International Journal of Engineering Inventions
e-ISSN: 2278-7461, p-ISSN: 2319-6491
Volume 13, Issue 4 [April. 2024] PP: 99-113

Design and implementation of smart greenhouse orchard system based on ARM series controller

Peizhe SUN, Bo YANG，Jianghao ZHANG，Zhu CHEN

ABSTRACT: *With the development of science and technology and society, intelligent applications have entered our lives, intelligent greenhouse orchard management has become a development trend, people's requirements for intelligence are getting higher and higher, how to scientific and intelligent management has become a development trend, people's requirements for intelligence are getting higher and higher, how to scientific and intelligent greenhouse orchard management has been paid attention to, improving efficiency and liberating labour productivity is the purpose of this research topic. greenhouse orchard management has been paid attention to, improving efficiency and liberating labour productivity is the purpose of this research topic. This intelligent greenhouse orchard system uses sensor acquisition technology and camera v4l2 technology to collect real-time environmental information and image information, and uses Cisco® CCDMA® to provide the most efficient and effective way to manage the greenhouse orchard. information and image information, and uses CortexA9 controller as a server to use multi-threading to receive and process real-time data information, and the client is developed by Qt and the client is developed by Qt.*

The intelligent greenhouse orchard system detects the environment in real time through CortexM0 and the camera, transmits the data to the server in time, responds after the server processes it, adjusts the environment and notifies the management personnel; The video, light intensity, temperature and temperature of the orchard is not a problem. responds after the server processes it, adjusts the environment and notifies the management personnel; The video, light intensity, temperature and humidity and other environmental information collected by the camera and CortexM0 and CortexM0 are not available. humidity and other environmental information collected by the camera and CortexM0 are displayed on the local LCD display, which allows users to monitor the greenhouse orchard, and can be used to monitor the environment. the greenhouse orchard, and can also be transmitted through the network socket, and can also be remotely managed on the remote client; Through local cameras, sensors, display screens and other equipment, servers, remote clients, the trinity of environmental detection and management control, scientific and intelligent greenhouse orchard management scientific and intelligent greenhouse orchard management has been realised.

Keywords: *intelligence, intelligence, data, monitoring, management*

Date of Submission: 02-04-2024 Date of acceptance: 12-04-2024

I. INTRODUCTION

1.1 Background and significance of the research project

At present, greenhouse orchards are slowly developing towards automation, and the increase of labour costs and some drawbacks of labour make mechanization develop rapidly; Networked and intelligent is the trend of development, and the intelligence of greenhouse orchard management industry is not a new technology. At present, greenhouse orchards are slowly developing towards automation, and the increase of labour costs and some drawbacks of labour make mechanization develop rapidly; Networked and intelligent is the trend of development, and the intelligence of greenhouse orchard management industry However, the degree of intelligence is not high, and there are no products that are more in line with the actual situation. However, the degree of intelligence is not high, and there are no products that are more in line with the actual situation. With the improvement of China's economic strength, the people's requirements for material living standards continue to increase, and With the improvement of China's economic strength, the people's requirements for material living standards continue to increase, and people's demand for meat products continues to increase, the greenhouse orchard management industry has now become one of the main industries of agriculture. The greenhouse orchard management industry has now become one of the main industries of agriculture.

The research on embedded development intelligence series is a hot topic in the current research, but there are still shortcomings such as incomplete technology promotion and incomplete software and hardware. With the improvement of China's economic strength, people's requirements for the quality of life are improving, and the requirements for the working environment are also improving. With the improvement of China's economic strength, people's requirements for the quality of life are improving, and the requirements for the working environment are also improving. orchards, the intelligent management of greenhouse orchards no longer requires a large number of manpower to manage, and improve the efficiency of In order to facilitate better management of greenhouse orchards, the intelligent management of greenhouse orchards no longer requires a large number of manpower to manage, and improve the efficiency of greenhouse orchard management, this study was carried out. orchard management and labour productivity of greenhouse orchard managers, and achieves intelligent intelligent greenhouse orchard management. system collects the required data through various sensors, then analyses the collected data, and finally makes corresponding responses, so as to achieve the purpose of intelligent control of greenhouse orchards, liberate manpower and scientific greenhouse orchard management, and improve the efficiency of greenhouse orchards. efficiency of greenhouse orchards.

1.2 Topic content

The ARM-based controller automatically adjusts the greenhouse orchard management environment. Collect environmental information through temperature sensors, light intensity sensors, etc., collect video information by cameras, and display information on LCD displays; The video monitoring and collected data are displayed to the LCD display in real time and sent to the client at the same time; The server processes and reacts to the The video monitoring and collected data are displayed to the LCD display in real time and sent to the client at the same time; The server processes and reacts to the information collected by sensors and cameras for real-time control (automatic control is carried out through the client setting parameters, and manual (automatic control is carried out through the client setting parameters, and manual control can be carried out in special cases); The client and the server transmit data through the network to realise the interaction of data, the client displays the data information of the interaction with the server, interacts with the server to change the parameters, and remotely controls the hardware ; The database stores the information collected by the hardware, and the information of the management personnel logged in with the client.

1.3 Objectives of the project

(1) Collect the data of the greenhouse orchard management environment to realise the automatic adjustment of the environment.
(2) The camera observes the livestock and poultry activities and monitors the environment in real time.
(3) Managers can monitor the greenhouse orchard management environment locally in real time and can set the environment manually.
(4) Managers can remotely monitor and adjust the greenhouse orchard management environment.
(5) Record and save the environment for later viewing and summarization.

II. Introduction to system equipment and platforms

2.1 Equipment introduction

The ARM series controller mainly includes five parts: the Cortex-M series is mainly used in cost- and power-sensitive applications, the Cortex-R series is mainly used in products with high real-time performance, the Cortex-A series is mainly used in applications with high image and computing. is mainly used in products with high real-time performance, the Cortex-A series is mainly used in applications with high image and computing requirements, the SecoreCore series is mainly used in high-security applications, and the Classic series is more classic applications.

