Intelligent Automation & Soft Computing
DOI:10.32604/iasc.2020.012757
Article

Tech Science Press

Design and Development of Unmanned Surface Vehicle for Meteorological Monitoring

Dongli Wu[1], Yunping Liu[2,3,*], Ze Xu[2,3] and Weiyan Shang[4]

[1]CMA Meteorological Observation Center, Beijing, 100081, China
[2]Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology, Nanjing University of Information Science & Technology, Nanjing, 210044, China
[3]Engineering Research Center of Digital Forensics, Ministry of Education, Nanjing, 210044, China
[4]Department of Mechanical and Manufacturing Engineering, University of Manitoba, Winnipeg, R3T 5V6, Canada
*Corresponding Author: Yunping Liu. Email: liuyunping@nuist.edu.cn

Abstract: In view of the paucity of hydrometeorological data, inefficiency and high cost of manual detection, a design scheme of meteorological monitoring unmanned surface vehicle (USV) based on STM32 MCU (Microcontroller Unit, which is also known as Single Chip Microcomputer) is proposed in this paper. The path planning is designed by combining the image data acquired by camera with improved RRT-path algorithm (rapidly-exploring random tree algorithm based on auxiliary path), and then motors are controlled so as to realize autonomous cruise control of USV using the incremental PID algorithm. In addition, the design can also realize real-time monitoring of meteorological data detected by meteorological sensors on the USV through upper computer. Compared with the traditional scheme, the path planning, autonomous cruise is as well as online monitoring function of the scheme presented below enable USVs have full autonomous capability to realize on-line monitoring of meteorological data.

Keywords: USV; autonomous cruise; path planning; online monitoring

1 Introduction

Hydrometeorological data has a wide range of needs in industries such as fisheries, tourism and meteorology, and the demand of the precision and timeliness of hydrometeorological data is substantial increase now. Besides, due to the wide range of waters in our country, the range of hydrometeorological data need to be collected is large. If the method of manual collection of water samples is adopted, which may cause delay in updating the data, the safety of operators is not guaranteed in many harsh environments.

At present, there are many ways to detect hydrometeorological data, but meteorological monitoring USVs are widely used in monitoring hydrometeorological data, because of their high flexibility, expandability and support for online monitoring. USVs are divided into three types, which are namely remote-controlled USV, semi-automatic USV and fully automatic USV, and the development direction of USVs is mainly the latter two now [1,2]. So, in order to achieve a highly automated unmanned ship, it should have the function of autonomous path planning and autonomous cruise.

The path planning algorithms used by the robot include A* algorithm, artificial potential field algorithm, mean field annealing algorithm and RRT algorithm et al. The RRT (Rapid-extended Random Tree) algorithm is most suitable for use in nonholonomic motion planning and motion planning for multi-self-robots [3–5]. But it has the disadvantages of lacking stability and slow convergence [6]. This paper proposes a fast-spreading random tree algorithm based on auxiliary paths to improve the above disadvantages.

Online monitoring of meteorological data has great advantages, and one of the key points of this technology is data transmission [7]. After studying various communication technologies such as Beidou satellite, digital radio and GPRS [8–11]. We found that the 4G mobile communication technology can meet requirements of low use-cost, long transmission distance, high transmission speed and low latency [12]. Therefore, this design will use 4G mobile communication technology to achieve the goal of online monitoring.

To sum up, this paper proposes a meteorological unmanned ship with path planning, autonomous cruise and online monitoring. Firstly, the path planning function is implemented by the RRT-path algorithm. Secondly, the autonomous cruise function is implemented by the incremental PID algorithm. Thirdly, the online monitoring function is mainly using 4G mobile communication technology to implement wireless transmission function. Finally, the upper computer is built by LabVIEW, which has the ability to extract, display, process and store meteorological data.

2 Overall Design

The design of meteorological monitoring USV proposed in this paper is mainly composed of three subsystems, which are USV path planning system, USV autonomous cruise system and USV online monitoring system. The USV is designed by stm32 MCU, which is connected with the motor, camera, ultrasonic sensor, electronic compass and various meteorological sensors. Firstly, the STM32 uses an ultrasonic sensor to capture the distance between the USV and the obstacle, then it uses the OV7725 camera to collect the information of obstacle. After the collected information being sent to the server through the data transmission system for image processing, the path planning is performed by using the RRT-Path algorithm. In the end, the planned path is sent to the STM32 which through the data transmission system to let the USV execute the path. Secondly, the STM32 uses the electronic compass to calculate the coordinates of the USV and the angle between the USV and the route. Then it uses the incremental PID algorithm to control the speed of the motor which can achieve the goal of the autonomous cruise. Finally, the STM32 can use meteorological sensors to collect meteorological data such as temperature, humidity, air pressure, wind speed, and wind direction. And the data will be sent to the upper computer through the data transmission system. What's more, the data transmission system is built by 4G DTU, and the upper computer has the ability to extract, display, process and store meteorological data. The diagram of the overall framework is shown in Fig. 1.

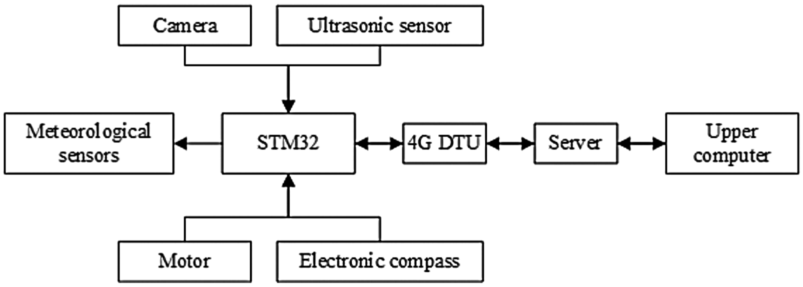


Figure 1: Diagram of the overall framework

3 Design of Path Planning System

3.1 Overall Design of Path Planning System

In the path planning system, the system uses the OV7725 to acquire image and use the HC-SR04 ultrasonic sensor to measure the distance between the USV and obstacle. When the distance is within the safe range, the USV is not threatened by obstacles. But when the distance between the obstacle and the USV is less than the preset threshold, the STM32 MCU controls OV7725 camera to collect the obstacle image and transmits the obstacle information to the server via the 4G DTU. After the path planning is performed by the RRT-path algorithm on the server, it is retransmitted to the STM32 MCU via the 4G DTU. Then the USV can start the obstacle avoidance navigation according to the planned path. The structure of path planning system is shown in Fig. 2.

The specific process of path planning system operation is shown in Fig. 3.

