Design And Implementation of Smart Pet House Temperature Monitoring System Based on STM32 Single-Chip Microcomputer

Yuhan Wang *

Department of College of Electronic and Information Engineering, Tongji University, Shanghai, 200092, China

* Corresponding Author Email: 1953899@tongji.edu.cn

Abstract. With the development of social economy and the acceleration of urbanization as well as the increasingly heavier burden of social competition, keeping pets has become an important pastime for more and more families which result in the improvement of the market of pet caring. In order to take better care of pets more efficiently, in this paper, the author designs a smart pet house temperature monitoring system based on STM32 single chip microcomputer with the interaction to the upper computer. The intelligent pet house temperature monitoring system chooses STM32F102C6T6 single chip microcomputer controller as the controlling core and designs several circuits to achieve each required function. The overall designed system realizes the function of pet house temperature measurement with displaying on the LED connected to the single chip and data transmission to PCs with real-time plotting through the MATLAB. Experimental results show that the temperature monitoring system can timely measure the temperature in the pet house, and prove the feasibility of the design. Consequently, the designed system quality can be guaranteed to a great extent.

Keywords: Single chip; STM32; temperature monitoring system; intelligent pet house.

1. Introduction

With the improvement of the quality of people's life in modern society, more and more people choose to keep pets at home, so as to obtain companionship and emotional sustenance, which in turn promotes the prosperity of some related industries [1-2]. However, the pet's body is relatively sensitive to the environment temperature, and either too high or too low the environment temperature will affect the pet's health. Therefore, the comfort level of the pet's living environment is the main focus of the pet owners and is also of great importance to the health of the pets so that it is necessary to have a thermometer which can monitor and display the temperature near the pet house in real time.

In this paper, a smart pet house temperature monitoring system is designed based on STM32 which mainly includes the main control chip module, sensor module, LED dynamic display module, USB interface module, etc. The system can achieve the goal of real-time monitoring of temperature near the pet house, as well as displaying the temperature through the LED, pet owners can timely grasp the temperature changes near the pet house in order to adopt some relevant measurements, and the measured temperature data can be displayed and recorded through the PCs so as to realize the intelligence and accuracy of data collection.

The remainder of this paper is organized as follows: Section 2 introduces the overall design of this system. The design and implementation of the hardware circuit as well as the algorithm programmed in the software are elaborated in Section 3. Section 4 demonstrates the test result of the system through the display on the LED screen, the serial port and also the plot using MATLAB. At the end of the paper, the conclusion as well as some future works are given in Section 7.

2. Overall Structure of the System

In the design, the single chip STM32F102C6T6 is used as the master controller, working with resistances, capacitors, crystal oscillator as well as several other devices, which altogether constitute the minimal system of single chip microcomputer. Based on the minimal system, a few other modules are constructed so as to achieve the desired functions, which include:

1. the temperature measurement module using a Triode PN junction, this module can accurately measure the temperature changes in the current environment and indicate the temperature through the change in the voltage across.

2. the LED dynamic display module, the LED is composed of four-digit eight-section nixie tubes, different nixie tubes can be enabled according to different temperature conditions, so that the corresponding temperature can be displayed on the screen.

3. USB interface module, the USB interface on the single chip microcomputer plays a role not only in 5V power supply, which guarantees the proper functioning and the stability of the entire system, but it is also designed to connect the PC with single chip microcomputer through serial ports, so that the temperature data can be transmitted to the PC and displayed on the serial debugging aides.

The system block diagram is shown in Figure 1.

