《Embedded System and Microcomputer Principle》Project Report

Topic	Embedded System and Microcomputer Principle Project Report					
Group No.	2					
Member 1	SID	12111611	NAME	Ruixiang Jiang	Coefficient	0.25
Member 2	SID	12111944	NAME	Yujing Zhang	Coefficient	0.25
Member 3	SID	12112841	NAME	Ziyu Han	Coefficient	0.25
Member 4	SID	12112910	NAME	Jiacheng Luo	Coefficient	0.25

I. System Function (1%)

Opening the MiniSTM32 board, a top design is shown on the screen. There are several modules, the user, the current time, and some functionalities act as buttons, which will be introduced below.

The first functionality, which looks like the WeChat app, is used to communicate with other users. This chat powered by the 2.4G communication module, enables seamless data exchange between two MiniSTM32 boards, facilitating a real-time chat experience. The system intelligently distinguishes between online and offline user states, defining online status when the board is powered and in the smartwatch interface, even if the user has not entered the chat component. Conversely, offline status is assigned when the board exits the main interface of the smartwatch program. Users can input text messages through serial communication tools, providing a user-friendly experience. The 2.4G communication module ensures the efficient transmission of chat content between boards in a peer-to-peer manner. A status detection mechanism notifies users if the recipient is offline, ensuring the reliability of message transmission. The chat interface offers a display on the MiniSTM32 board's LCD screen. It includes basic display functions showcasing user conversations with sender’s username. The interface intelligently handles line breaks for lengthy messages, ensuring readability. Additionally, a pagination feature enables users to navigate through extensive chat histories comfortably. Expanding beyond one-on-one communication, the system enhances the chat component to support multiple users concurrently. It incorporates features such as storing information about users with whom the local user has interacted, facilitating seamless transitions between online and offline states. Notifications for user status changes, including online and offline alerts, contribute to a comprehensive and user-centric chat experience. To further enrich the communication experience, users can leverage a predefined list of emojis, injecting a sense of expressiveness and engagement into their conversations. The entire chat interface is designed with a full-color display.

The second functionality is a calculator, which has three modes, decimal computation, equation solving and binary computation. The decimal computation mode, akin to commonplace calculators used in daily life, is employed for performing arithmetic operations between decimal numbers. It facilitates addition, subtraction, multiplication, and division, as well as exponentiation. Additionally, it accommodates alterations in the order of operations with parentheses. Naturally, this mode supports the amalgamation of integer and decimal operations. The “left” and “right” buttons are responsible for sliding the equation window. Note that if the equation is very long, you can use it moving left or right to see the whole input equation. The “change” button is used to switch the function of the calculator, i.e., decimal computation, equation solving and binary computation mode. In the decimal or binary computation part, once you click the “=” button, the result will be calculated, while in the equation solving part you should double click “=” to make the equation been solved.

The third functionality is an album feature. Upon inserting an SD card and clicking this button, the board will read all the images within the "picture" directory on the SD card and display them on the screen. In addition to full-screen image display, the name of the image and the current playback progress will be shown in the top-left corner of the screen. To enhance user convenience, it supports automatic slideshow of images. Of course, users can also

manually navigate through images using the key0 and key1 buttons to switch to the previous or next image.

The fourth functionality is a Tetris game. At the outset, a randomly shaped block descends slowly from the top of the play area. While it is descending, players can use the "turnL" or "turnR" buttons on the screen to rotate the block by 90 degrees, the "left" or "right" buttons to move the block horizontally by one grid unit, or the "c" button at the top of the screen to instantly drop the block to the bottom. When the block reaches the bottom of the play area or lands on top of other blocks, preventing further downward movement, it becomes fixed in place. Subsequently, a new random block appears at the top of the play area and begins to descend. The game concludes when fixed blocks accumulate to the top of the play area, making it impossible to clear lines. When all cells in a horizontal (or vertical) row are filled with blocks, that row is eliminated, contributing to the player's score. Additionally, the top portion features a "p" button for pausing or resuming the game and a "back" button to return to the main interface. On the right side of the screen, a scoreboard displays the current score, and there's an area reserved for previewing the next block to appear.

II. System Design (2%)

1.working principle

The working principle of this program can be summarized as follows:

(a) Basic Framework: Implement an initial interface for users to choose which component to use. Display current time and user information. Provide a method for users, that is, press KEY_WK, to return to the initial interface without affecting component usability.