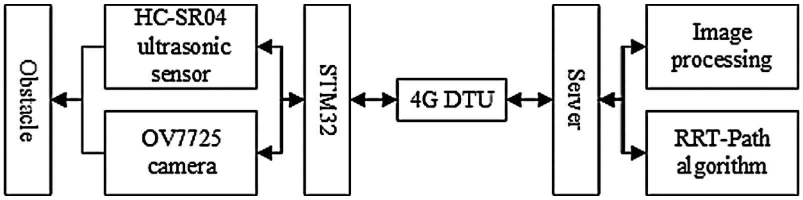


Figure 2: Structure of path planning system

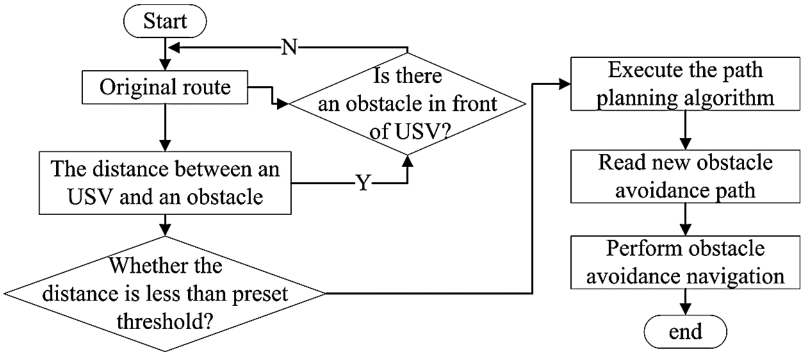

Start
N
Original route
Is there an obstacle in front of USV?
The distance between an USV and an obstacle
Y
Whether the distance is less than preset threshold?
Execute the path planning algorithm
Read new obstacle avoidance path
Perform obstacle avoidance navigation
end

Figure 3: Process of path planning system operation

3.2 Algorithm of RRT-Path Path Planning

S.M. LaValle proposed rapid-expansion random tree (RRT) algorithm in 1998. This algorithm can perform path planning without environment construction, but the basic RRT algorithm is slightly cumbersome, heavy workload, high search cost and low stability in the stage of constructing random tree. Therefore, in order to improve the efficiency of path planning, this system uses the RRT-Path algorithm which is based on the auxiliary path to plan the path. The algorithm uses an auxiliary path that attracts trees growing along the path from the initial configuration to the target configuration. Besides, this algorithm avoiding the purposeless exploration of the global environment and reducing the computational cost to save search costs.

The basic idea of the algorithm is described below. Firstly, system constructs a feasible auxiliary path from the starting point to the target point by using the obstacle avoidance algorithm. Secondly, the system explores and randomly samples the surrounding environment along this path and guides the random tree to grow towards the target point. It avoids the purposeless sampling in the global environment to build a random tree, which improves the efficiency of the algorithm greatly. Finally, when the random tree is close enough to the target point, the target coincides with the target on the auxiliary path. Then the random tree reaches the target point, and the auxiliary path completes the guiding task which means the end of path planning. The state of RRT-Path algorithm extended phase diagram is shown in the Fig. 4.

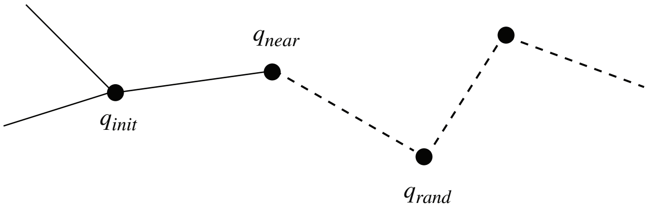


Figure 4: The state of RRT-Path algorithm extended phase diagram

The specific workflow of the algorithm is shown below. First of all, the path planning function PlanPathRRT() and the tree RRT is initialized to add the location of the USV to the tree as the initial state, that is, which is the root node. Then, according to the auxiliary path, when the random node generating function generates a random node around the path, the collision detecting function InCollision_Node() is used to check whether the point qrand collides with an obstacle in the space. If the point is far from the obstacle, system will find the node qnear closed to the point in the tree. Then the collision detection function InCollision_Edge() will check whether the straight path between the sampling point qrand and the nearest point qnear in the tree has a collision. If the straight path doesn't collide, system will use the node add function AddNode() to directly add the sample point qrand to the tree which use the closest point qnear as its parent node. Else if the straight path collided, this point will be thrown. What's more, the system uses the step size adjustment function each time to take the minimum step size as a criterion for generating random points.

Finally, a node is added to the tree each time. When the distance of the node from the target position qgoal is less than a certain threshold distance, the system will check whether the target point qgoal can arrive from the added node in a straight path. And if the target location is reachable, system will add the target location qgoal to the tree with the nearest node as its parent. At this point, the path planning is complete. Besides, if the target location is still unreachable, the system will sample other points.

3.3 Process of Path Smoothing

After the system finds the path, it can apply path smoothening to make a smoother and more direct target route. Firstly, the sample points between nodes on the entire path are iterated by using the path smoothening function which is SmoothPath(). Secondly, the system checks whether two points can be connected directly to reduce the length of the entire path. Thirdly, the system makes the path shorter and smoother by removing the redundant points between the two points and connecting the two points directly. As shown in Fig. 5, the red path isn't smoothed and the green path is smoothed.

3.4 Simulation and Analysis

This paper uses MATLAB to design a simulation platform. Firstly, the starting point of the experimental environment is represented by a green circle, and the target position is represented by a red circle. Secondly, the obstacle is represented by a blue circle. Thirdly, the area where the USV will sail is the area surrounded by a thick black line. Fourthly, the side length of each grid is set to 1, and the radius of the circle is set to 0.5. Fifthly, the coordinate of starting point is (–3, 10), and the coordinate of target position is (1, –4). Sixthly, three rectangles composed of blue circles represent obstacles, and USV can only sail in the area restricted by the thick black line. In the simulation, the random expansion tree is represented by a pink line that the width is 2; and the unsmooth path is represented by a red line that the width is 3. Besides, the smoothed path is represented by a green line that the width is 5. The specific working environment of USV is shown in Fig. 6.