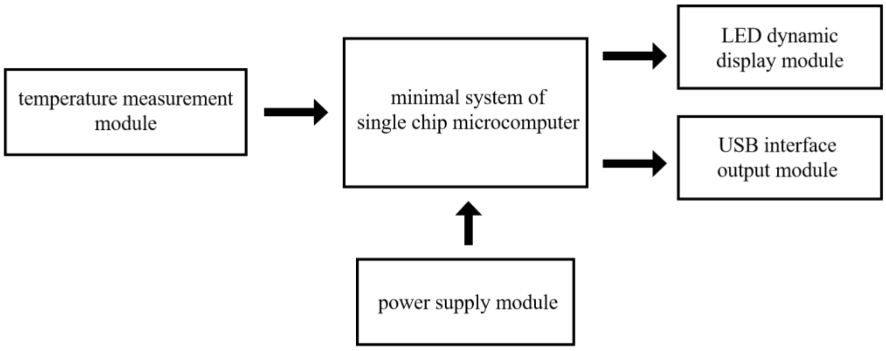

Fig. 1 Block Diagram of the System

3. Hardware Implementation of the Embedded System

3.1. Design of the Schematic Circuit Diagram

3.1.1 On-chip Resources of the Single Chip STM32F102C6T6

In order to correctly and efficiently design the entire embedded smart pet house temperature monitoring system, the choose of a suitable type of single chip is the most important. According to the design of related hardware, the single chip STM32F102C6T6 is chosen for the following advantages. The medium-density USB access line in the chip STM32F102C6T6 takes advantages of the high-performance Arm Cortex-M3 32-bit RISC core which is set at the frequency 48 MHz. The chip possesses a large number of enhanced peripherals and I/Os which can be connected to 2 APB buses [3]. It also characterizes a high-speed embedded flash memory. In order to better communicate with upper computers, widely accepted communication interfaces are provided, which include I2C, SPI as well as USB and USARTs. Additionally, ADC of 12-bits with two multi-purpose 16-bit timers can be made use of to design required functions. It can be applied for applications of numerous purposes, for example, application design and user interface, PC peripherals, alarm systems and so on [4]. Therefore, the single chip STM32F102C6T6 possesses all the resources needed when designing the smart pet house temperature monitoring system, and the hardware circuit of the embedded minimal system is shown in Figure 2.

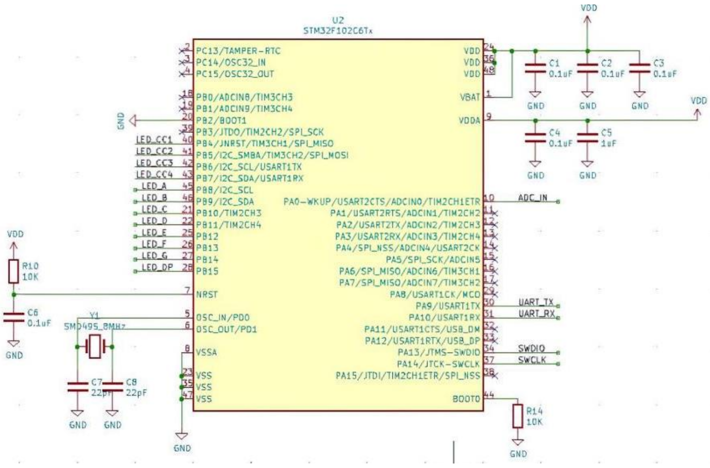

Fig. 2 Circuit of the Embedded Minimal System

As is shown in Figure 2, serial port PB4-PB15 on the main chip is connected to the pins of the LED, pin 5 and pin 6 is designed to connected with the crystal oscillator circuit, pin 34 and pin 37 connecting with SWD and SWC corresponding to NANO board is mainly used for burning the code onto the main chip, and finally the temperature monitoring circuit is designed with pin 10, the ADC_IN, accomplishing the conversion of temperature data.

3.1.2 Design of the Temperature Measurement Circuit

When designing the temperature measurement circuit, a Wilson current source is used in order that the current flowing through the component around the power source is set to be 1mA, the resistors R17, R18, R19 are designed to change the magnification of the op amp, the resistors R21, R22 are designed to biasing the circuit so as to let the curve of the voltage start from the original point [5]. The output of the temperature measurement circuit is at the ADC_IN point which is connected to the main chip to get the value converted. The overall temperature measurement circuit is connected as is shown in Figure 3.