The CortexM0 is a member of the M family of five ARM families. It is mainly composed of various sensors. Including temperature and humidity sensors, light intensity sensors, LED lights, buzzers, etc. for measurement; It can communicate through Zigbee protocol, and can also be connected to the computer through the USB port for debugging; The CortexM0 is a member of the M family of five ARM families. It can communicate through Zigbee protocol, and can also be connected to the computer through the USB port for debugging; The CortexM0 collects data primarily through sensors and transmits the data to the server via the Zigbee protocol. The CortexM0 is shown in Figure 2-1.

Figure 2-1 CortexM0

The CortexA9 is a member of the A family of five ARM families. Using the Linux operating system, it has high performance, can be used as a server, has high stability, and is mainly used for high computing requirements, rich operating system scenarios, and providing interactive and graphical experience. Using the Linux operating system, it has high performance, can be used as a server, has high stability, and is mainly used for high computing requirements, rich operating system scenarios, and providing interactive and graphical experience. The Cortex A9 is shown in Figure 2-2.

Figure 2-2 CortexA9

The camera has the function of photography, it is essentially a device for collecting information, it is through the lens to collect data, by some light sensing devices in the camera to process the data and convert it into an image that can be recognised by the computer, and then through the frame buffer The camera has the function of photography, it is essentially a device for collecting information, it is through the lens to collect data, by some light sensing devices in the camera to process the data and convert it into an image that can be recognised by the computer, and then through the frame buffer technology to turn a frame of data into a video. The LCD display becomes a video by quickly refreshing the data to achieve the effect of deceiving the human eye; The video is displayed by processing the data into a video. The LCD display becomes a video by quickly refreshing the data to achieve the effect of deceiving the human eye; The video is displayed by processing image data, and the font is designed to display text and numbers.

2.2 Platform tools

Linux is a relatively secure operating system with open source code, and developers can learn and modify it as they wish. There are many different versions of Linux, both paid and free, but they all use the Linux kernel. There are many different versions of Linux, both paid and free, but they all use the Linux kernel. And ubuntu is a member of the Linux family, which is a free operating system. The Linux system can be used in a variety of computer hardware devices, from mobile phones commonly used in people's hands, to computers, to various servers and supercomputers. The development of the server took place on a Linux virtual machine and was finally ported to run on a Cortex A9 with a Linux operating system. system.

Qt Creator is a tool for designing interfaces, including various libraries, tools that can be used to design interfaces, based on a variety of encapsulated objects, object-oriented programming ideas, you can use the tools contained in it to draw the interface directly by clicking on the interface components, or you can write your own code to achieve, which is very suitable for software developers to develop.

2.3 Development technology

Framebuffering technology is essentially caching technology, which is to cache a frame of data, and then quickly output these frames of data to form a coherent video.

YUV is a data format that is divided into three elements: "Y", "U", and "V", where the elements represent luminance and chromaticity, respectively. YUV is actually a colour-coding method. actually a colour-coding method.

RGB is a data format, it is based on the three primary colours we often say r (red) g (green) b (blue) to represent colors, it can represent any colour, these colours are regarded as a space, this is the colour space.

JPEG image format is a very widely used photo format, and its advantage is that developers can set the compression ratio to save it according to their own needs, saving space, and it is a very convenient format.

Zigbee is a communication protocol, which is a short-range, low-power communication protocol used in the wireless field. It has a number of features, among which proximity and low power consumption increase its usability, and low complexity and low cost increase its usability. It has a number of features, among which proximity and low power consumption increase its usability, and low complexity and low cost increase its usability. The main application environment is automatic control scenario and remote control scenario, which can be applied in various embedded devices. In short, Zigbee is a very cheap In short, Zigbee is a very cheap communication technology that can be applied in practice.

Socket communication is a commonly used network communication method. Colloquially known as a "socket", it is a unique web application that identifies a unique web application by binding the IP address and port number of the host. Socket communication is a commonly used network communication method. Colloquially known as a "socket", it is a unique web application that identifies a unique web application by binding the IP address and port number of the host. An IP address is the identity of a host on a network, and the port number can uniquely identify an application, so that a communication line is established. TCP protocol is a network control transmission protocol, by establishing a TCP link, you can carry out safe and reliable communication, the client and the server bind the IP address and port port number to identify the communication, through the TCP protocol three packet interaction (three handshakes) to establish the link, after the end of the four packet Select is a multiplexing technique that allows multiple clients to be linked.

Most of today's computers support multi-threaded concurrency, it uses time slice switching, fast switching causes computer users to feel that many programs are executed together, this is because of the deception formed by the CPU processing speed is very fast, like continuous pictures as long as the speed is fast, so it just looks like executing multiple things at the same time, it is essentially only executing one program, but it feels like multiple programs are carried out at the same time.

sqlite is a database that provides some C function interfaces that you can use to manipulate the database, which is very convenient. By using these By using these interfaces, pass some standard SQL statements to the SQLite function, and SQLite will operate the database for you. SQLite is the same as a file, that is, a database is a file, and SQLite is an embedded database that doesn't even require a database to be installed.

III. Requirements analysis and design

3.1 Feasibility analysis

The greenhouse orchard can be made smart and intelligent by collecting data and processing data, and then reacting according to the setting. monitoring and automatic adjustment of the environment can be realised by collecting data from hardware devices such as sensors, the database can store The monitoring and automatic adjustment of the environment can be realised by collecting data from hardware devices such as sensors, the database can store the collected data and the set data, the server can process the data to react, and the client can set and display.

The monitoring of the environment can be achieved through cameras, which are now very common devices that can monitor greenhouses and orchards very well. Environmental information can be realised through sensors, and the more common and cheap sensors include temperature sensors, humidity sensors, light intensity sensors and other sensors, which can well obtain the surrounding environmental information of greenhouse orchards. Environmental information can be realised through sensors, and the more common and cheap sensors include temperature sensors, humidity sensors, light intensity sensors and other sensors, which can well obtain the surrounding environmental information of greenhouse orchards.

