**Final report of comprehensive experimental project of computer hardware design foundation**

**Lin Junbo Machinery 92 2019010636**

**Project Name:**Music Player and Simple Electronic Organ

1. **Function description and function operation description:**

In this experimental project, a total of K1-K8 buttons and S2 buttons were used, a total of 9 buttons, two passive buzzers, L1, L2, L3, L4, L5, L6, L7 and L8, a total of 8 light-emitting diodes, Bluetooth modules and an Android mobile phone with Bluetooth.

At startup, buzzer 1 will emit Nokia's classic boot sound and have LED light effect, and then the serial assistant on the Android phone connected to the Bluetooth module will appear on the boot interface for the user to select the function mode, enter 1 will enter the play mode, enter 2 will enter the play mode.

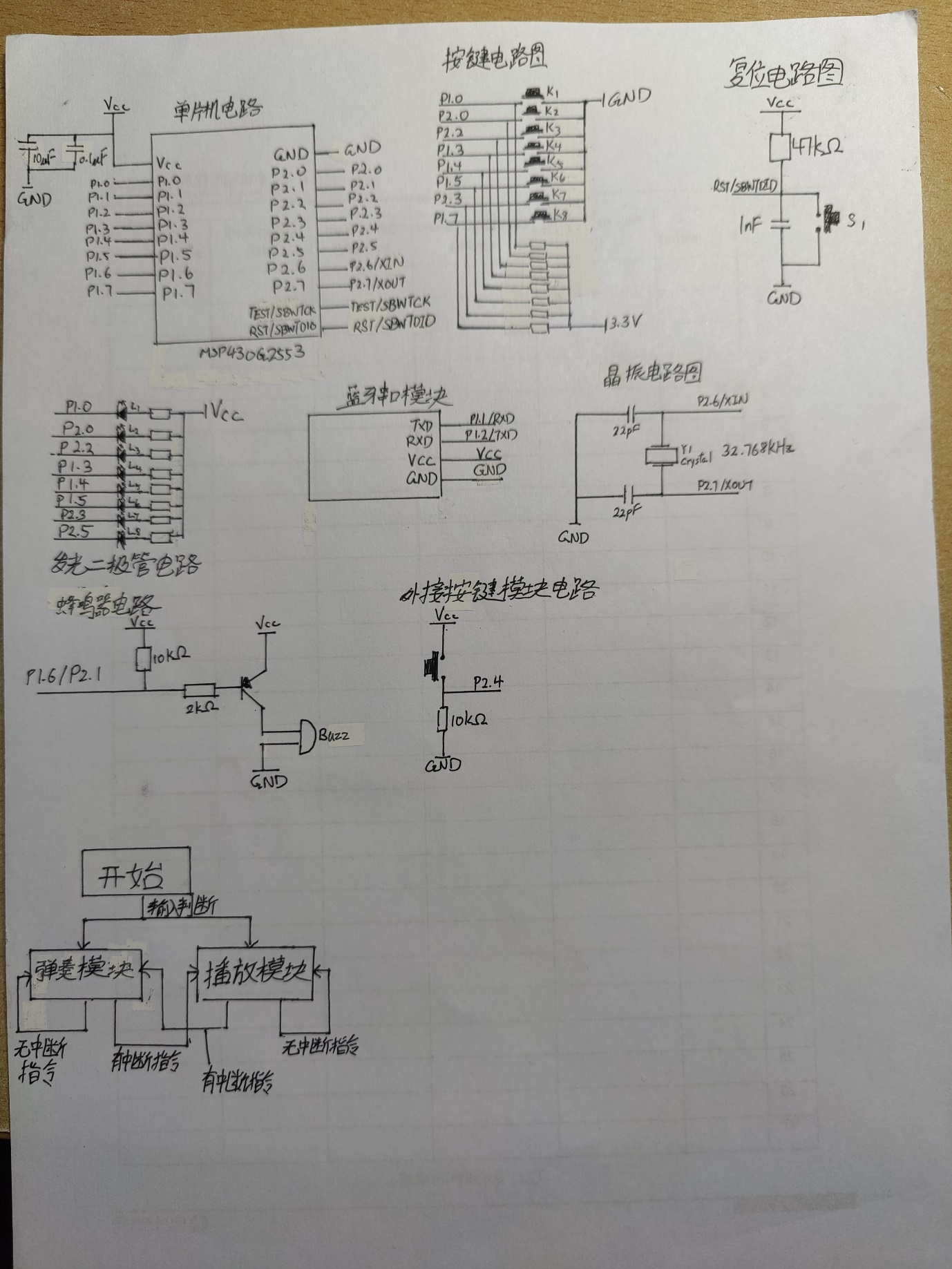
After entering 1, the serial assistant on the mobile phone will appear in the performance window indicating that it has entered the performance mode, in the performance mode press K1 to K7 to play do, re, mi, fa, so, la, ti 7 tones, when pressing K8, and then press any key in K1~K7, the buzzer emits a low octave tone, such as pressing the S2 key and then pressing any key in K1~K7, it will emit a high octave tone. If you enter 's' in the Bluetooth serial window, it will stop playing mode and enter play mode.