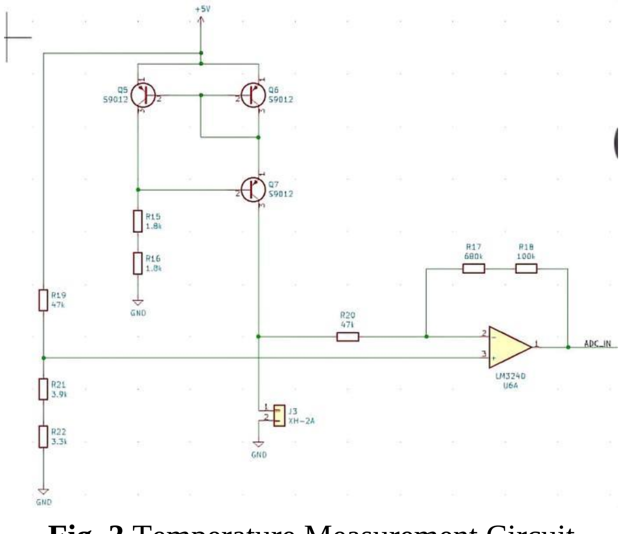

Fig. 3 Temperature Measurement Circuit

3.1.3 Design of the LED Dynamic Display Circuit

In the LED dynamic display circuit, the LED type CC56-12CGKWA is chosen for it can display four numbers at the same time. Because each number is displayed by eight pieces of tubes, the input port a-g as well as the DPX which is connected to the main chip is used to control whether a tube is working nor not so that a correct number is shown on the screen. The ports from CC1 to CC4 are also connected to the main chip to decide which number is lighted to show the corresponding value of temperature through programming. The overall LED dynamic display circuit is achieved as is shown in Figure 4.

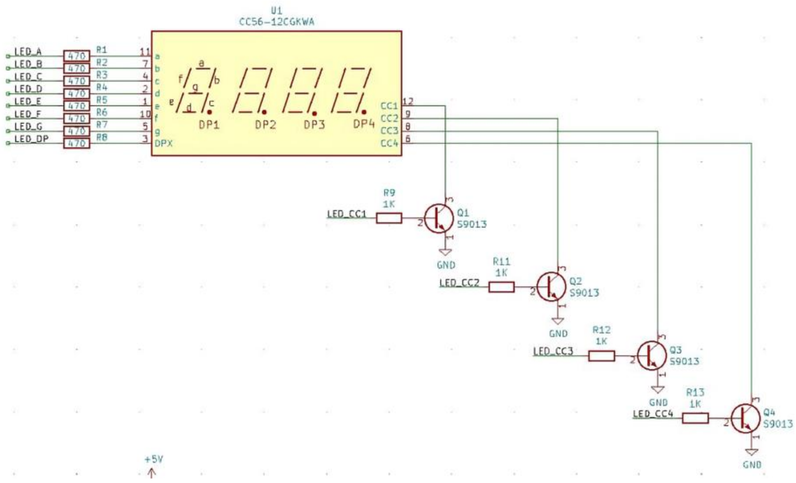

Fig. 4 LED Dynamic Display Circuit

3.1.4 Design of the USB Interface Circuit

In order to make it possible to send the temperature data to the PC and show on the serial debugging aides, the single chip microcomputer is connected to the USB port through serial ports UART_TX and UART_RX. After connecting these ports, the single chip microcomputer will be able to be programmed so that the data can be recorded. The overall USB interface circuit is shown in Figure 5.

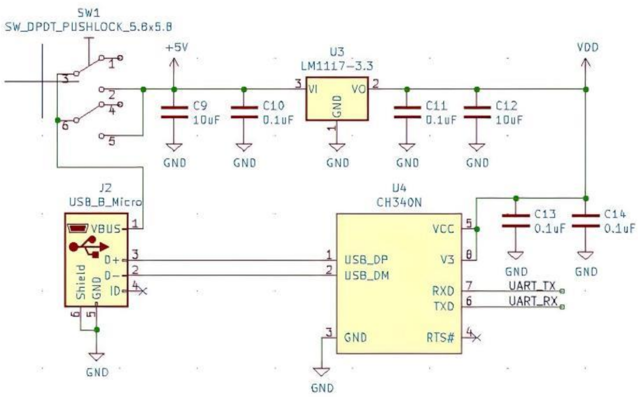

Fig. 5 USB Interface Circuit

3.2. Chip Pin Configuration

3.2.1 Introduction of STM32CubeMX

The pin configuration is achieved through the software STM32CubeMX, a convenient tool with user interface easy to understand. The software also permits the application of C code in order to initialize the micro-processors for the Arm Cortex-M core. Several steps should be followed to finish the processes of the configuration [6].