The client interface can be developed on the Windows system by using the qt development tool, and then the Linux environment can be developed by The client interface can be developed on the Windows system by using the qt development tool, and then the Linux environment can be developed by configuring the virtual machine under Windows, and the environment chain can be ported to the ARM controller by configuring the cross-compilation tool The ARM controller uses an operating system with Linux kernel and comes with a sqlite database; The design environment and tool techniques can be well implemented. The design environment and tool techniques can be well implemented.

3.2 Requirements analysis

The needs of the greenhouse orchard administrator are to monitor the greenhouse orchard environment in real time, and the environment can be automatically adjusted, so that the administrator does not need to work all the time, liberate the administrator's time, and the administrator can alarm automatically adjusted, so that the administrator does not need to work all the time, liberate the administrator's time, and the administrator can alarm the administrator in case of emergencies, and the administrator can manually adjust the environment; Administrators also need to review historical monitoring information to summarize greenhouse orchard management experience and check greenhouse orchard management.

3.3 Development environment analysis

Operating system: PC Window operating system + ARM Linux operating system.
Database: sqlite database.
Development Tools: Qt Creator.
Development environment: CortexA9, CortexM0

3.4 Functional module design

1, Server Module Design.
(1) TCP server construction.
(2) Processing of each module.
(3) Communication with the client.
2, Camera module design.
(1) Collection of video data.
(2) Processing of video data formats.
(3) Camera control.
3, M0 module design.
(1) Read and write data.
(2) The LCD display is turned on and off.
(3) Real-time display of LCD display information.
4, Client module design.
(1) UI interface design.
(2) Business logic processing.
(3) Communicate with the server network.
5, Database module design.
(1) Data analysis and management.

(2) Secondary encapsulation of basic database operation functions.
(3) The implementation of the database operation interface.

IV. Design and implementation of the system

The system will consist of two parts of the programme, running on the CortexA9 development version of the C language program as a server. Running on the PC qt program as a client program. The system will consist of two parts of the program, running on the CortexA9 development version of the C language program as a server, running on the PC qt program as a client program. Both parts of the program are running in the Linux environment, but the server running programme is after cross-compiled to run The client program mainly includes: accept the real-time data sent by the server, analyze and display data, and interact with the server. The client program mainly includes: accept the real-time data sent by the server, analyze and display data, and interact with the server. The server program mainly has multiple functions: through the Zigbee communication mode to receive the data collected by M0, through the The server program mainly has multiple functions: through the Zigbee communication mode to receive the data collected by M0, through the serial port to accept the pictures collected by the camera, control the LCD display, and interact with the client. The system architecture diagram is shown in Figure 4-1.

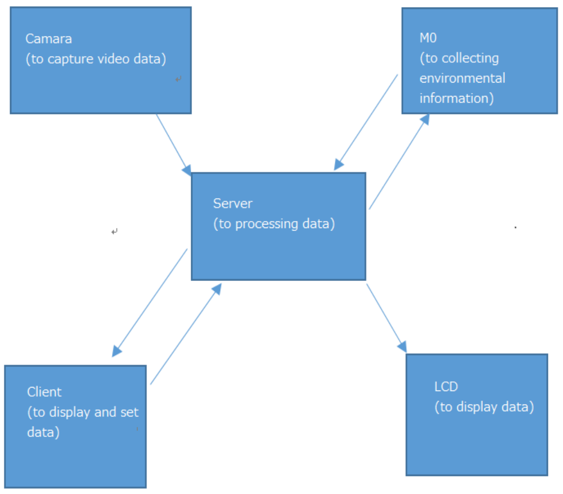


Figure 4-1 System architecture diagram

4.1 Design and realisation of camera functions

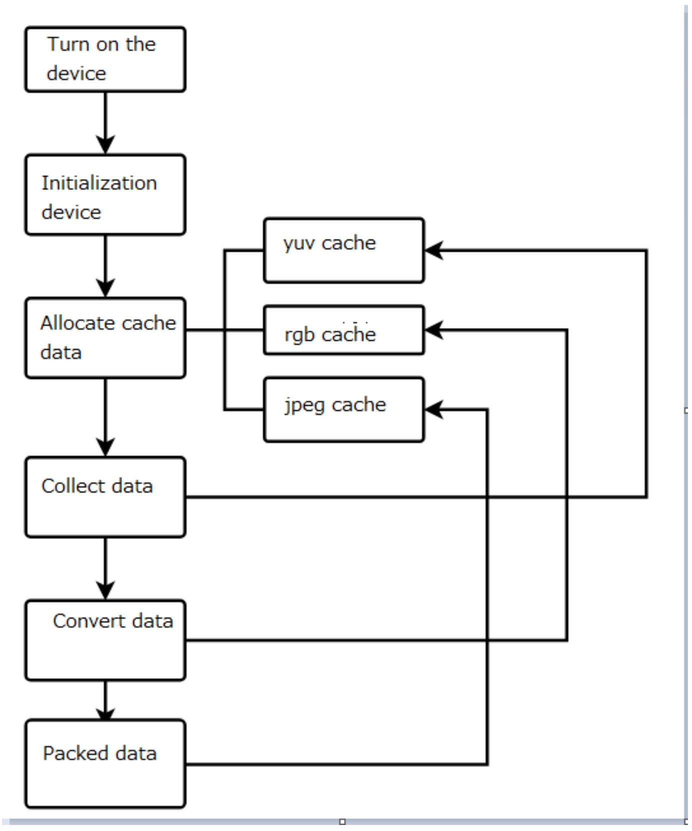
The implementation of the camera module is mainly to control the camera to collect data and process the data for display through code implementation under Linux.