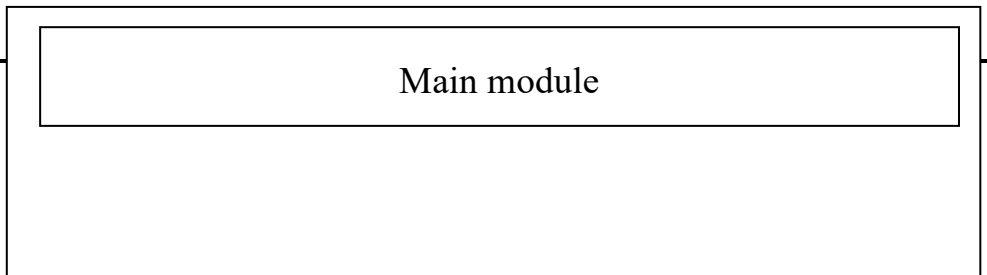
(b) Chat Module: We utilize the development board's 2.4G communication module for data exchange between two development boards, differentiate between users' online and offline statuses based on entering/exiting the smartwatch interface, and implement chat input through serial communication tools. Users can transfer chat messages between development boards in a peer-to-peer manner and can also detect the status of the friend.

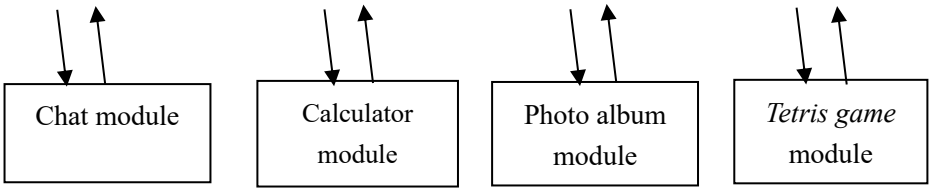
LCD is used to show conversation information, on which we ensure proper line breaks for long messages and provide page navigation for viewing extensive chat content, and we can store user information for users who had a chat history. Allow users to send emojis from a predetermined list during chats. Users can initiate chat invitations and receive chat invitations even while using other components.

(c) Calculator Module: Users input information through the resistive touch screen. The calculator supports basic arithmetic operations with decimal results displayed. Users can input numbers and use buttons for operators and clear. Error detection is implemented, and the user's input formula is displayed. The calculator calculates and displays the result when the "=" button is pressed. The calculator also supports parentheses for changing the order of operations. It can solve quadratic or linear equations and display the result with the unknown variable. A separate interface is provided for binary operations. The calculator allows users to continue entering when the input formula exceeds the page limit. Users can scroll the displayed formula left or right to view the entire input.

(d) Photo album module: This module allows users to read and display JPEG format photos stored on an SD card. It supports full-color display and provides options to switch between photos through swiping the screen or using serial communication. The photos are located in the "/PICTURE" directory on the SD card. The interface also includes the ability to show the total number of photos in the album. Additionally, users can utilize an infrared remote control to switch and display photos. The implementation may vary, but these functionalities offer a convenient and enjoyable photo browsing experience on the smartwatch.

2. system frame diagram





3.Sub-module design

The system is divided into 4 sub-modules, they are the chat (communication) module, the calculator module, the photo album module and the Teris game module.

(a) Chat module:

We mainly implement the send and the chat_touch_handler function, which are responsible for sending and receiving messages via the NRF24L01 wireless module respectively. The send function takes a message as input and sends it through the module. We display of messages on an LCD screen. It includes the ability to display different types of messages, including text messages and emoji messages. And we divide longer messages into multiple lines to fit on the screen and manages the total number of lines displayed. It also saves the messages in an array for future reference.

Additionally, in both send and receive function, we check for successful transmission of messages and handles the scenario when there is a failure in sending a message. It updates the status of the recipient and displays appropriate notifications on the LCD screen.

(b) Calculator module

We use several functions to implement this module, the core function work_for_calc provides the core functionality for handling user inputs and performing calculations in a calculator program. And other functions, including draw_calc_screen, calc_touch_screen_handler and display_calc_equation, contribute to presenting the calculator screen, handling touch events, and updating the displayed equation based on user input.