Several steps are supposed to be taken according to the sequence, of which the first one is selecting the right STM32 controller or processor which is suitable for peripherals, development platform as well as requirements needed. Next, configuration of each necessary embedded software is supposed to be achieved with the assistance of the pinout-conflict solver an utility that configures the peripherals included in the software, as well as the clock-tree setting helper, the power-consumption calculator and the USB or TCP/IP.

STM32CubeMX has a number of characteristics and advantages which include, the pinout has automatic conflict resolutions, the peripherals as well as the middleware functional modes possess dynamic conformation of parameter restrictions for Arm Cortex-M core, the clock tree in it has dynamic conformation of the pin configuration as well as its power sequence can be estimated and offer the consumption results. STM32CubeMX owns the availability and accessibility as complete software operating on Windows, Linux as well as macOS, directing the systems mentioned above and 64-bit Java Runtime environment [7].

3.2.2 Basic Code Configuration on STM32CubeMX

The interface of the pin configuration is shown in Figure 6.

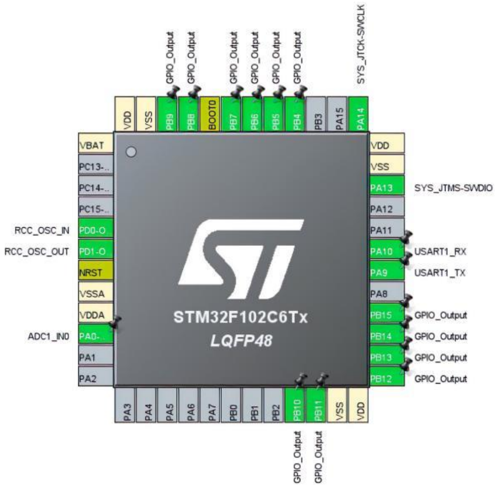

Fig. 6 Configuration on STM32CubeMX

First of all, the mode serial wire is chosen for the main chip. Next, the pin configuration is set according to the function and connection with the circuit of each pin. To be specific, the pin from PB4 to PB15 which is in charge of controlling the LED is set to be GPIO_Output, the pin PA0 is set as ADC_IN which is used for serial output as well as configuring the pin PA9 and PA10 as USART1_TX and USART1_RX, respectively, responsible for receiving and sending the data to the PC. The crystal oscillator is configured automatically to PD0 and PD1. Because of the default frequency in STM32CubeMX is 72Hz, while the maximum frequency applied to STM32F102 is 48Hz, therefore the SYSCLK is supposed to change to 48 Hz accordingly.

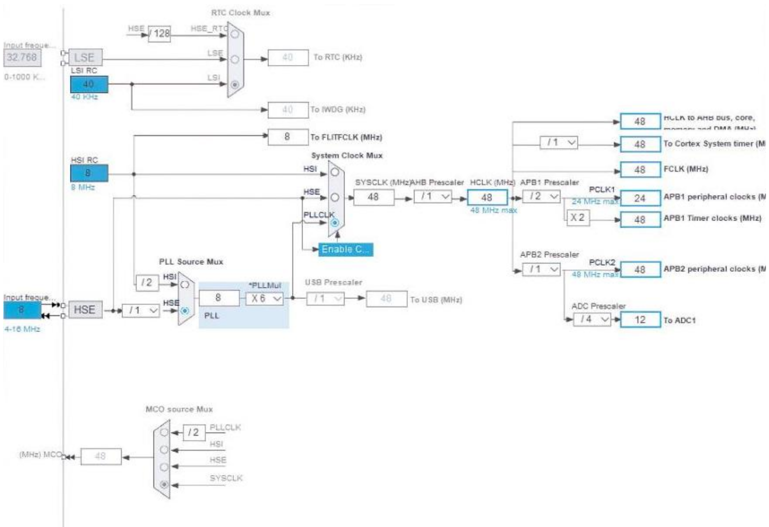

Fig. 7 Changing the SYSCLK

3.3. Program Design of the Software

3.3.1 Introduction of Keil

As to achieve the solution concerning setting up the parameter, developing as well as debugging the whole embedded system based on the Arm micro-controllers, the multi-functional software Arm

Keil MDK is put into utilization. The vision of the software μVision IDE is the most appropriate for it can offer the most excellent performance for Cortex-M-based development [8].