After entering 2 in the start window or 's' in the playing window to enter the playback function, the serial assistant will appear in the playback window, and then prompt the available tracks and the corresponding tracks. Input 1 will play the famous episode of Disney's animation Aladdin, "A Whole New World", input 2 will play Lo Dayou's "Shining Days", and input 3 will play the theme song of JOJO Golden Wind, "il vento d'oro". After selecting the track to play, the serial port window will prompt which song is playing, and the LEDs L4 and L5 will light up for a short time before playing "A Whole New World", and then turn off to indicate that the first track is playing, and then start playing the track; Before playing "Shining Days", the LEDs L3, L4, L5 and L6 will light up for a short time and then turn off, indicating that the second track is being played, and then the track will start playing; Before playing "Il Vento d'Oro", the LEDs beat quickly for a short time to match the style of the song, then turn off, indicating that the third track is being played, and then start playing the track. After the song finishes playing, it will return to the playback track window, allowing the user to re-select the next song or enter 'p' to enter play mode. During the playback of the song, you can terminate the current playback track by entering 's' and enter the initial playback interface, select the song again or switch to strumming mode.

**(1) Design ideas:**

The playback of the track is to output PWM signal waves of different frequencies by adjusting the timer TA, and output the tone of the corresponding frequency of the corresponding scale. The frequency of tones used in the program is as follows

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| scale | A | A# | B | B# | C | D | D # | And | And # | F | F# | G |
| Lower octave |  |  | 294 |  |  |  | 367 | 391 | 415 | 440 | 466 | 494 |
| C tune | 523 | 554 | 587 | 622 | 659 | 698 | 740 | 784 | 831 | 880 | 932 | 988 |
| High octave | 1046 | 1108 | 1174 | 1244 | 1318 | 1397 | 1479 | 1568 | 1661 | 1760 |  | 1976 |

**(2) Hardware system design:**

The transmit and receive pins RX/TX and P1.1/P1.2 of the serial interface are the same pin, P1.1/P1.2

The XIN/XOUT connected to the external crystal is P2.6/P2.7 of the breadboard

The PWM comparison output pins of the timer are P2.1 and P1.6

The S2 of the breadboard is connected to P2.4

Therefore, the pinout is as follows:

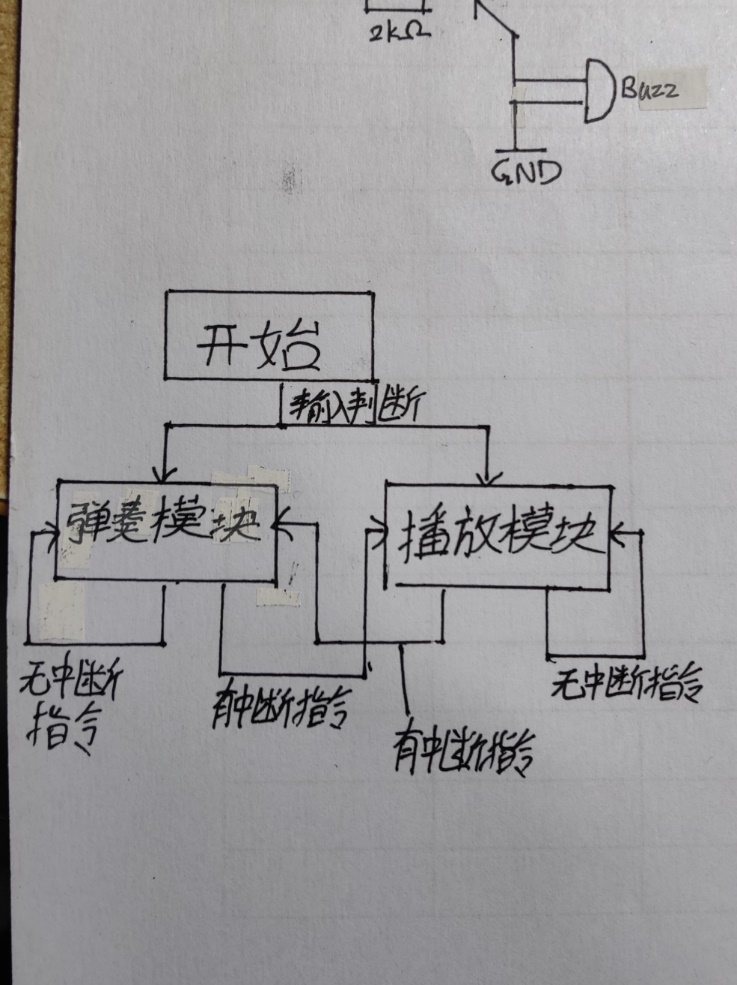
* P2.1 and P1.6 are connected to external passive buzzers as PWM comparison output pins of timers TA1 and TA0, respectively.
* P1.1 and P1.2 are connected to the TXD and RXD of the Bluetooth module as UCA0RXD and UCA0TXD, respectively.
* P2.6 and P2.7 are connected to the crystal oscillator of the experimental expansion board.
* P2.4 is connected to an external S2 button.
* P1.0, P2.0, P2.2, P1.3, P1.4, P1.5, P2.3 and P2.5 are connected to the LEDs L1 to L8 on the board.
* P1.0, P2.0, P2.2, P1.3, P1.4, P1.5, P2.3, P1.7 Connect the keys of K1-K8