Based on the coordinates of the button pressed, the work_for_calc() function for calculator performs various actions such as appending characters to the equation, clearing the equation and result, calculating expressions and equations, and displaying the results. It also manages the screen state of the calculator program. We also check for valid input coordinates and returns if they are out of range. If the input is valid, it updates the equation array accordingly and performs the corresponding calculation or action based on the current screen state. In addition, the function handles the case when the equation length exceeds the display limit by adjusting the display range and clearing the equation if necessary.

display_calc_equation() function displays a portion of the equation on the LCD screen. It extracts the relevant part of the equation based on the current display range (calc_equation_left and calc_equation_right) and shows it on the screen using the LCD_ShowString() function.

draw_calc_screen() function updates the UI of the calculator screen. It clears the screen, displays background images and mode notices based on the current screen state. It also sets the coordinates for the calculator buttons.

calc_touch_screen_handler() function handles touch screen events. It checks for touch events and performs corresponding actions based on the touch position and button states. If a touch event occurs within a specific button's area, it calls the work_for_calc() function to perform the calculation associated with that button. It also handles button shifting events when the shift buttons are pressed.

(c) Photo album module

This album fuction implements a photo album feature. It utilizes an NRF24L01 module to receive photos wirelessly. We first check if the folder containing the pictures is accessible. If not, an error message is displayed. It

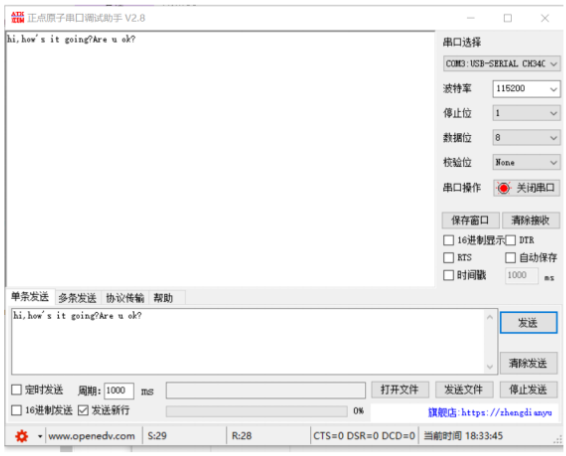
then retrieves the total number of pictures in the folder. If no pictures are found, a message stating “No picture!” is shown. Memory is allocated for storing file information, file paths, and picture offsets. It proceeds to open the folder and iterate through the files, recording the offsets of image files. After displaying a message indicating the start of the slideshow, it initializes the display library. A loop is entered to display the images in the album. The current image’s index and path are used to load and show the image on the LCD screen. The current image’s index and total number of images are displayed on the screen. It continuously checks for user input to allow for navigation through the album. Pressing buttons can move to the previous or next image, pause/resume the slideshow, or return to the initial screen. LED indicators are used to show the state of the slideshow and user actions. The process continues until the end of the album is reached. Memory allocated for file information, paths, and offsets is freed.

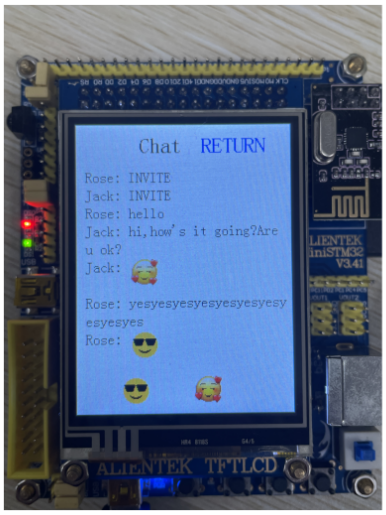
(d) Tetris game module

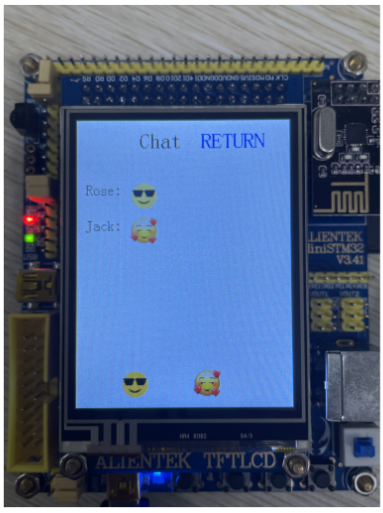
In this module we implement a Tetris game, allowing the player to control the blocks using touch inputs and displaying the game map and score on an LCD screen. The startGame() function initializes the shape, form, and id of the first block to be randomly generated. It also sets a flag hasPressed to track whether a button has been pressed. It then switches to the NRF24L01 module’s receive mode. Inside a while loop, the code continuously checks for inputs and generates the next shape and form randomly. It sets the initial position of the block and enters another while loop. In this loop, it scans for touch inputs, draws the current block, prints the game map, and shows the next block. If the game is paused, the code waits for a touch input to resume the game. If there is no input from the touch board, the block either falls to the bottom or moves left, right, rotates counterclockwise, or falls quickly to the bottom based on the input received. If the block reaches the bottom, the lines are cleared and the loop is broken. If the input is ‘R’, the game returns to the initial screen. After the game loop ends, the shape and form of the next block are updated, the game score is incremented, and the score is displayed on the screen. The tetris() function initializes the game and draws the Tetris screen. It then calls the startGame() function to start the game.