First of all, for the camera first through the Linux library function to collect information about the device, and then according to the device information for the appropriate settings and processing, mainly to view its video capture format, output format, device capabilities, and set their own desired video format; through the function to open the camera, to the camera to allocate space to map the kernel space to the user space, the camera begins to collect data and store data to the allocated space, by setting up a queue to cycle through the image capture, wait for the completion of the capture, to obtain the raw data of the image. Store the data into the allocated space, by setting up a queue for cyclic acquisition of the image, wait for the acquisition to complete and get the raw data of the image.

Processing of the raw data collected. The raw data collected by the camera is yuv format data can not be displayed, need to be converted to rgb format data through the yuv format data can be displayed through the general display equipment; rgb format data need to be converted to bmp format pictures can be viewed through the picture viewer to save.

Because the data need to network transmission and save, rgb and bmp format data is too large, waste of space and reduce network transmission efficiency, so you need to compress the data. rgb or bmp format data processing will be placed in a piece of data in the cache space, by reading this cache to obtain the data through the processing of data compressed into jpg format data, jpg format data through the compression of the data into smaller, can be transmitted through the network to the client for display. The data in jpg format is compressed to a smaller size and can be transmitted to the client through the network for display. Camera flow chart shown in Figure 4-2.

Figure 4-2 Camera Flowchart



- interface design
  int start_v4l2();//initialise camera
  int v4l2();// video data acquisition, transcoding
  long rgb_to_jpeg();//image compression
  int stop_v4l2();//turn off camera
- data design

char yuyv[WIDTH*HIGHT*2];//get video data storage buf
char rgb[WIDTH*HIGHT*3];//store buf after video conversion
unsigned char *jpeg=NULL;//compressed video data stored buf
function implementation:
data acquisition

- int Init_Cameral(int Width , int Hight) //camera initialisation
- – video_fd = open(videoname , O_RDWR);//open device file
- – ioctl(video_fd , VIDIOC_S_FMT , &fmt);//set video format
- – ioctl(video_fd , VIDIOC_REQBUFS , &buf);//request buf
- – mmap(0,buf.length , PROT_READ | PROT_WRITE , MAP_SHARED , video_fd , buf.m.offset);//map to user space
- – ioctl(video_fd , VIDIOC_QBUF , &queuebuffer); //put into cache queue
- int Start_Cameral() //turn on the camera
- – ioctl(video_fd , VIDIOC_STREAMON , &on).
- int Get_Picture(char *buffer)//get picture data
- – select(video_fd + 1, &fds, NULL, NULL, &timeout);//listening to the data situation
- – ioctl(video_fd, VIDIOC_DQBUF, &dequeue);//out of queue
- – memcpy(buffer , yuv[dequeue.index] , dequeue.length); //copy

– ioctl(video_fd , VIDIOC_QBUF , &enqueue); //in queue
• int Exit_Cameral(void) //Exit to free space
• int Stop_Cameral(void) // turn off the device
– ioctl(video_fd , VIDIOC_STREAMOFF, &off).

Processing data

• int yuv_to_bmp() //convert to rgb, bmp
– int yuyv2rgb24(u8 *yuyv, u8 *rgb, u32 width, u32 height) //convert yuyv format data to rgb data format
– void set_bmp_header(struct bmp_header_t *header, u32 width, u32 height) //convert rgb data format to bmp data format

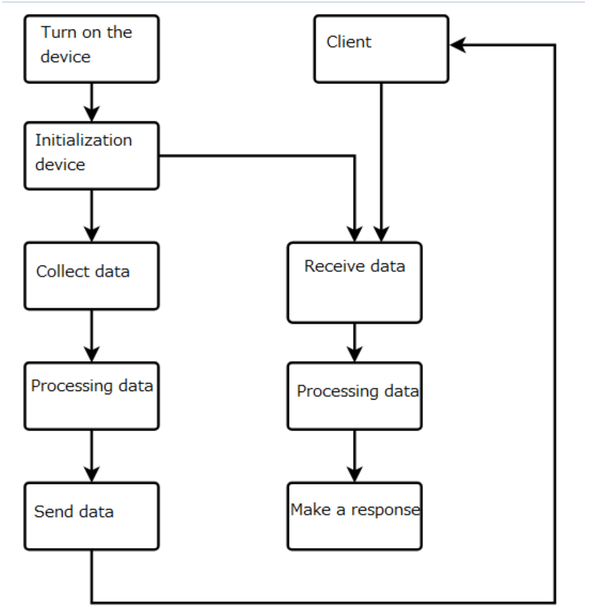
Compressed data

• long rgb_to_jpeg()//compress to jpg
– jpeg_create_compress(&jcs);//set compression parameters
– jpeg_mem_dest(&jcs, &jpeg, &jpeg_size);//get data
– jpeg_start_compress(&jcs, TRUE);//start compressing data
– jpeg_write_scanlines(&jcs, row_pointer, 1); //compression method
– jpeg_finish_compress(&jcs); //finish compression
– jpeg_destroy_compress(&jcs); //free space

4.2 Design and Implementation of Sensor Functions

CortexM0 mainly controls the sensors and LCD display, firstly, it controls the switch of these hardware devices, opens each sensor, and then it starts to collect data, the temperature sensor collects the surrounding environment temperature information, the humidity sensor collects the surrounding environment humidity information, and the light intensity sensor collects the surrounding environment light intensity information, and then it transmits these environment information to the server, and the CortexA9 The server processes and encapsulates the data and sends the processed data back to the CortexM0 to control the M0's led's to light up or go out, and the buzzer's to sound or stop.The M0 flowchart is shown in Figure 4-3.

Figure 4-3 M0 Flowchart



Specific functions:
Serial port open function int UART0_Open(int fd,char *port);.
Serial port receive function int UART0_Recv(int fd,int *tmp,int *hum,int *light);
Serial port initialisation function int UART0_Init(int,int,int int ,int,int);.
Serial port send function int UART0_Send(int fd,char send_op);.