With a ready-to-use database of more than 5,000 Arm-based microcontroller devices, developers can get up and running in seconds, regardless of which MCU they have. MDK's software pack manager and royalty-free middleware components developed for professionals increase productivity and accelerate time to market [9]. Makes software reuse easy. The compiler included in the MDK which can process both C and C++ is at the advancing of performance, code density, architectural accuracy, and safety for Arm bare-metal and real-time OS applications [10].

The flow chart of the overall temperature monitoring system concerning the program development is described in Figure 8. As can be seen in this Figure, after the initialization of each module, ADC_IN port begin to get the value, then the chart of the system goes into the function selection step to tell whether the temperature monitoring circuit is operating in accordance with the requirement. Once the value is obtained the system enters the displaying part both on PC and the LED screen. The whole system will keep working until the power is off.

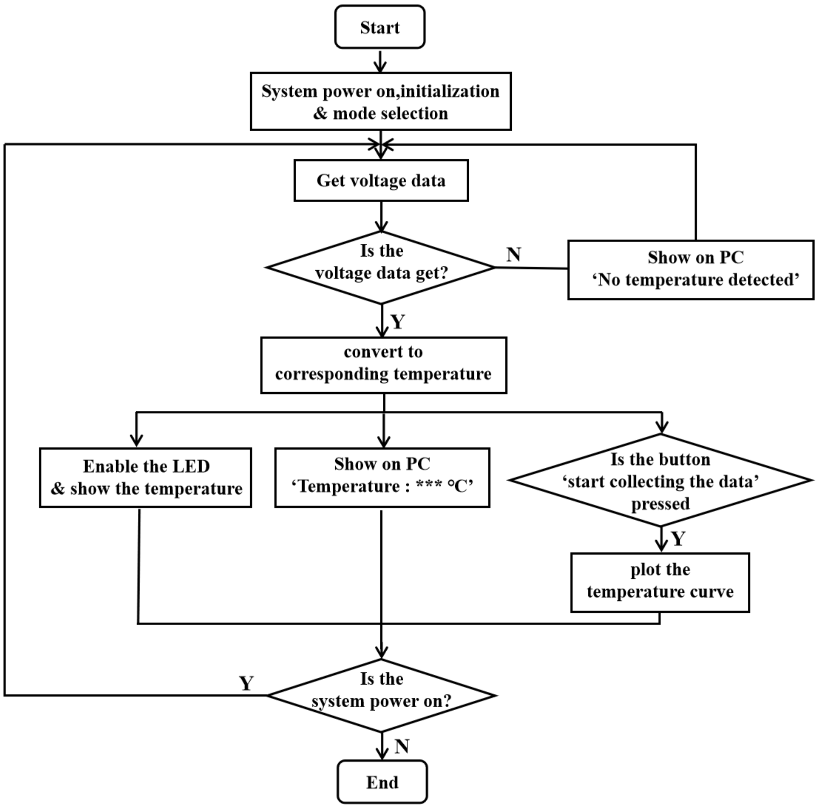

Fig. 8 Flow Chart of the System

The program design of each module included in the system will be described more specifically in the rest of the section.

3.3.2 Preparation of the Marquee

Write the function ShowLED（int num）, controlling the tube to show the number from 0 to 9. The function switch(num) is applied to choose the number shown on the LED screen according to the value of the temperature, and then the function HAL_GPIO_WritePin is used to enable the pin on LED. However, because of the characteristic of the LED component that when different digits are enabled together, the numbers shown on each digit are the same. Consequently, persistence of vision is adopted to address this issue (HAL_Delay(2)) which means each digit on LED is lighten for a literally short time that cannot be perceived by human eyes [11].

After acquiring the temperature value, function GetShowtemp(int temp) will be in charge of displaying the value on the screen. The temperature is shown by the three integers and one decimal format, and each digit of number is stored in an array A= [a1, a2, a3, a4]. Combining all the functions mentioned above and then add them into the loop function while () in the main function, the target of displaying the temperature on LED is achieved.

3.3.3 Obtain the ADC Conversion Value

The voltage value across the triode obtained from the pin ADC_IN is read by the function HAL_ADC_GetValue(&hadcl) and then stored in the variable ‘value’. The value obtained from the function is a certain number between 0 to 4096, therefore, so as to transfer it to the corresponding voltage, the following formula is used.

$$volt=value*3.3/4096 \quad (1)$$

3.3.4 Fit the Temperature Curve

The ADC values corresponding to the temperature of 0-100℃ at an interval of about 1℃ are measured and calculated by curve fitting, the temperature table is as follow in table 1.