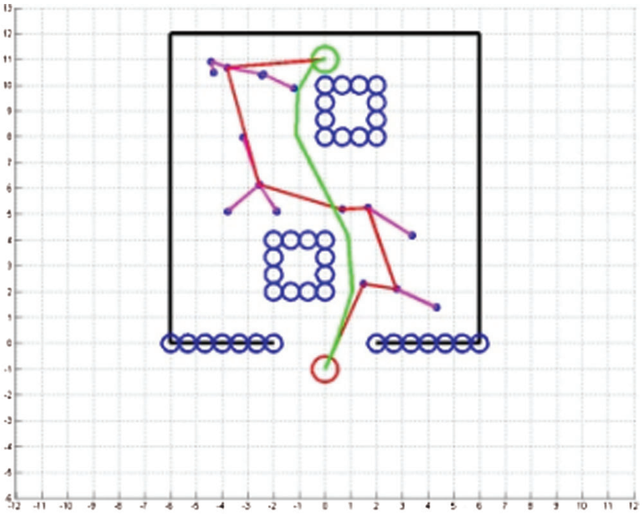


Figure 5: The effect of path smoothening

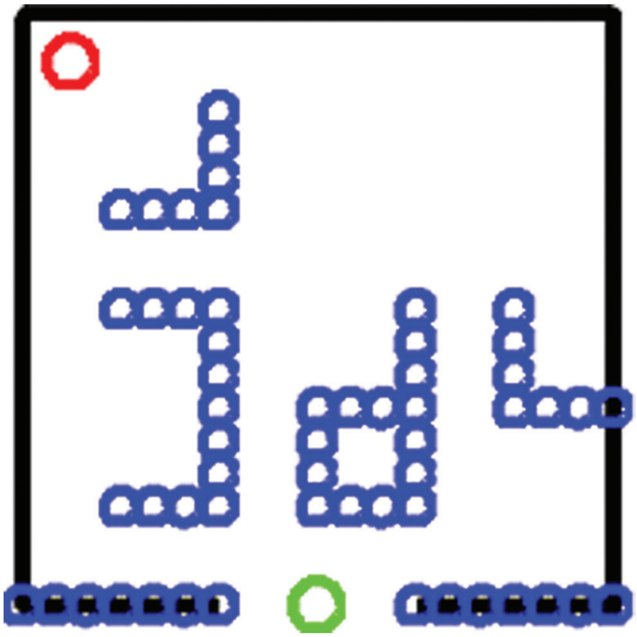


Figure 6: The working environment of USV

When the random step length (param.res) is equal to 0.3 and the distance parameter (param.thersh) between USV and destination is equal to 0.5, the planned path (red) and the path (green) smoothed by path smoothening function are shown in Fig. 7a (The number of times is 619). As can be seen from the planning results, the algorithm overcomes the drawback of being trapped in a local minimum.

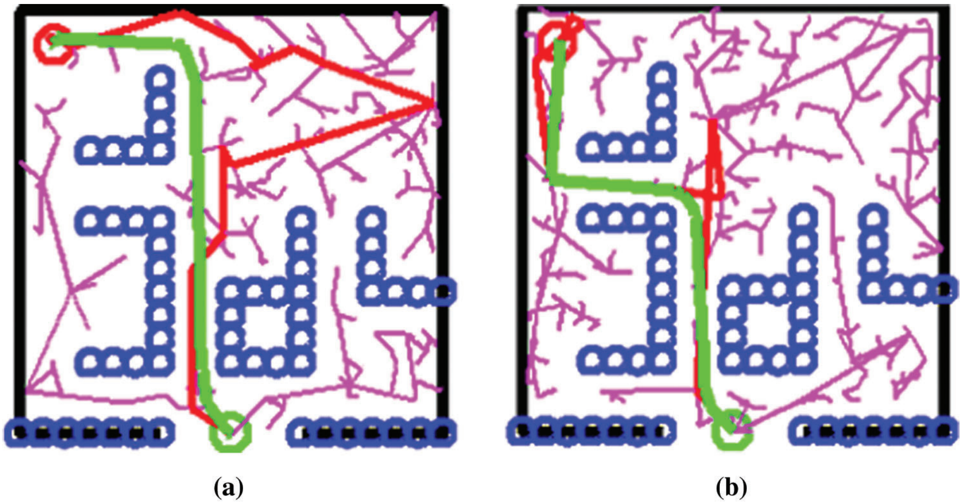


Figure 7: (a) The path before and after path smoothening process and (b) Path after planning again

When the path is planned again under the same environment and same parameters, a different path will be obtained, the path being shown in Fig. 7b (the number of iterations is 748).

It can be seen from the above two simulations that the smoothed path shortens the path length, making the path from the starting point to the target point shorter and smoother. But the repeatability of the planning is poor.

When the random step size (param.res) is equal to 0.7, and the distance parameter (param.thersh) between USV and destination is equal to 0.5, the smoothed path is shown in Fig. 8 (the number of iterations is 399).

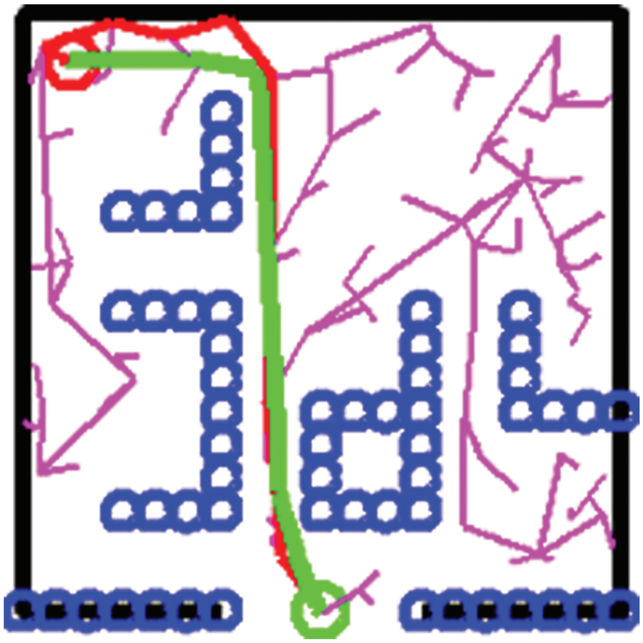


Figure 8: The results of the second simulation experiment

It can be seen that the probability of decreasing the number of iterations is inversely proportional to the increase of the random step size.

When the random step size (param.res) is equal to 1, and the distance parameter (param.thersh) between USV and destination is equal to 0.5, the situation of fail to avoid obstacles is shown in Fig. 9.