The LCD display mainly displays the information collected by the sensor to the LCD display after processing, including the temperature information of the surrounding environment, humidity information, light intensity information, through the design of the word model to display this information on the LCD display; the camera collects the data through the server to transmit to the LCD display, the LCD after processing the video data to show the video on the display.LCD flowchart As shown in Figure 4-4.

- Specific functional realisation
  - Screen information fetch ioctl(fbfd, FBIOGET_VSCREENINFO, &vinfo).
  - mapping user space mmap(0,screensize,PROT_READ|PROT_WRITE, MAP_SHARED, fd,0);
  - Show picture int LCD_show_bmp()
  - Display Chinese characters // show_data(char *q,int pos,int length);
- interface design
  - int show_hz(double data_w,double data_s,double data_l);//open device /dev/fb0, map user space, handle parameter conversion to char *, call printf_hz function, unmap, close device.
  - int show_data(char *q,int pos,int length); // swipe the character corresponding to *q onto the display
  - int dev_init();//open device /dev/fb0, map user space.
  - int dev_close();//cancel mapping, close device.
  - int LCD_show_back();//Get the picture information and write it to an array, use the array to display the picture.
  - int LCD_show_bmp();//extern call external array variable rgb to display picture.

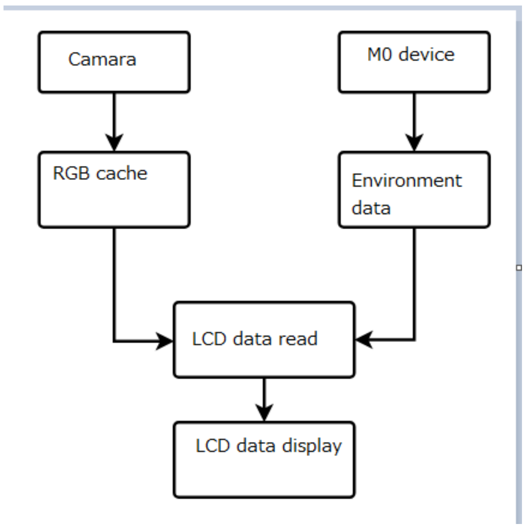


Figure 4-4 LCD Flowchart

4.3 Design and implementation of database functions

The function of a database is mainly to store data, by creating a database, adding data tables, and then performing operations. This is achieved through the interface provided by the database. The data tables are mainly user table for user registration and login, USER as shown in Table 1, environment information table for recording environment temperature, humidity, light intensity, ENVIRONMENT as shown in Table 2, and boundary setting table for determining boundaries, LIMIT as shown in Table 3.

Firstly, plan the required data, design the data table according to these data; connect to the database through the function provided externally by sqlit database, then open the database, encapsulate the function interface twice, design the function required by the server, and the server will process the database by calling the function encapsulated twice to achieve the addition, deletion, modification, and query of the data in the database.

table 1 user

The name of the field	data type	Whether it is empty or not	Other constraints	Remark
user_id	Int	not null	Primary key, auto-increment	Identifier
user_name	varchar(20)	not null		User Accounts
user_password	varchar(20)	not null		User password

Table 2 environment

The name of the field	data type	Whether it is empty or not	Other constraints	Remark
id	Int	not null	Primary key, auto-increment	Identifier
wendu	Int	not null		temperature
shidu	Int	not null		humidity
guangqiang	Int	not null		Light intensity

Table 3 limit

The name of the field	data type	Whether it is empty or not	Other constraints	Remark
id	Int	not null	Primary key, auto-increment	Identifier
wendu_max	Int	not null		Upper temperature limit
wendu_mix	Int	not null		Lower temperature limit
shidu_man	Int	not null		Maximum humidity
shidu_min	Int	not null		Lower humidity limit
guangqiang_max	Int	not null		The maximum light intensity
guangqiang_mix	Int	not null		Lower limit of light intensity

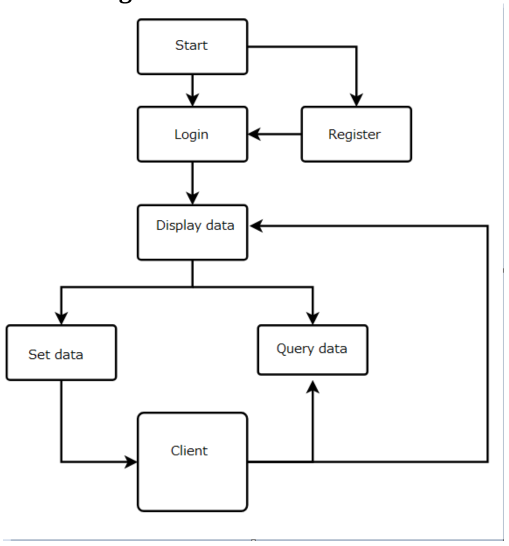
function implementation:
Deletion function for the user information table (logout) int deleteUser(char *userId);
Query function for the user information table (login) int selectUser(struct userInfo **data,int *num);
Update function for the user information table int updateUser(struct userInfo *data,int op);.
Insert function for the environment information table int insertEnv(struct envInfo *env);.
Delete function for the environment information table int deleteEnv(char *time);.
Query function for environmental information table int selectEnv(struct,struct,int *,int);
Update function for the environmental limit setup table int updateLimit(struct limit *data);.
The query function to set the table for the environment limit int selectLimit(struct limit **data);

4.4 Design and Implementation of Client Functions

The client is used for displaying and setting data through qt design interface, and data interaction with the server through network communication. The data collected by the sensor is transmitted to the client through the server, and then the client displays it; the client can also set the data, which is transmitted to the M0 controller through the server to control the device.

The implementation of the function mainly involves the processing of data and interaction with the server. The processing of data mainly includes judging data, encapsulating data and parsing data; the interaction with the server is mainly through socket communication, which transmits the processed data to the server side to regulate the environment and accept the environmental information from the server. The client-side flowchart is shown in Figure 4-5.