Table 1. Corresponding Temperature with ADC_IN Value

Temperature (℃)	1.6	1.8	2.5	2.8	3.2	3.7	4.3	5.3	……
ADC_IN	3902	3889	3858	3840	3804	3770	3670	3613	……
Temperature (℃)	15.0	15.4	16.0	16.5	16.9	17.3	18.4	19.2	……
ADC_IN	2128	2096	2038	2004	1954	1915	1880	1808	……
Temperature (℃)	40.5	41.2	41.8	42.5	43.0	43.6	44.3	44.9	……
ADC_IN	752	737	723	714	702	686	679	669	……

The curve sketched according to the data in Table 1 as well as the curve calculated by curve fitting is shown in Figure 9.

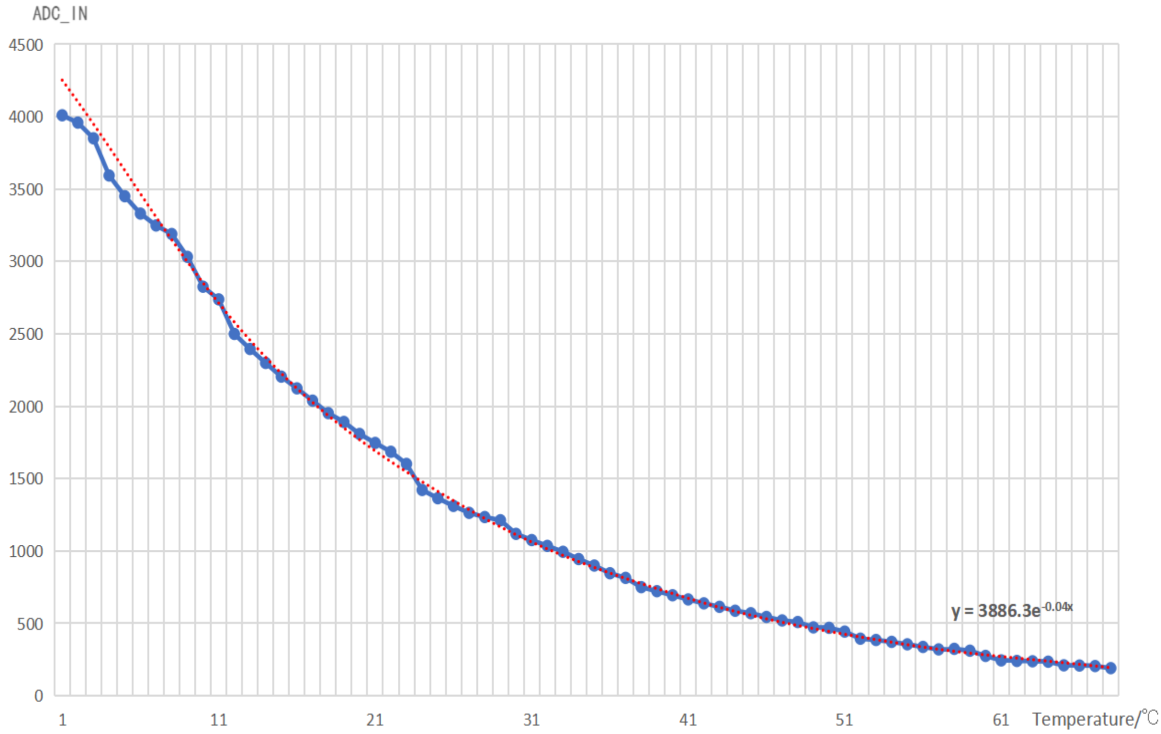

Fig. 9 Temperature Curve

Then, the formula can be expressed as.

$$Temperature= -24.76ln(value) + 204.67 \quad (2)$$

3.3.5 Serial Data Transceiver

After the serial port pin configuration is complete, select the HAL_UART_Transmit function automatically generated in the usart.c file to send data to the serial port [12]. Due to default format of the data transmission, this function can only send uint_8 data, so the data to be sent is written in a self-defined array BUFF [] and output from the serial port.