III. Results (screenshots and hardware photos) (1%)

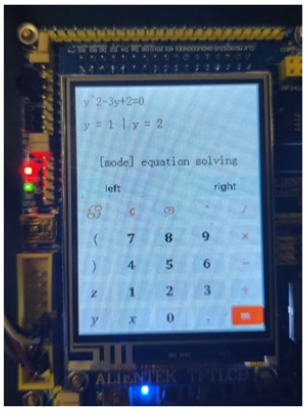
(1) 2.4G communication

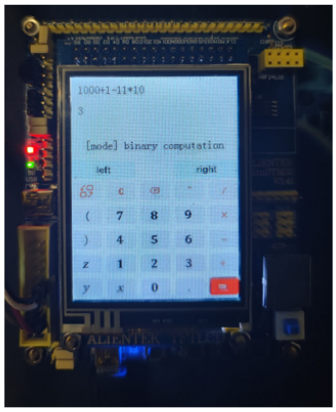




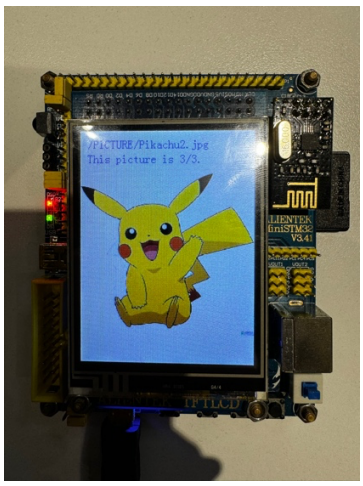


(2) Calculator

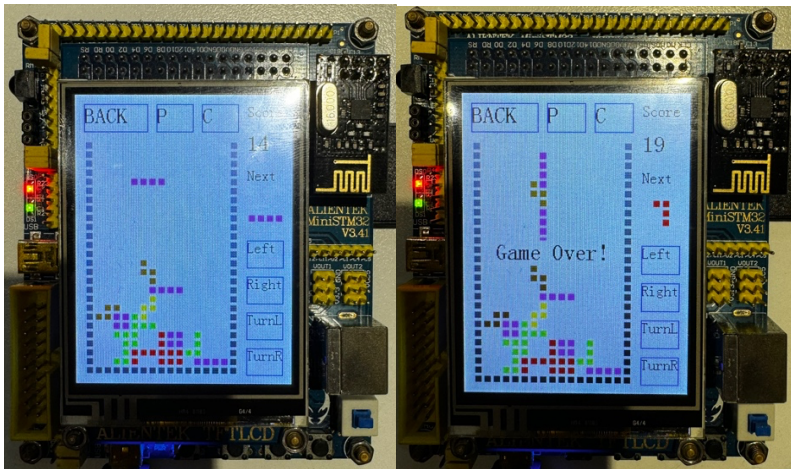




(3) Album



(4) Tetris game



IV. Work allocation description (1%)

(1) Member 1: R.X. Jiang, who is responsible for the logical part of calculator, the album, and the Tetris game.

(2) Member 2: Y.J. Zhang, who is responsible for 2.4G communication part.

(3) Member 3: Z.Y. Han, who is responsible for 2.4G communication part.

(4) Member 4: J.C. Luo, who is responsible for the initial screen and the apparency of calculator.

V. Problems encountered and solutions (1%)

Problem1: The problem of blocking when two development boards confirm each other's status

Solution: We have changed the method of confirming the status. At the beginning, we hope that both parties can always send a specific string "check" to confirm the status of the other party, and the successful sending means that the other party is online. However, this method leads to the continuous sending of information by both parties at the same time, resulting in the blockage of the communication channel. To solve this problem, we changed the method so that one party sends a specific string of "check" to determine the status of the other party according to whether it is successfully sent, and the other party determines the status of the other party according to whether it receives "check".