Figure 4-5 Client Flowchart



The designed interface mainly includes the registration interface for customer registration, the registration interface has some judgement on the account and password, there is an error box, and then the data will be passed to the server to be stored in the database, so as to achieve the user registration; the login interface, as shown in Figure 4-6, the login interface will be passed to the server to query the data, judge the errors of the account and the password, and then show the login success or failure of the prompt box, so as to achieve the user login; display interface, as shown in Figure 4-7, is used to display environmental information, mainly temperature information, humidity information, light intensity information, to achieve the monitoring of greenhouse orchard environmental information; video display interface, is used to display the greenhouse orchard video monitoring display to achieve real-time monitoring of the greenhouse orchard; setup interface, as shown in Figure 4-8, is used to set up the environmental boundaries, and when the boundaries are reached, the environment will be regulated to achieve automatic environment when it reaches the limit, so as to achieve automatic regulation of the environment.

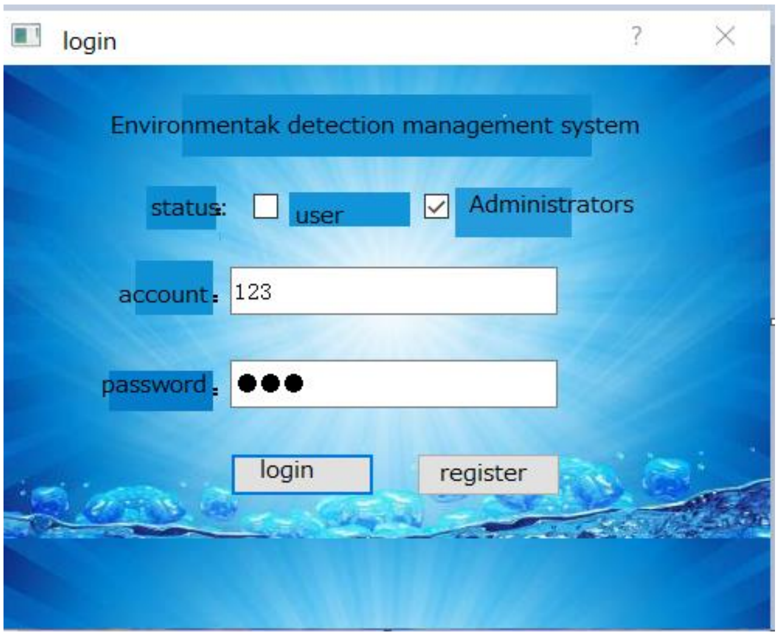

login
?
Environmentak detection management system
status: ☐ user ☑ Administrators
account: 123
password: ●●●
login
register

Figure 4-6 Login Screen

checkChangeUser();// Determine the type of user person, normal user
checkChangeAdmin(); // determine the type of user person, administrator
on_pushButton_login_clicked(); //Judge input format (empty, unreasonable)
void slotConnect(); //send login information
void slotRecv(); //receive server information, determine whether the login is successful or not

on_pushButton_signup_clicked(); //call up the signup screen
Real-time display

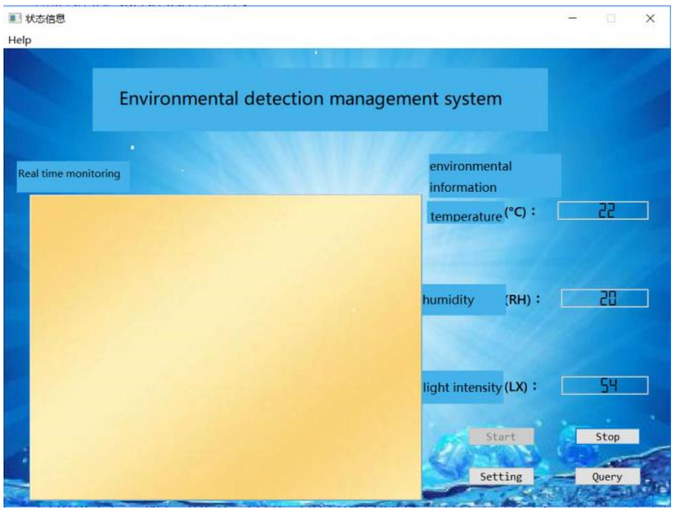
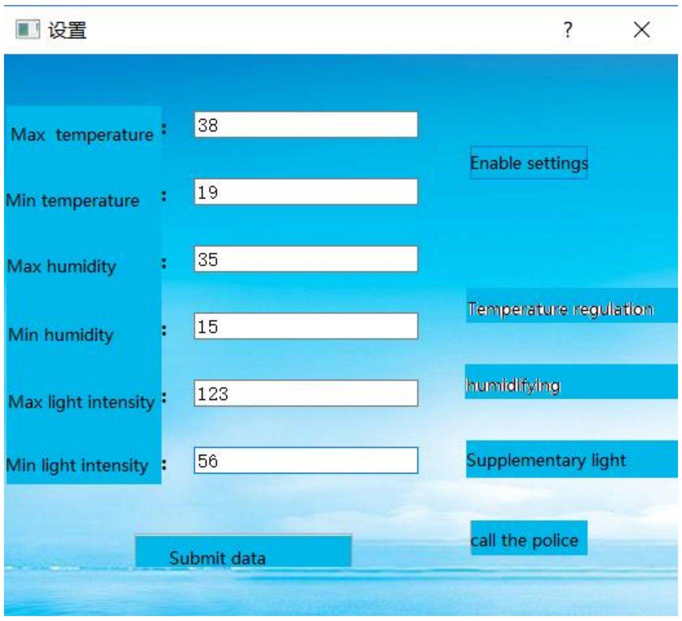


Figure 4-7 Display Interface

void updateM0Info();//update the displayed sensor information
bool dataIsValidJPEG()//determine whether the JPEG image is complete or not
void updatepic();//update video display