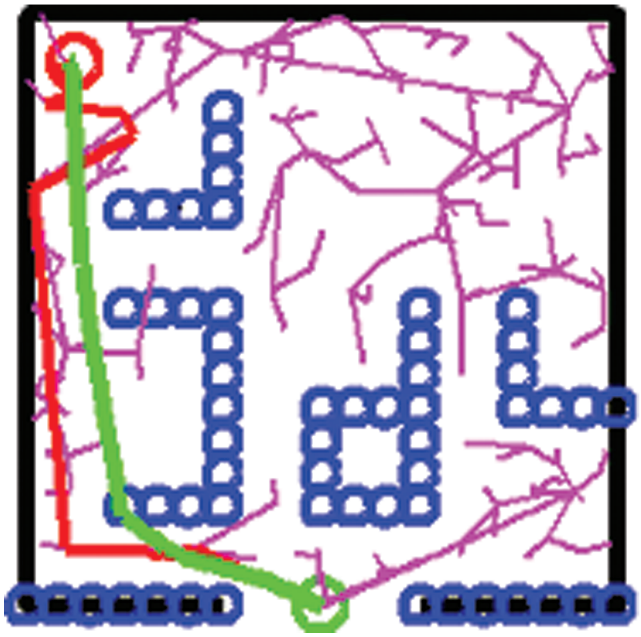


Figure 9: The case of fail to avoid obstacles

If the random step size is larger than the free space range that can be navigable, when the extended random tree finds the path, the path will directly pass through the obstacle which means the algorithm loses the ability to avoid obstacles. Therefore, the random step size needs to be set reasonable by the distance between USV and obstacles.

4 Design of Autonomous Cruise System

4.1 Overall Scheme Design of Autonomous Cruise Control System

The autonomous cruise system uses STM32 MCU and electronic compass to calculate the coordinates and heading of USV. Then the MCU calculates the angle between the heading of USV and the target route. Finally, the incremental PID algorithm is used to precisely adjust the state of motors, and USV is always allowed to sail along the target route. The block diagram of overall design is shown in Fig. 10.

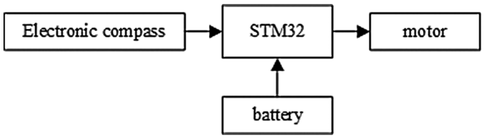


Figure 10: The block diagram of overall design

The design flow of this system is as follows. Firstly, the STM32 MCU reads coordinates acquired by the electronic compass after the system is powered. Then MCU processes the angle between the heading of USV and the target route. If the direction angle is greater than the preset value, the MCU will adjust the duty cycle of the PWM which is equal to adjust the motor speed. If it is not greater than preset value, the MCU will continue to acquire coordinates. This is easy to implement [13]. The overall flow chart of software design is shown in Fig. 11.

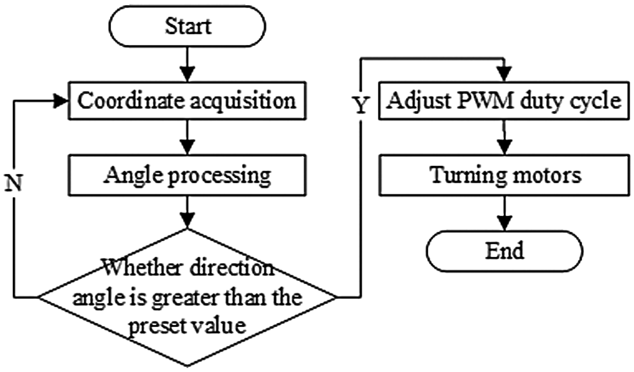


Figure 11: Overall flow chart of software design

4.2 Algorithm of Incremental PID Control

The incremental PID control algorithm is a control algorithm that performs PID control on the increment of the control amount. In general, the basic PID formula is shown in formula (1):

$$u(t) =k_p[e(t)+\frac{1}{T_i}\int_{0}^{t} e(t)dt+T_d\frac{de(t)}{dt}] \tag{1}$$

In the above formula, the *u(t)* is the output of controller, and the *e(t)* is the input of controller. The *Kp* is the proportional amplification factor of controller. The *Ti* is the integral time of controller. And the Td is the differential time of controller.

Let *u(k)* be the output value of the controller at the kth sampling time. So, a discrete PID formula can be deduced.

$$u(k) = K_pe(k) + K_i\sum_{j=o}^{k} e(j) + K_d[e(k) - e(k-1)] \tag{2}$$

In the above formula, $K_i = K_pT/T_i$ is the integral coefficient and $K_d = K_pT_d/T$ is the differential coefficient. The formula of incremental PID can be deduced by formula (1). It is shown in formula (3).

$$\begin{aligned} &\Delta u(k) = u(k) - u(k-1) = K_p\Delta e(k) + K_ie(k) + K_d[\Delta e(k-1)] \\ &\Delta \mathrm{e}(k) = e(k) - e(k-1) \end{aligned} \tag{3}$$

Furthermore, we can rewrite the formula (3) into the formula (4).

$$\Delta u(k) = Ae(k) - Be(k-1) + Ce(k-2) \tag{4}$$

In the above formula, $A = K_P(1 + T/T_i + T_d/T)$, $B = K_P(1 + 2T_d/T)$, $C = K_pT_d/T$.

If the appropriate coefficient is selected, it is only need to calculate deviation values a few times before. Then the increment *(Δu(k))* can be deduced from the formula (3) or the formula (4).

4.3 Implement Incremental PID on STM32

Formula (1) is written as a form that is easy to program in STM32. It is shown in formula (5).

$$\begin{aligned} &U(k) = U(k-1) + K_p[e(k) - e(k-1)] + K_i e(k) + K_d[e(k) - 2e(k-1) + e(k-2)] \\ &\Delta U(k) = U(k) - U(k-1) \end{aligned} \tag{5}$$

The system uses PWM wave to control motors. The output of the PWM wave in STM32 MCU is equal to output by TIM3 timer. And the USV only needs two motors to drive. So, the TIM3 timer will directly outputs two PWM waves to drive USV. What's more, the MCU uses TIM1 timer and TIM4 timer to capture two feedback signals. And use TIM6 timer for PID sampling and processing.

The specific control process is as follows. Firstly, the MCU set initial values of the PID algorithm to calculate the current error, and the values are *Kp*, *Ti* and *Td*. If it is within the allowable range, the incremental calculation (*ΔP*(*k*) = *P*(*k*)–*P*(*k*–*1*)) will be performed, and the PID will calculate sampling values. Then MCU will store error values for next calculation. If the sampling time is reached, the interrupt will be updated after the interrupt processing. Then the TIM1 timer and TIM4 timer will start capturing PWM waves and calculating the frequency. Finally, MCU will output PWM duty cycle. The duty cycle in this design is represented by a number [14]. The flow chart of PID program is shown in Fig. 12.