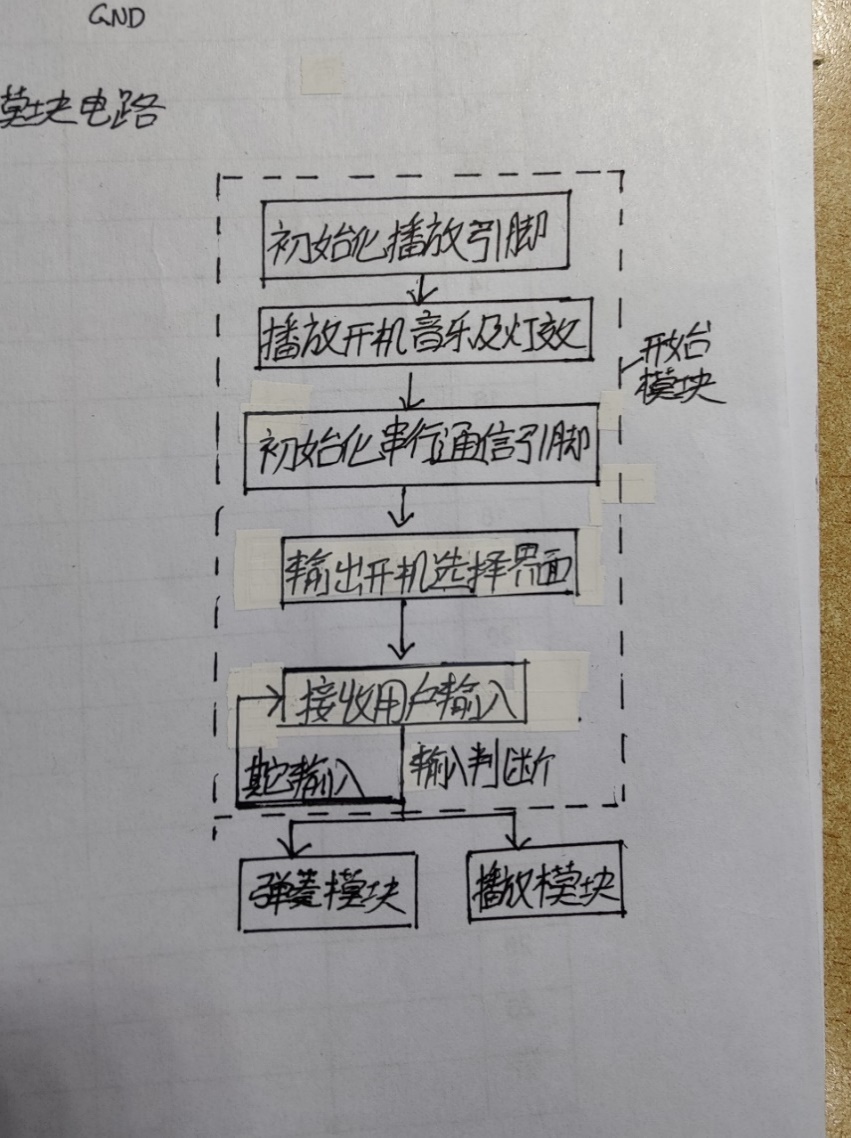
**(3) Software design**

The software is mainly divided into three modules: start module, play mode module and play mode module.

The starting module is mainly for boot prompt, initialize serial pin, output playback boot interface, and after the user enters the command to select the mode, initialize the corresponding pin and enter the mode (playback mode or performance mode)

The performance module is mainly responsible for outputting the playback interface and playing the corresponding tone when the user presses the corresponding button.

