screen interface mode, and setting interruption more importantly. Coordinates of the touch screen can be obtained through interrupt service subroutine when the user taps the touch screen. Designed interrupt service subroutine on the touch screen, firstly, "Waitting for Interrupt Mode" is set. After an interrupt is generated, Auto (Sequential) X/Y Position Conversion Mode is set, and the coordinate value of contact point is sequentially obtained. The serial interface is used to output calibration coefficient and other verification information. Initialization of the serial interface includes setting the serial port baudrate, data bits, parity bit, stop bits and so on. Initialization of LCD prepare to display "+" and serial number on the screen.

C. Experimental Result

5 calibration points that are shown in Figure 4 are tapped successively. According to the serial interface output information in Figure 6, it can be seen that 5 coordinate (XP, YP) of touch screen shall be obtained through interrupt service subroutine. For each group of calibration coefficients, the common denominator is firstly output and then the numerators of each group (A, B, C, D, E and F), the process is a little different from the previous mathematical analysis. Because integer arithmetic is applied in the calibration algorithm, if we directly divide the numerator by the denominator, once the numerator is less than the denominator, the quotients might be 0, which doesn't conform to the actual situation. Therefore, calibration coefficients (A, B, C, D, E and F)

in Figure 6 are the numerator of the actual coefficient. The last two lines in Figure 6 are the coordinates of one touch screen and the corresponding converted LCD coordinates. Limited by space, the screenshot only gives the coordinate (742, 367) of one touch screen and the corresponding LCD coordinate (51, 102). It can be found that LCD and touch screen coordinate values exist numerically larger gap.

The concrete realization of deviation correction function in the main program is based on the obtained calibration coefficient. Tap a point on the touch screen, and then calculate the LCD coordinates, and draw a small solid square with the coordinates as the center, and then determine whether the location of the square is the same as the position where the use taps the screen. If so, the calibration coefficient shall be stored in the memory, otherwise, the calibration coefficient shall be re-calculated, as shown in Figure 5. The calibration coefficient shall be stored in the EEPROM memory in the development board [13].

V. CONCLUSION

Considering that, parameters of resistive touch screen changes for the physical properties of itself, frequent calibrations are needed. This paper proposes the five-point calibration algorithm (also known as multi-point mean calibration algorithm) to solve the calibration of touch screen. This algorithm uses the integer arithmetic instead of floating-point operations and thus reduces the memory space and increases the computing performance and is suitable for characteristics and operation environment of embedded devices. The deviation correction function solves the problem of great deviation from the cross center point of the sample points of the touch screen for hand trembling or obscure vision, which shall leads to incorrect calibration coefficient. Such function can effectively reduce the possible maloperation during calibration process and improves the practicability. Experiments show that the algorithm is accurate and has actual value. Therefore, the algorithm has been used for calibration of touch screen of embedded devices.

ACKNOWLEDGMENT

Funding Project: Chengdu College (Chengdu University) School Funding Project (Project No.: 2012XJZ18)

REFERENCES

[1] Yunlong Xu, Zeng Zhao, Weiguo Wu, Yixin Zhao, RPPA: A Remote Parallel Program Performance Analysis Tool, *Journal of Software,* Vol. 6, No. 12, pp. 2399-2406, 2011

[2] Jun-jun Wu, Ming-wei Fang, Xin-fang Zhang, Tong-yang Wang, Trusted Anonymous Authentication Scheme for Trusted Network Connection in Mobile Environment, *Journal of Networks,* Vol. 7, pp. 1341-1348No. 9, 2012

[3] Peng Xiao, Zhigang Hu, Two-Dimension Relaxed Reservation Policy for Independent Tasks in Grid Computing, *Journal of Software,* Vol. 6, No. 8, pp. 1395-1402, 2011

[4] Lijun He, Shuang Liu, Jian Yun, Yongkui Liu, A Line Generation Algorithm over 3D Body-centered Cubic Lattice, *Journal of Multimedia,* Vol. 8, No. 1, pp. 40-47, 2013

[5] Tariq Faisal. An Easy-to-Understand Explanation of Calibration in Touch Screen Systems. *Electronic Engineering Times,* pp. 36-37, 2012

[6] Xu rongbin, Xie ying, Zhu yonghong. Usual Algorithm Analyzing of Touch Screen's Calibration. *Industial Control Computer,* No. 4, pp. 77-78, 2006

[7] Song Xuerui, Cai Ziyu, Duan Qingqing. Data Processing Algorithm of Touch Screen. *Computer Engineering,* No. 12, pp. 255-257, 2008

[8] Shi Jian, Bai Ruilin, Zou Junyu, Ma Tao. Design and Realization of Android Multi-point Touch Screen Input System. *Computer Engineering and Applications,* No. 28, pp. 66-67, 2012

[9] Jing Wang, Yourui Huang, Decimal Integer Multiplication based on Molecular Beacons, *Journal of Networks,* Vol. 8, No. 12, PP. 2762-2768, 2013

[10] Xiaoxiao Liu, Jun Steed Huang, Zujue Chen, The Research of Ranging with Timing over Packet Network for the Mine Safety Application, *Journal of Networks,* Vol. 7, No. 7, PP. 1054-1062, 2012

[11] Yanjun Zhao, Wanqing Zhang, Wenqing Ge, Shuguang Li, Finite Element Simulation of Soldier Lower Extremity Exoskeleton, *Journal of Multimedia*, Vol. 8, No. 6, pp. 705-711, 2013

[12] Qian Zhao, Research on family and shops real-time status of 3G Wireless Remote Monitoring System, *Journal of Software,* Vol. 6, No. 5, pp. 814-818, 2011

[13] Meng xianglian, Sun ping, Zhu guohan, Gao hongzhi. Embedded System Principle and Application Tutorial. Beijing: *Tsinghua University Press, 2010*

Li Ping-yong (1974-), Male, Born in Chengdu, Sichuan, Lecturer, Master of Science, Main research fields: embedded system and automatic control system.