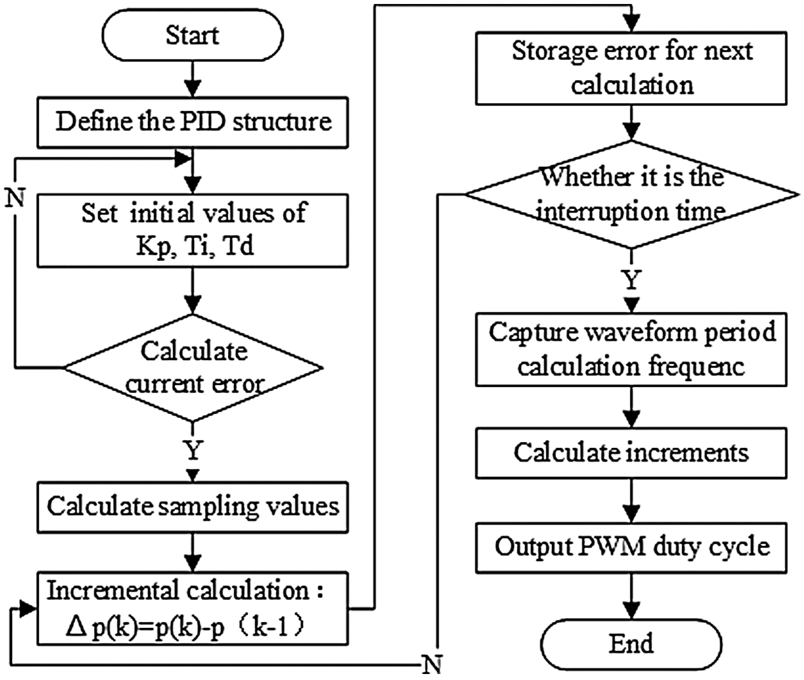

Start
Define the PID structure
N
Set initial values of Kp, Ti, Td
Calculate current error
Y
Calculate sampling values
Incremental calculation : Δ p(k)=p(k)-p（k-1）
N
Storage error for next calculation
Whether it is the interruption time
Y
Capture waveform period calculation frequenc
Calculate increments
Output PWM duty cycle
End

Figure 12: The flow chart of PID program

5 Design of Online Monitoring System

5.1 Overall Plan of Online Monitoring System

The data acquisition system of online monitoring system is made up of STM32 MCU, DHT11 temperature and humidity sensor, BMP180 air pressure sensor and HS-FS01 wind speed and wind direction sensor. The communication system is made up of 4G DTU, base station, server, virtual serial port software and VISA serial communication interface. The data monitoring system is made up of the upper computer which is built by LabVIEW. Besides, the adapter board is used to connect the MCU with the 4G DTU which adopts the USR-G780. And the communication protocol uses the TCP protocol to ensure the stability of data transmission.

The working process of online monitoring system design is as follows. Firstly, STM32 MCU is connected to each sensor and receives the meteorological data which are collected by meteorological sensors. Secondly, the adapter board will convert the TTL level outputted by the MCU into an RS232 serial port signal and send it to the 4G DTU. Thirdly, the 4G DTU packs the data and sends it to the base station. Fourthly, the server obtains the data from the base station and sends it to the virtual serial port software for data conversion. Then the upper computer can process the received data and send control signal to MCU. The flow chart of online monitoring system is shown in Fig. 13.

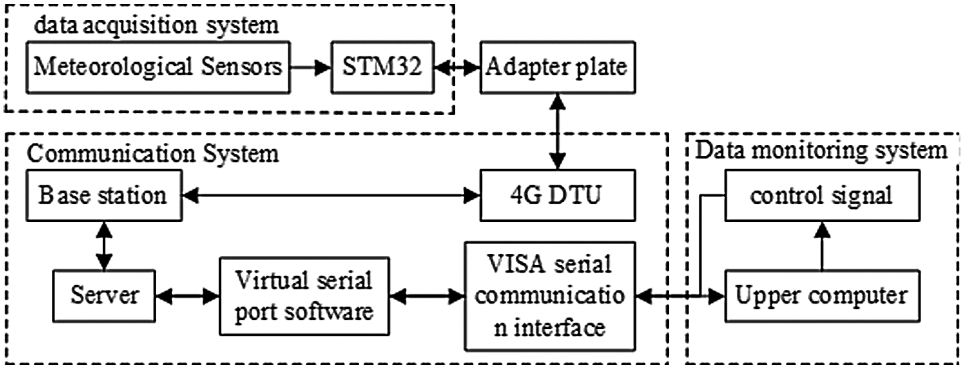


Figure 13: The flow chart of online monitoring system

5.2 Design of Data Acquisition System

The data acquisition system can acquire meteorological data such as temperature, humidity, air pressure, the wind speed and wind direction. The system adopts DHT11 to acquire temperature and humidity. The system adopts BMP180 to acquire air pressure. And the system adopts HS-FS01 to acquire wind speed and wind direction.

The DHT11 temperature and humidity sensor communicates with the STM32 MCU via a single bus. The BMP180 air pressure sensor communicates with the STM32 MCU via the I2C protocol. It uses real-time temperature to compensate for the air pressure value, improving accuracy. HS-FS01 sensor is a wind speed and wind direction sensor of voltage output type.

The operation process of the data acquisition system is as follows. Firstly, the STM32 MCU performs an initialization operation. After the initialization is completed, the MCU starts reading the data of each sensor. When the MCU reads all the data, it will integrate all the data and output the integrated data through the serial port. Finally, the 4G DTU and the STM32 system board are connected together using an interposer board that converts TTL levels and RS232 signals.

5.3 Design of Communication System

When the STM32 MCU sends the serial port data to the USR-G780 (4G DTU), the USR-G780 can convert the serial port data into a TCP packet and send it to the base station. Then the data will be sent from the base station to the server, which will send the data to the virtual serial port software. Finally, the virtual serial port software converts the TCP data into serial port data and sends it to upper computer. The transmission format of the data between the USR-G780 and the virtual serial port software is TCP data packet, and the transmission format between the virtual serial port software and the upper computer is serial port data. The overall architecture of communication system is shown in Fig. 14.

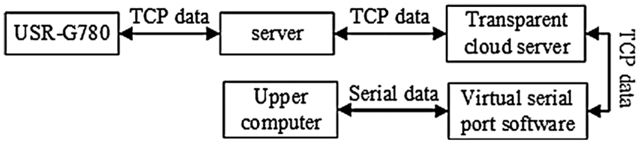


Figure 14: Overall architecture of communication system

The serial communication interface of the upper computer is written by VISA node function which is built in LabVIEW. The working principle of the VISA serial communication interface is as follows. After configuring the parameters of the serial port by “VISA Configure serial port”, use the “VISA Read” and “VISA Write” in the loop to read or send the serial port data respectively. Finally use “VISA close” and “Clear errors” to end the communication of the serial port. The program of VISA serial communication interface is shown in Fig. 15.