3.3.6 Plot the Temperature Curve

Before plotting the temperature curve, the data transmitted from the serial port is supposed to written in a txt file. This procedure is completed using MATLAB. The variable ReadData is set in the main function and connected with the serial port COM19. Then it is written the callback function in which the function fopen is used to store the temperature data into the file ReadData.txt as well as Time.txt. Since the serial port is constantly sending data and instructions, the callback function is continually invoked in this program to write the temperature data to the file.

Set up the GUI graphical interface through MATLAB and set the function when the corresponding key is pressed by invoking the callback function of each component. Call the callback function in the image drawing function to continuously read the data sent by the serial port which is stored in ReadData.txt and time.txt, and then plot the data to the corresponding x, y axis, so as to finally achieve the real-time dynamic display of the serial debugging assistant function.

4. Result of the experiment

4.1. Temperature Display on LED

The temperature is divided into three different regions, low temperature (0-10℃), normal temperature (10-80℃) and high temperature (80-100℃), and each of them is tested respectively. The temperature data displayed by the LED is compared with the standard thermometer to check whether it can show the right temperature. An image of the thermometer testing result is illustrated in the Figure 10. It can be seen that the displayed temperature value of the thermometer designed by the development board in the low temperature region and high temperature region is about 1.5℃ in difference, and the temperature difference existing in the normal temperature region can be kept within 0.8℃, so it can explain that the performance of the designed thermometer is acceptable and the gap is within the range that can be used normally, moreover, the value displayed on the LED can updated in real time, changing according to the continuous change of the test temperature.

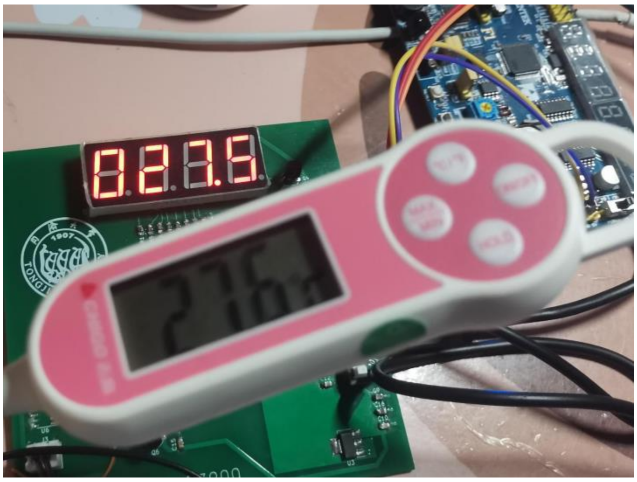

Fig. 10 Thermometer Testing Result

At the same time, the measured temperature data can be displayed in real time in the serial port debugging assistant of the PC. Figure 11 shows a captured image of the testing result on the PC.

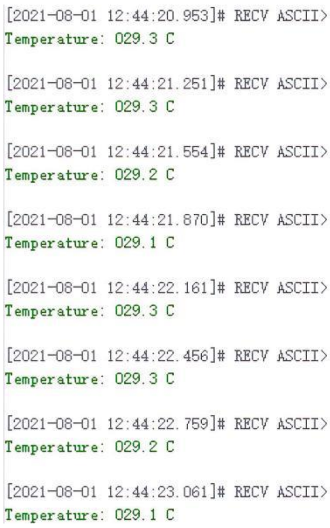

Fig. 11 Testing Result on the PC

4.2. Temperature plot on Matlab

After receiving and saving the temperature data, the MATLAB reads the txt file of the saved temperature and plots it as a real-time changing temperature curve, which can be controlled by setting the start and stop keys. The entire interface with plotting results is shown in Figure 12.