Max temperature :	38
Min temperature :	19
Max humidity :	35
Min humidity :	15
Max light intensity :	123
Min light intensity :	56

Figure 4-8 Setting Interface

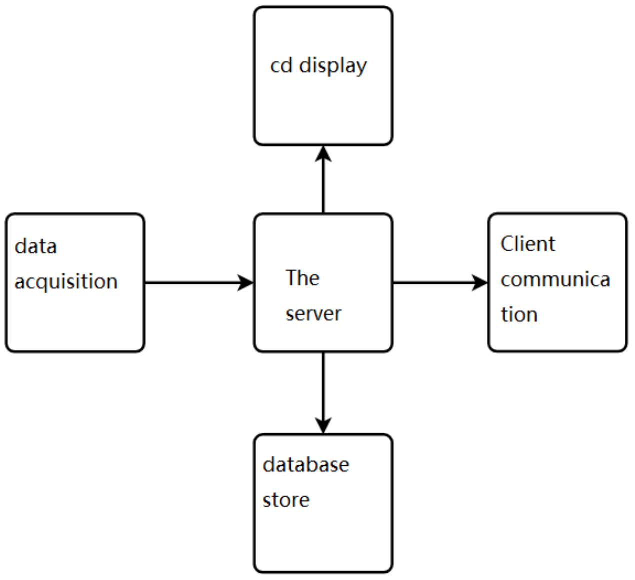
void isSuccess(); //determine whether the operation is successful or not
on_pushButton_submit_clicked(); //Submit environmental alert information
on_pushButton_alARM_clicked(); //alarm
on_pushButton_tempering_clicked(); //tempering
on_pushButton_humidifying_clicked(); //humidifying
on_pushButton_inLight_clicked(); //fill light

4.5 Design and Implementation of Server Functions

The server contacts and organises each functional module through multiple threads, first receiving data from each module, the camera thread receives data from the camera module, the serial port thread receives data from the sensors, the client thread interacts with the client through network communication and then starts processing the data, sends the data needed by each module and saves the data in the database.

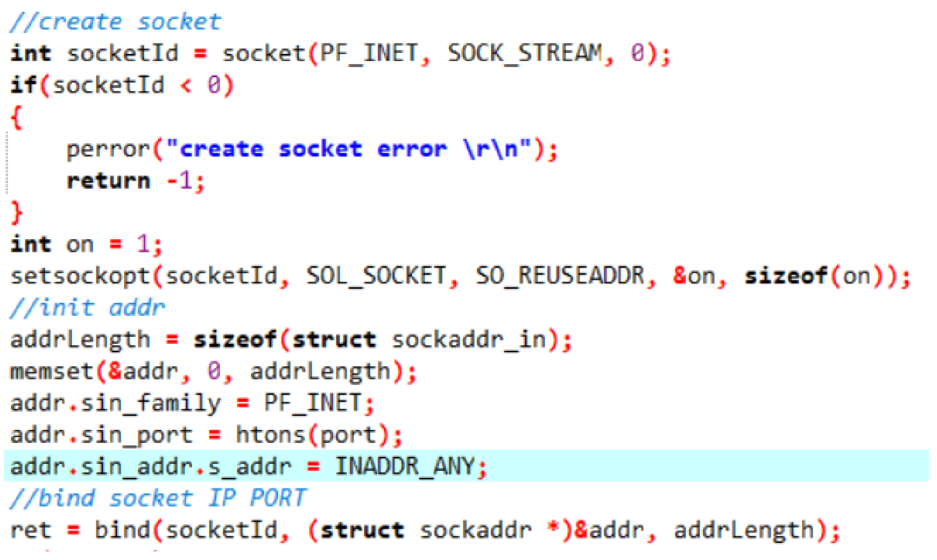
The server uses socket communication with tcp/ip protocol and uses tcp to establish the communication link; first of all, the server local ip address and port port are bound first. Then listen to the client connection, read the data from the client and send the data from the server, the client establishes a tcp connection through the server ip address and port port number and then interacts. The data from the camera module is processed by the camera thread, sent to the LCD display to show and save; the data from the sensor module is processed by the serial port thread, sent to the LCD display to show and save; all kinds of data are sent to the client through the client thread and accept the data from the client; the database thread is mainly used for the storage and finding of data. The server flowchart is shown in Figure 4-9.

Figure 4-9 Server Flowchart



Specific functional implementations:

- Call the interface to get the serial port data
- Serial port data written to database
- Call the interface to get the video image
- Serial data and video images in LCD display
- Establish a link with the client to enable login and registration
- Serial data and video image transfer to the client
- Receive client commands to control buzzers, fans, LEDs and digital tubes.
- Receive client commands to query data information in the database
- void *camera_thread_func(void *arg);// video picture fetching thread
- void *LCD_thread_func(void *arg);//Serial port data as well as the display thread for video images
- void *send_pic_func(void *arg);//serial data sending thread
- void *send_m0_func(void *arg);//send thread for video images
- void *client_deal_func(void *arg);//client request processing thread
- Network socket communication



```
//create socket
int socketId = socket(PF_INET, SOCK_STREAM, 0);
if(socketId < 0)
{
    perror("create socket error \r\n");
    return -1;
}
int on = 1;
setsockopt(socketId, SOL_SOCKET, SO_REUSEADDR, &on, sizeof(on));
//init addr
addrLength = sizeof(struct sockaddr_in);
memset(&addr, 0, addrLength);
addr.sin_family = PF_INET;
addr.sin_port = htons(port);
addr.sin_addr.s_addr = INADDR_ANY;
//bind socket IP PORT
ret = bind(socketId, (struct sockaddr *)&addr, addrLength);
```

V. Project testing and problem analysis

Project testing mainly includes client testing and server testing. First of all, we have to configure the running environment, and then turn on each device and run the client programme and server programme to test the designed functions and see whether they meet the expected standards.