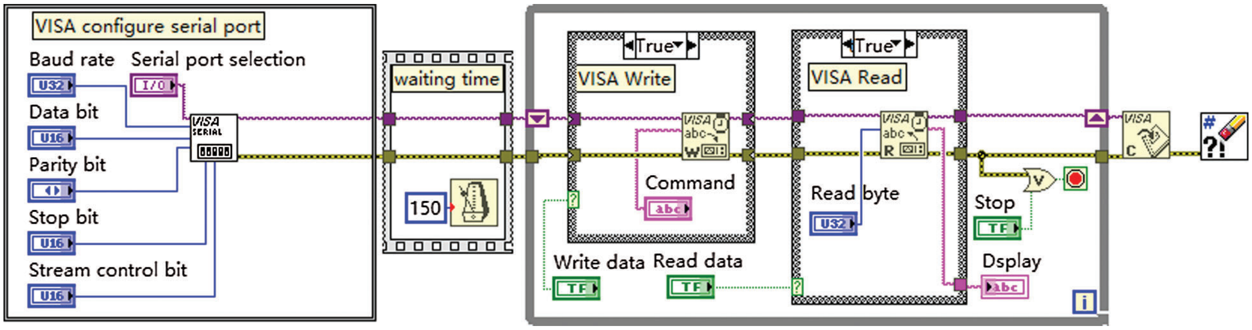


Figure 15: The program of VISA serial communication interface

5.4 Design of Data Monitoring System

The data monitoring system has functions of data display, data storage, data processing, data query, USV control and autonomous cruise. The main operation interface of data monitoring system is shown in Appendix A.

When the system is running, the operator can set the serial port parameters in the main interface and click the “Read Data” button. If the serial port has a signal, the system can read serial port data. Firstly, the system extracts the serial port data and displays the result on the main interface. Secondly, the system processes the data that has been extracted. Thirdly, both of being extracted data and processed data are automatically stored in the corresponding document. Fourthly, the system will judge data at the same time. And if it is an abnormal value, it will be rejected and will not be further processed. Fifthly, if the operator clicks on the “temperature alarm” button and sets the alarm value of temperature, an alarm will be issued after the temperature exceeds the preset value. Sixthly, if the operator clicks on the “autonomous cruise” button, the system will send a command to the USV, and if the operator intervenes in the speed and heading of the USV during this process, the autonomous cruise will be terminated. The specific flow chart of data monitoring system is shown in Fig. 16.

This paper takes the air pressure value of BMP180 as an example to explain the process of extracting the data of each meteorological sensor in upper computer. Firstly, the “String subset” function is used to intercept 5 characters as the air pressure value of BMP180. Secondly, the “Fract/Exp string to number” function is used to convert a string to a numeric value. Finally, the “Quotient & Remainder” function is used to convert the numeric value to the correct air pressure value. Since it is necessary to save the above air pressure values, the “Number to fractional string” function is used to convert the correct air pressure value to the string.

Once the data has been extracted, the system uses the waveform chart control to display the air pressure real-time curve. The extraction process for data such as temperature, humidity, wind speed, wind direction, latitude, longitude and heading are similar to the extraction air pressure value. The program of data extraction is shown in Fig. 17.

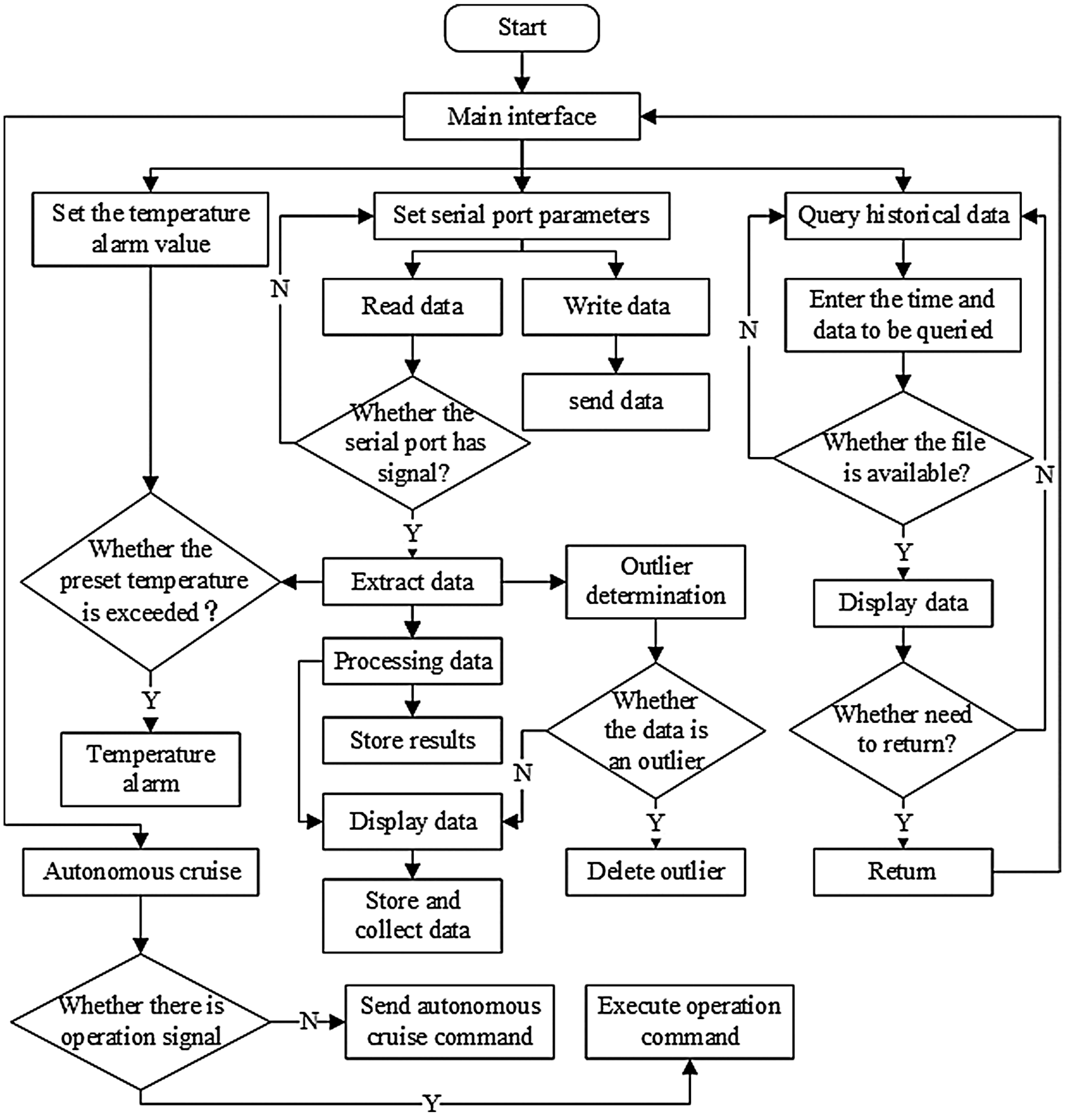

Start
Main interface
Set the temperature alarm value
Set serial port parameters
Query historical data
Read data
Write data
Enter the time and data to be queried
send data
N
Whether the serial port has signal?
N
Whether the file is available?
N
Y
Y
Whether the preset temperature is exceeded ?
Extract data
Outlier determination
Display data
Processing data
Y
Whether the data is an outlier
Whether need to return?
Store results
N
Temperature alarm
Display data
Y
Y
Autonomous cruise
Delete outlier
Return
Store and collect data
Whether there is operation signal
N
Send autonomous cruise command
Execute operation command
Y