Problem2: While creating the initial interface, the imperative was to design a power-resilient clock for uninterrupted

timekeeping. However, complications arose, notably with the Real-Time Clock (RTC) failing to retain counter data. Rigorous troubleshooting revealed an unexpected issue: a systematic date reset every 24 hours, disrupting temporal integrity. Additionally, the timer displayed a significant timing error, adding complexity to the developmental phase. Solution: After thorough debugging of the code revealed no issues, it was identified that the button cell battery on the stm32 development board was faulty. This aspect of hardware programming can be frustrating since during code debugging, it is challenging to discern whether the issue lies in the code itself or the hardware. After replacing the button cell battery on the stm32 development board, it was noted that the RTC operates by automatically saving a 16-bit counter (representing hours, minutes, and seconds), but it does not automatically save the year, month, and day information. Subsequently, the year, month, day, hour, minute, and second information was converted into a 32-bit counter and stored separately in two 16-bit memory locations preserved after a power loss. To address the problem, a conversion from seconds to year, month, day, hour, minute, and second information was implemented, involving considerations such as leap year calculations. It was later discovered that the calendar used the internal LSI crystal oscillator as the RTC crystal source. However, due to fluctuations in the LSI crystal frequency, which is around 40 KHz and can vary between 30 and 60 KHz, the timekeeping was imprecise, resulting in an approximately 1-minute deviation per hour. Ultimately, the issue of timing deviation was resolved by utilizing an external 32.768 KHz LSE crystal oscillator as the RTC crystal source, providing a stable and accurate timekeeping solution.

VI. Personal proposal （4%）

Course Project: MiniSTM32 Children's Learning Device

1. Introduction:

This course project aims to develop a compact learning device for children based on the MiniSTM32 board, integrating various features to enhance learning and interactivity. The proposed functionalities include real-time weather display, wireless quiz competitions, MP3-based language learning, and a diary module for personal growth tracking.

2. Components Used:

a. MiniSTM32 Development Board: Serving as the core embedded system, this board offers versatile peripherals and communication capabilities. MiniSTM32 Board Information

b. DS18B20 Temperature Sensor Module: Connected to the MiniSTM32 board, this sensor provides real-time temperature data, which is displayed on an LCD screen. DS18B20 Module Information

c. 2.4G Communication Module: Facilitating wireless communication, this module enables synchronized quiz competitions with other children. It includes features for timing the quiz and displaying accuracy rankings on the LCD screen.

d. VS1053 MP3 Music Module: Integrated with the MiniSTM32 board, this module stores and plays English audio files to support language learning. VS1053 Module Information

e. Diary Module (UART Communication + SD Card): Using UART communication, this module allows children to record daily experiences, which are stored on an SD card. The diary entries persist even after restarting the development board.

3. Design Requirements:

a. Weather Display: The DS18B20 temperature sensor will continuously monitor the ambient temperature. The MiniSTM32 board will process the data and display real-time temperature information on the LCD screen.

b. Wireless Quiz Competitions: The 2.4G communication module will enable quiz competitions with other devices. The MiniSTM32 board will manage the quiz timing and display accuracy rankings on the LCD screen after the quiz concludes.

c. Language Learning with MP3: The VS1053 MP3 module will store English audio files for language learning. The MiniSTM32 board will control audio playback and provide a user-friendly interface.

d. Diary Recording and Persistence: The UART-enabled diary module will allow children to record daily experiences. The MiniSTM32 board will manage the data storage on an SD card, ensuring that diary entries persist across power cycles.

4. Feasibility:

a. MiniSTM32 Compatibility: The chosen components align with the capabilities of the MiniSTM32 board, making integration feasible. Necessary GPIO pins, communication protocols, and libraries are available for seamless implementation.

b. New Component Integration: The integration of the DS18B20, 2.4G module, and VS1053 MP3 module introduces new functionalities not previously explored. The feasibility is supported by the availability of documentation and community resources.

c. Diary Module and Persistence: Utilizing UART communication and SD card storage for diary entries is a practical and achievable goal. The MiniSTM32 board's support for these technologies ensures feasibility.

5. References:

DS18B20 Module Information - http://47.111.11.73/docs/modules/other/DS18B20.html

VS1053 Module Information - http://47.111.11.73/docs/modules/other/VS1053.html