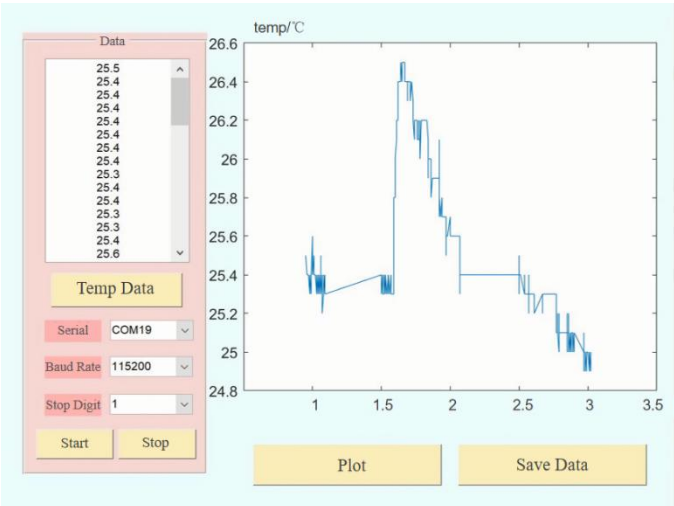

Fig. 12 Plotting the Real-time Temperature on MATLAB

5. Conclusion

In this paper, an embedded smart pet house temperature monitoring system is designed and implemented based on STM32. The hardware constructed the overall system as well as the functions of each circuit with corresponding programs are introduced in detail. The feasibility of the system are tested and verified through several experiments which shows that the designed temperature monitoring system can not only display the correct real-time temperature of the surrounding but it can also transmit the data, plot the data using MATLAB and save it on the PC. The existing value gap between the temperature measured using the designed thermometer and the real temperature are analyzed and explained at the end of the paper as well.

In the future, the designed thermometer can be further improved with more complicated as well as elaborate temperature measurement circuits and components. Encapsulations of all the hardware included in the design are supposed to be done to make the integral system portable and beautiful. The ability of communicating with mobile devices can also be taken into consideration that sending the data using WeChat Mini Program which could make checking the temperature more convenient.

With such improvement, the temperature system could actually be applied into more applications and situations, for instance, monitoring the temperature indoor and instruct the air-conditioner to maintain the temperature with a suitable value.

References

[1] Reniers P W A, Declercq I J N, Hediger K, EndersSlegers MJ, Gerritsen D L, Leontjevas R. The role of pets in the support systems of community-dwelling older adults: a qualitative systematic review. Aging & mental health, 2022.

[2] Kuzma Edson, Sehnem Simone. Proposition of a structural model for business value creation based on circular business models, innovation, and resource recovery in the pet industry. Business Strategy and the Environment, 2022, 32(1).

[3] Chen Jialong, Xiao Jingjing. Design of intelligent temperature and humidity monitoring system based on STM32. Journal of Physics: Conference Series, 2023, 2493(1).

[4] Wang Xinxin, Yun Hongli, Cui Pengxin, Wang Lijun, Niu Jinxing. Design of thermometer based on STM32 and Bluetooth. Journal of Computational Methods in Sciences and Engineering, 2021, 21(5).

[5] Yao Ying. Construction of Intelligent Temperature Control System Based on STM32 Single-chip Microcomputer. Advances in Computer, Signals and Systems, 2023, 7(4).

[6] Liu Meili, Zhang Caizhong. Design of formaldehyde concentration detection system for smart home based on STM32 controller. Journal of Physics: Conference Series, 2021, 1780(1).

[7] Wang Hao, Wang Shudong, Liu Shen, Chen Yong, Li Runqing. Research on Wireless Water Quality Monitoring Method Based on STM32. Journal of Physics: Conference Series, 2021, 1750(1).

[8] Cong Zhang, Zhisheng Wang, Dan Zhu, Haiwen Gao, Nianyu Zou. A lighting control system of art museum based on image recognition and STM32. International Journal of Sensor Networks, 2021, 37(2).

[9] Xi Liu, Qibin Lu. Design and Implementation of Control Unit for Medium and Short-Wave Broadcast Exciter Based on STM32. Journal of Physics: Conference Series, 2019, 1213(5).

[10] Zhang MingHui, Du Juan. Design and development of smart socket based on STM32. Journal of Computational Methods in Sciences and Engineering, 2019, 20(1).

[11] Zhang Chi, Wu Xiao Guang, Zhang Cheng Jun. Design and Implementation of LED Display Screen Controller Based on STM32 and FPGA. Applied Mechanics and Materials, 2012: 268-270.

[12] Yang Fan, Wei Qi, Yang Fan. Design of Smart Home Control System of Internet of Things Based on STM32. Journal of Physics: Conference Series, 2021, 1972(1).