Figure 16: Flow chart of data monitoring system

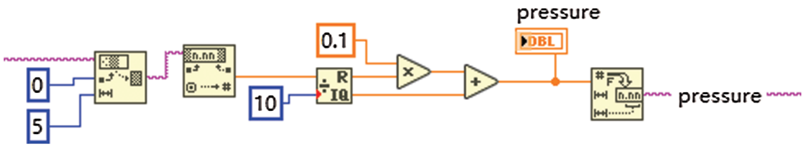
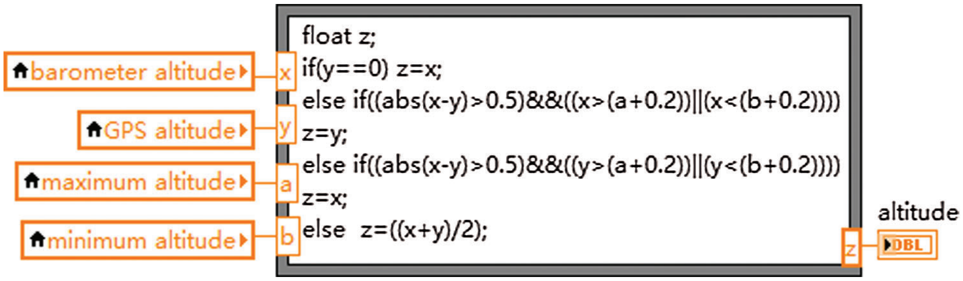


Figure 17: The program of data extraction

Some of the elements collected by this system have multi-group data. So, they can be fault-tolerant. Taking the altitude value as an example, the system performs fault-tolerant processing on the altitude values that is acquired by air pressure sensor and GPS sensor, which improves the stability of the data. The specific judgment rules are as follows:

1. The difference between the distances of the two sensors is greater than 0.5 m;
2. The output of one of two sensors is 0.2 m higher than the maximum value or 0.2 m lower than the minimum value. The program of altitude fault-tolerant is shown in Fig. 18.



```
float z;
if(y==0) z=x;
else if((abs(x-y)>0.5)&&((x>(a+0.2))||(x<(b+0.2))))
z=y;
else if((abs(x-y)>0.5)&&((y>(a+0.2))||(y<(b+0.2))))
z=x;
else  z=((x+y)/2);
```

Figure 18: The program of altitude fault-tolerant

The system uses "statistical" and "histogram" functions in LabVIEW to process extracted data. Taking the temperature value as an example, firstly, the "Feedback Node" and "Create Array" functions are used to generate a one-dimensional array. Then the system converts it into the dynamic data. Finally, the "statistics" function is used to count multiple statistics such as the total number of samples, average value, maximum value, and minimum value. And the "histogram" function is used to display temperature values as a histogram. The program of data processing for temperature is shown in Fig. 19.

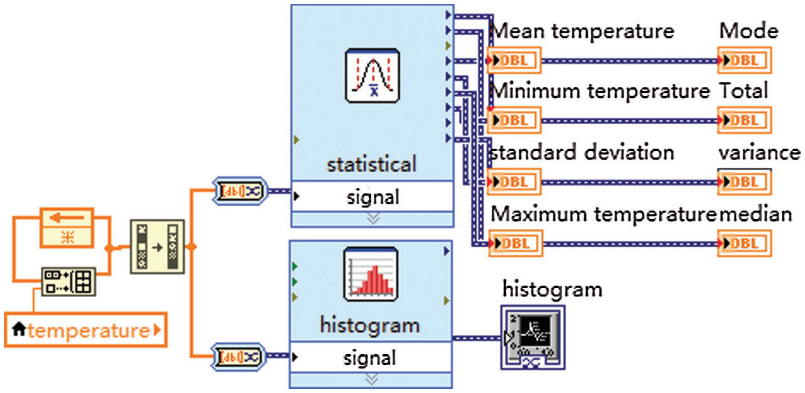


Figure 19: The program of data processing for temperature

In this system, the way to store data is described below. Firstly, the "Strip path" and "Build path" functions are used to save the file in the "historical data" folder of the path where the main function is located. Then the "Format Date/time string" function is used to extract the date. Finally, a one-dimensional array created from each meteorological data and date is converted into a two-dimensional array by a "Feedback node" and stored in a spreadsheet. The program of data storage is shown in Fig. 20.

The operator can open the spreadsheet that needs to be queried by typing the file name. The process of querying historical data is similar to the process of storing data described above, except that the "write spreadsheet” function is changed to “read spreadsheet file” function. The program of query historical data is shown in Fig. 21.

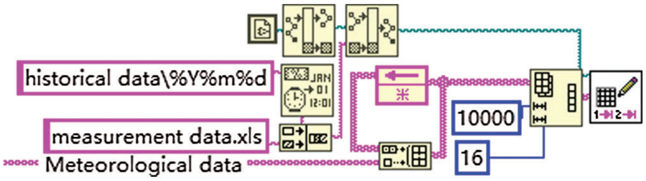


Figure 20: The program of data storage

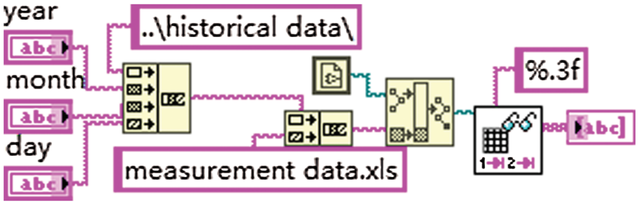


Figure 21: The program of query historical data

6 Conclusions

The design proposed in this paper basically realizes the goal of designing and developing the meteorological monitoring USV. Firstly, the USV can use the camera, ultrasonic wave sensor and RRT-Path algorithm to realize the path planning function. Secondly, the USV uses coordinates which are acquired by the electronic compass to calculate the angle between the heading of USV and the target point. Then the incremental PID algorithm is used to adjust speed of motors to control the heading of USV. Thirdly, the USV can acquire meteorological data such as temperature, humidity, air pressure, the wind speed and wind direction. These data are transmitted through the communication system and they can be processed in the upper computer to realize the online detection function. The upper computer is built by LabVIEW, which has the ability to extract, display, the process and store meteorological data.