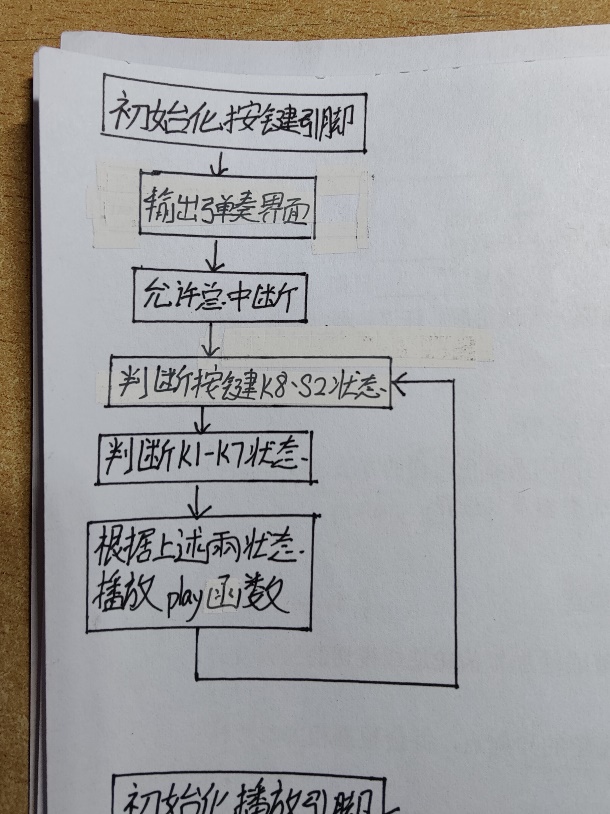
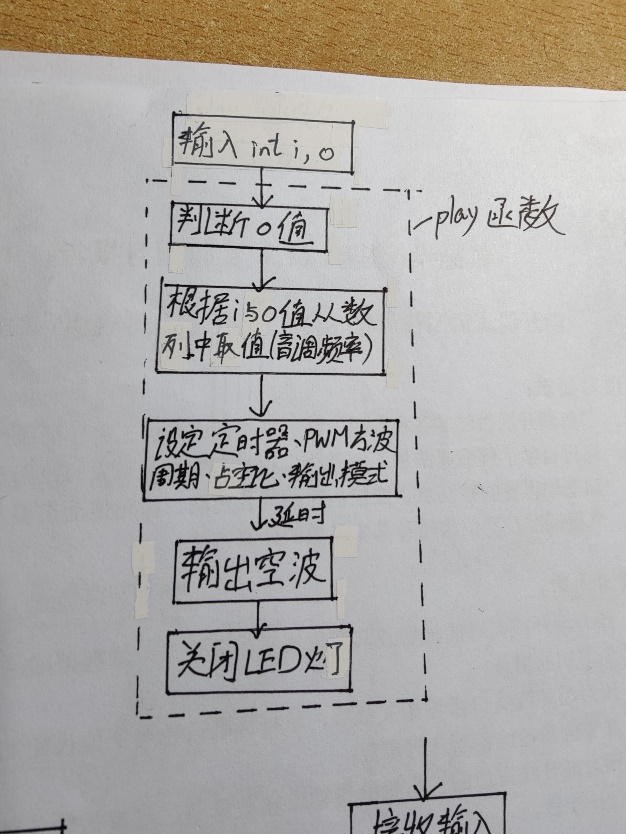
The playback module is responsible for outputting the playback interface, waiting for the user to enter the corresponding command to select the track and play the corresponding track

1. Power on the module

After completing the pin initialization, the boot music will be played, and then the Bluetooth serial output boot user interface, and the user input will be judged, if it is 1, it will enter the play mode, if it is 2, it will enter the playback mode.

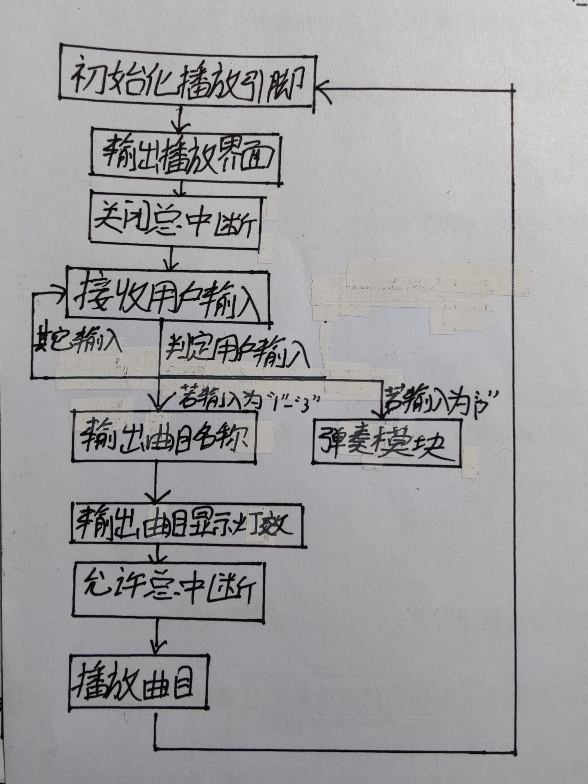
Initialize serial communication:

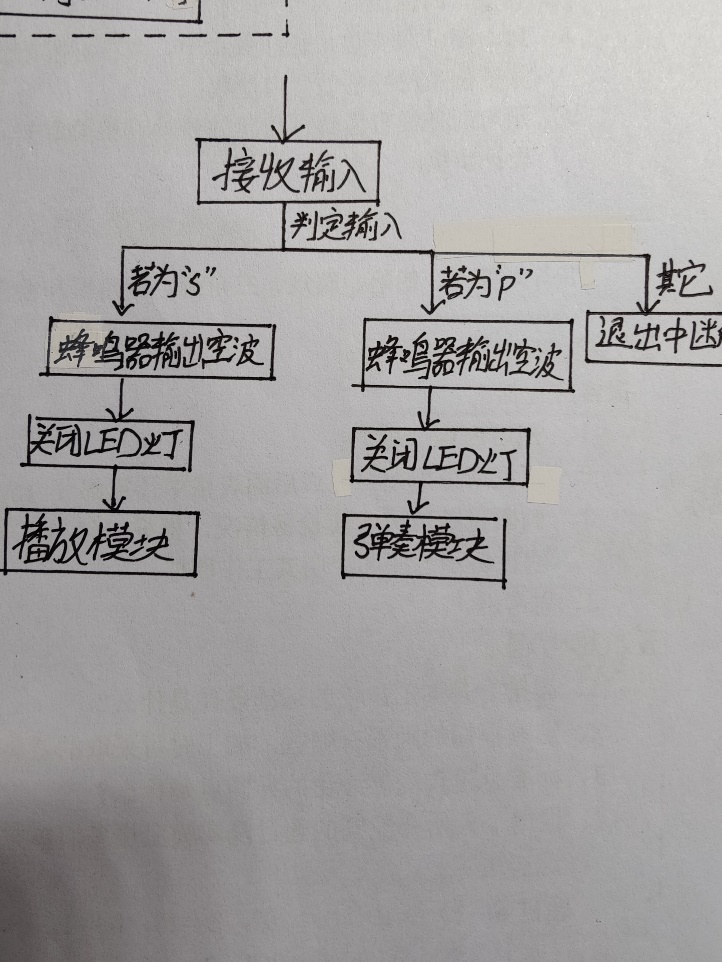
First, set the software reset swrst to 1. Then set P1.1 and P1.2 to the serial interface receive and transmit pin functions, and set the baud rate clock to ACLK at power-on reset, 32.768KHz, right and wrong. Then set the baud rate to 9600. The final software reset SWRST is 0, and the serial port is set.

1. Strumming module

After entering the playing mode, the interface of the playing mode will appear, after entering the piano() function, whether K8 or S2 is pressed will be judged, if K8 or S2 is pressed, the value of the variable o will be equal to 2 and 3 respectively, otherwise the value of o will be equal to 1, and then whether the key is pressed will be judged, if it is pressed, it will enter the corresponding play (note, o) function (the value of note in the function is the corresponding number of the key, i.e. The function after pressing the K1 key is play(1,o)). After entering the play function, the frequency of the corresponding note will be obtained from the string playnote[] according to the value of i, and then the value of o will be judged, if o=1, the frequency of the note will remain unchanged and the corresponding frequency will be output; If o = 2, the frequency of the note will be divided by 2 to reduce its pitch by an octave; If o = 3, the frequency of the note will be multiplied by 2 to raise its pitch by an octave; Then the timer is initialized, and the counting clock, PWM period, PWM output mode, TA1 comparator setting value CCR1 and increase counting method of TA1 timer are set. The PWM period and the setting value CCR1 are set to (32786/note frequency)-1 and 1/10 of the period, respectively, to output 90% of the square wave. After the note is output, a delay is performed, followed by the PWM period and the setpoint set to 0, the buzzer is turned off, and the play(note,o) function ends and the piano() function loop is returned. When the serial communication application is interrupted, the function operation will be stopped to enter the interrupt function, in the interrupt function, the instructions entered by the user will be read, if it is not 's', it will return to the piano() function loop; if it is 's', it will enter the playback function module.