5.1 Preparation of the system environment

(1) Firstly, the compilation environment (installation of cross-compilation tools).
① Source code download: http://ymorin.is-a-geek.org/download/crosstool-ng/
② Unzip the toolchain zip
Downloaded file: gcc-4.6.4.tar.xz
Unzip: tar -xvf gcc-4.6.4.tar.xz
③ Add to environment variables
Modify the file /etc/bash.bashrc to add the following
export PATH=$PATH:/home/farsight/toolchain/gcc-4.6.4/bin
④ Restart the configuration file
$source /etc/bash.bashrc
⑤ Testing of the tool chain
ARM-none-Linux-gnueabi-gcc -v
(2) Simple U-boot burning method
1) Copy U-boot to /root directory (u-boot-fs4412.bin)
pwd to see the directory of u-boot-fs4412.bin
② Enter the following command
dd iflag=dsync oflag=dsync if=/root/u-boot-fs4412.bin of=/dev/sdb seek=1
If the following phenomenon occurs, it means that the burning is successful:
1029+1 records in
1029+1 records out
527104 bytes(527 KB) copied, 5.31438 s, 99.2KB/s
(3) Configuring the tftp server
① Uninstall the tftp server and client: sudo apt-get remove tftp-hpa tftpd-hpa
② Install tftp server and client: sudo apt-get install tftp-hpa tftpd-hpa
(4) Modify the configuration file
① In the /etc/default/tftpd-hpa file
vim /etc/default/tftpd-hpa
Specifying that files can be created requires that TFTP_OPTIONS be set.
② Manually start/stop the service
Stop: sudo service tftpd-hpa stop

Start: sudo service tftpd-hpa start
Restart: sudo service tftpd-hpa restart
Check network service status: sudo service tftpd-hpa status
(5) tftp server test
① Login to the server tftp localhost / tftp ip (host ip)
② Download file from tftp server get filename
③ Upload file to tftp server put filename

5.2 System testing

test step
(1) First of all, set up the environment on the controller sub first
(2) Cross-compile the server-side code
(3) Porting server code to the controller
(4) Switch on each device
(5) Open the client application
(6) Test each function
test function
(1) Log in to each interface of the client to see if there are any problems with the design of each interface
(2) Enter the wrong and correct accounts and passwords respectively to test if the login function works properly
(3) Check whether M0 data and client data are consistent
(4) Setting Boundary Values to see if the environment performs the expected action when it reaches a boundary value
(5) Check if the video display interface is displayed properly
(6) Aim the camera at your surroundings and move it around to see if the screen displayed on the interface matches the screen of the environment that the camera is aimed at.
Test results
(1) The login and registration screens are prompted for success and failure, and are consistent with the account passwords in the database.
(2) The display interface is consistent with the environmental information and can display video in real time.
(3) Setting function in the setting screen allows you to adjust the environmental information.
(4) Consistent LCD display and remote client display

VI. CONCLUSION

This system is designed to achieve the monitoring and automatic adjustment of greenhouse orchard, the intelligent implementation is mainly through the sensor to collect information transmitted to the server, the server by querying the data in the boundaries table in the database to determine the results of the implementation, and the results will be processed into commands transmitted to the hardware facilities to regulate the environment, to achieve the set range of environment, to achieve the intelligent greenhouse orchard management.

This intelligent greenhouse orchard system can manage greenhouse orchard automatically and manually. Mainly through the design and implementation of each functional module, with the completion of a complete system. Through the use of sensor acquisition technology and camera technology to collect real-time environmental information and image information, and use the CortexA9 controller as a server using multi-threading to accept and process real-time data information, through the network communication technology and qt development of the client communication, using a variety of technologies for integrated management.

REFRENCES

[1]. He Yanping. Wireless sensor network networking based on ZigBee protocol[D]. Tianjin University,2009.
[2]. Yu ZG. Research on key technology of embedded Linux-based video surveillance system[D]. Hunan University of Science and Technology,2014.
[3]. Feng Lin. Wireless Sensor Networks and Application of ZigBee Technology [D]. Hefei University of Technology,2006.
[4]. ZHU Rui,LI Zhifei,WU Qingping,LIU Zongxue. A digital image acquisition and processing system[J]. Information Letter,2014(05):58-59.
[5]. Liu X. Design and implementation of embedded WEB server based on ARM platform[D]. University of Electronic Science and Technology,2010.
[6]. YU Zhigang. Research on key technology of embedded Linux-based video surveillance system[D]. Hunan University of Science and Technology,2014.
[7]. Qian Gang. Design of embedded intelligent video surveillance system based on ARM9[D]. Anhui University of Technology,2016.

[8]. LIU Jia. Design and Implementation of Embedded Web Remote Video Surveillance System[D]. Northwest Normal University,2016.
[9]. Shi Lifen. Design and implementation of network camera based on ARM system[D]. Beijing Jiaotong University,2014.
[10]. Li Yuli. Research and application of ARM-based embedded Linux system[D]. Xi'an University of Electronic Science and Technology,2007.
[11]. Lai, H.-L. Real-time network communication based on TCP/IP[J]. Microcomputer and Application,1997(10):39-41.
[12]. Chang, Eddie J., Zhang, Bit-Yong. Design of LCD driver based on Framebuffer[J]. Office Automation,2008(24):30-31.
[13]. Li Ke. Temperature control of greenhouse greenhouse based on embedded [D]. Shenyang University of Technology,2013.
[14]. Song Bohua.Linux device driver development in detail[M]. Mechanical Industry Press,2015:20-25.
[15]. Liu G. Linux system porting [M]. Tsinghua University Press,2013:35-36.
[16]. YANG Haiqing,ZHOU Andong,LUO Yong,CHEN Mu. Design method of LCD display in real-time network communication for embedded system[J]. Computer and Digital Engineering,2010,38(02):155-157+164.