The program can be further improved on the following two points. Firstly, Multiple sensors are used to detect a certain meteorological value at the same time, which can improve the accuracy of monitoring data. Besides, this measure can also increase the stability of online monitoring. Secondly, use the hardware visualization extension provided by the ARM MCU to construct a secure display path between the applications and the display device [15], which can increase the security of communication system when transferring data.

Acknowledgement: We thank other members of the laboratory for helpful and insightful remarks.

Funding Statement: This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ICAN(ICT Challenge and Advanced Network of HRD) program(IITP-2020-0-01832) supervised by the IITP(Institute of Information & Communications Technology Planning & Evaluation) and the Soonchunhyang University Research Fund.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] S. K. Zhang, Z. J. Liu, X. K. Zhang and Y. Liu, "Development and prospect of unmanned boats," *World Shipping*, vol. 38, no. 9, pp. 29–36, 2015.

[2] J. L. Li, "Development and application of surface unmanned boats," *Fire Control & Command Control*, vol. 37, no. 6, pp. 203–207, 2012.

[3] S. M. H. Rostami, A. K. Sangaiah, J. Wang and X. Z. Liu, "Obstacle avoidance of mobile robots using modified artificial potential field algorithm," *EURASIP Journal on Wireless Communications and Networking*, vol. 70, pp. 1–19, 2019.

[4] L. G. Liu, Y. X. Peng and W. Q. Xu, "To converge more quickly and effectively—Mean field annealing based optimal path selection in WMN," *Information Sciences*, vol. 294, pp. 216–226, 2015.

[5] G. L. Sun, Z. G. Tao, B. Zhang, D. G. Hou and H. P. Li, "Application of Beidou satellite communication technology in the landslide remote monitoring and early warning system," *Advanced Materials Research*, vol. 945–949, pp. 2150–2154, 2014.

[6] C. Fang, "Research on data transmission system of unmanned boat based on Beidou," Ph.D. dissertation. Jimei University, China, 2017.

[7] B. Yin and X. T. Wei, "Communication-efficient data aggregation tree construction for complex queries in IoT applications," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3352–3363, 2018.

[8] C. S. Tan, "Design of wireless remote monitoring system based on digital radio station," *Industry and Mine Automation*, vol. 41, no. 4, pp. 9–12, 2015.

[9] Z. Y. Lai, C. P. Li and J. Zhang, "Design of GSM/GPRS based on STM32 in hydrological monitoring," *Practical Electronics*, vol. 22, no. 3, pp. 23–24, 2015.

[10] Y. H. Deng, "Overview of 4G communication technology," *Digital Communication World*, vol. 1, no. 2, pp. 58–63, 2005.

[11] X. Y. Yan, J. H. Yang and W. Lu, "Design of remote GPRS-based gas data monitoring system," *IOP Conf. Series: Earth and Environmental Science*, vol. 108, no. 4, pp. 1–4, 2018.

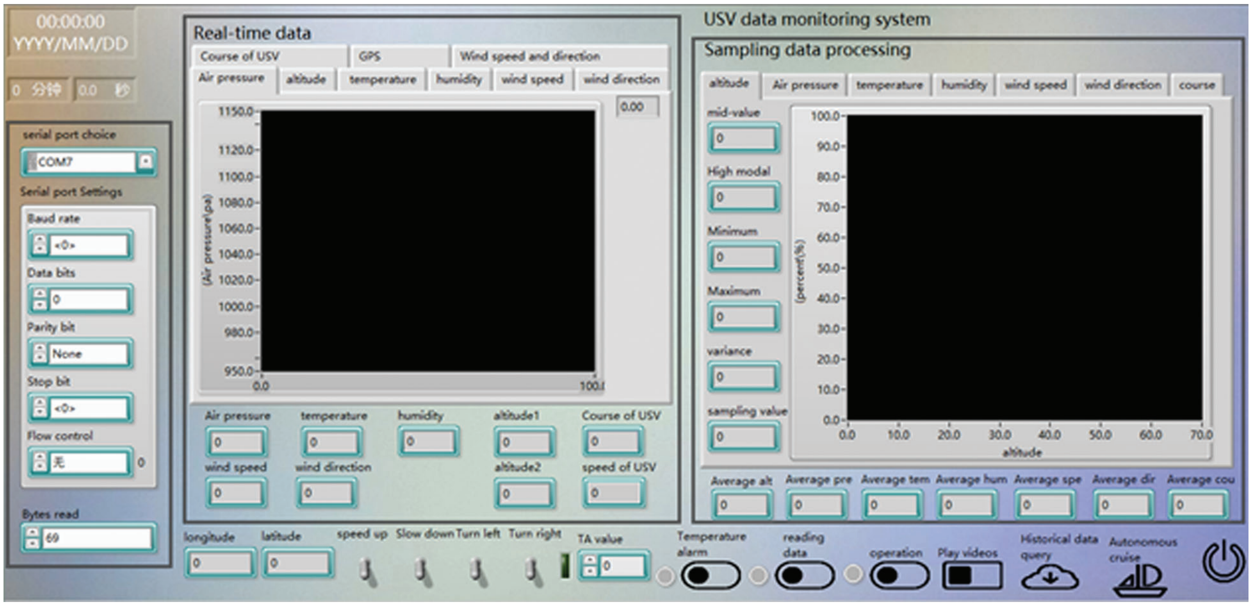
[12] C. Yang, I. K. Mei, J. Y. Chen and J. N. Zhu, "4G communication technology and its application prospects," *Communication & Information Technology*, vol. 41, no. 3, pp. 58–60, 2011.

[13] S. Ozana and T. Docekal, "PID controller design based on global optimization technique with additional constraints," *Journal of Electrical Engineering*, vol. 67, no. 3, pp. 160–168, 2016.

[14] T. Albena, P. Michail and G. Ivan, "Hybrid PID control algorithms for nonlinear process control," *Journal of Electrical Engineering*, vol. 63, no. 3, pp. 147–152, 2011.

[15] J. H. Cui, Y. Y. Zhang, Z. P. Cai, A. F. Liu and Y. Y. Li, "Securing display path for security-sensitive applications on mobile devices," *Computers Materials & Continua*, vol. 55, no. 1, pp. 17–35, 2018.

Appendix A. Schematic diagram of main interface


00:00:00
YYYY/MM/DD

0 分钟 0.0 秒

serial port choice: COM7

Serial port Settings

Baud rate: <0>

Data bits: 0

Parity bit: None

Stop bit: <0>

Flow control: 无

Bytes read: 69