1. Play module

After entering the playback module, the interface of the playback mode will appear, after entering the song() function, the instructions entered by the user will be judged, if the input instruction is a number of '1'-'3', it will enter the track function corresponding to the corresponding number; if the input instruction is 'p', it will enter the playing module; if the input instruction is not one of the above two, an error will be reported and the user will be asked to re-enter the instruction. When entering the numbers '1'-'3', the program will output the name of the track that is now playing to the user, and the LEDs will show the corresponding lighting effect for each song, and L4 and L5 will light up when playing "A Whole New World"; L3, L4, L5 and L6 light up when playing "Shining Days", and L1 to L8 flash quickly when playing "il vento d'oro"; Then start playing the track. After the track finishes playing, it will return to the song playback mode interface, allowing the user to select the song again. During music playback, users can terminate playback by entering 's' and return to the playback interface.

1. Interrupt function

In playback mode and strumming mode total interrupt application will be allowed, when the Bluetooth serial has input will apply for interrupt, at this time the input command will be judged, if the command is 's' interrupt function will turn off the buzzer output and LED output, and then initialize the playback pin and output the interface and playback function of the playback mode; if the command is ' The p' interrupt function will turn off the buzzer output and LED output, and then initialize the key pins and output the interface and play function of the play mode

**2.**  **Project realization:**

**(1) Three difficult problems and solutions in the debugging process**

During debugging, the biggest problem encountered is that the memory RAM is small, so that it is necessary to use const and define strings reasonably when it becomes necessary. In the original design, more functions were used to define strings (y[60], etc.), which made it easier to use more RAM and easily run out of RAM. For this reason, I optimized the program by using fewer strings (only using the string y[2] up to 2 in the end), so that the memory usage is less, and the memory freeing allows me to do more functions.

The second problem is the problem of insufficient pins on the single-chip microcomputer, because of the need to realize the use of buttons, light-emitting diodes, buzzers, serial communication, Changzhen, etc., but only P1.0 to P1.7, P2.0 to P2.7 16 pins can be used, so when initializing the pins, I initialize two types according to the two modes of playing and playing, and before entering each mode, you can initialize the same pin for two types of initialization and functions.

The third problem is the way to deal with several modules. Since the music needs to cooperate with two buzzers and light-emitting diodes when playing, in addition to considering the tone emitted by the buzzer, it is also necessary to arrange the beat of each tone, lighting effects, etc., so that each function needs to be carefully programmed.

**(2) Insufficient project completion and areas for improvement**

I think I have implemented most of my original functions for this project, but there are two functions that I have not implemented due to technical problems, they are breathing light debugging and buzzer to achieve simultaneous playback of different beats.

First of all, in the debugging of the breathing lamp, I originally planned to make the light-emitting diode to achieve a different human-like breathing lamp gradient according to the music passage, but I found it very difficult to achieve it when it needed to match the number of tone outputs, so the final solution was abandoned. I think it's a shame because I think this scheme fits well with my choice of track, "A Whole New World", and it's a big pity for me to experiment.

In the playback of the buzzer, I originally planned to use two buzzers to play the bass and treble parts of the piano respectively to achieve better harmony. However, in actual operation, it has been found that the beats and beats of bass and treble are often different, resulting in difficulty in achieving simultaneous playback of different white tones when they become them. I have also tried other methods such as beat delay counting in sixteenth notes, but they were not ideal and were invalidated. So that's one of my biggest regrets in this project.

In general, the above two are not enough, I think that the completion of my project is high, and I am quite satisfied with my completed work, very grateful to the teacher's guidance and answering my questions on WeChat in my spare time to help me complete my work.