Realization of Multidimensional Data Acquisition and Visualization System for Intelligent Pasture Monitoring in High and Cold Regions

Tao Ye[1,*], Yucheng Dong[1], Xiaoyan Ma[2]

[1] College of Computer, Qinghai Nationalities University, Xining 810007, China
[2] Anyang senior technical school, Anyang 455000, China
* Correspondence author: Tao Ye

Abstract: targeting the problems of harsh environment, backward infrastructure in pastoral areas and traditional pasture management in high and cold regions, a set of low-cost, low-power, highly available multi-dimensional data acquisition and data visualization management scheme is proposed. For the environment and livestock position information of pastures in high and cold regions, sensors, the GPS positioning technology and the ZigBee ranging technology are employed to collect the multi-dimensional data of pastures. The Arduino motherboard is used to integrate the data of terminal nodes. The ZigBee networking technology provides the underlying data transmission. Through the raspberry motherboard, the data is gathered and transmitted to the cloud server through the 4G network for data storage. Besides, Java, JavaScript and other programming languages are used in the cloud server to build a Web front-end for data visualization and analysis. Remote intelligent pasture management is achieved with the help of filed data collection.

Keywords: intelligent animal husbandry; data visualization; ZigBee technology; cloud server

1. Introduction

In recent years, as more and more attention has been paid to the development and promotion of the new animal husbandry and the animal husbandry with Internet of Things (IOT), the intelligent animal husbandry is embracing new development opportunities. In order to improve the economic benefits of the animal husbandry and promote the sustainable development concept of environmental and ecological protection and social and economic development in the Three-River (the Yangtze River, the Yellow River, and the Lancang River) source area in Qinghai Province, the provincial government has proposed to employ the big data platform management to vigorously promote the advancement of smart ecological environment and smart animal husbandry. In this context, the IOT technology is combined to improve the efficiency of ecological breeding in high and cold areas, monitor the data of farms, protect the pasture ecology, and effectively improve the economic benefits of herdsmen.

In China, all kinds of natural grassland account for about 40% of the total land area. The grassland in the Qinghai-Tibet Plateau is mainly characterized by being high and cold. The pasture has long and cold winter, small annual precipitation and wind-blown sand [1]. With harsh environment in the high and clod region, the grassland degradation becomes serious in recent years. Being remote and inconvenient in transportation, most of the pastoral areas lack the intelligent grazing technology [2]. At present, the intelligent pasture monitoring is mainly achieved with the help of automation and the IOT technology, and the control system is largely built on the terminal and cloud [2-5]. The sensing layer of the system is mainly composed of multiple sensors, the single chip microcomputer Arduino, the Raspberry and so on [6]. The ZigBee networking technology has been the most widely applied in environmental data collection, animal related data collection, medical monitoring, agricultural data collection, etc. The ZigBee technology has also been studied and applied in indoor positioning, but the positioning process is greatly affected by environmental factors [7-12]. With the popularization of intelligent breeding, combination of GPS and GPRS technologies has alleviated the problem of livestock positioning in remote areas to some extent, but problems such as high cost and low positioning accuracy still exist. A low-cost and high-precision positioning module is proposed for design [1,9,13].

2. Technology Review

2.1. Introduction of Hardware Technology

ZigBee is a new wireless communication technology with low power consumption, low cost and short distance. The standard IEEE802.15.4 is applicable to its physical layer and media access layer. With 2.4G high frequency transmission, the transmission distance can be from tens of meters to hundreds of meters [14]. The ZigBee technology boasts the advantages of low power consumption, low cost, low rate, short delay, large network capacity, high security, and flexible ad-hoc network function. Compared with WiFi and Bluetooth, the ZigBee technology has a long transmission distance. With poor penetration, it is easy to be affected by obstacles in data transmission [15].

The GPS locator has to be composed of three main parts: the space satellite, the ground monitoring and the user receiving, which can provide real-time positioning information with high-precision position, speed and time for global users [9]. The system positioning module (WH-GN100) supports the GPS multi-system or single-system independent positioning, D-GNSS differential positioning and A-GNSS auxiliary positioning, so the cost is greatly reduced, and it can be applied in remote pastures [16].

With small amount of data of acquisition nodes and control commands, and low maintenance cost in the data acquisition system, it is unnecessary for the system to use large database for data storage and management. Considering the advantages of small size, high speed and low maintenance cost of the Myaql database, the need to meet the requirements of rapid response, high reliability and multi-platform access of data transmission, and the actual functional needs, the MySQL database is selected for data storage.

The cloud server provides support for building websites and databases. One of the biggest advantages of the cloud server lies in its flexibility and convenience. Its own configuration can be expanded according to the demand, with low failure rate and strong stability. Compared with ordinary servers, it can save hardware cost and reduce maintenance cost.

2.2. Introduction of Software Technology

The system is equipped with the B/S architecture, the iframe tag in HTML to arrange the web page layout, and the div+css network layout method to make the web page more beautiful. Some JavaScript statements are used to ensure the system performance. The Charts library is utilized to visualize the graphics, making the data display more intuitive and convenient [17]. Using the Autonavi API interface program, the longitude and latitude coordinates of livestock can be directly displayed and marked on the map, so that the user can have a more intuitive experience. At the same time, the MVC framework (JSP + servlet + Javabean) also solves the problems of high concurrency and real-time data extraction in this system, and provides multi-person operation and a large amount of data to ensure that the system will not make mistakes in advance [18].

3. System Design and Implementation

3.1. Overall Structure Design of the System

The system is based on combination of the sensor network (the ZigBee network technology) and 4G network. The monitoring objects are chiefly divided into three categories: environmental monitoring, livestock monitoring and water quality monitoring. The raspberry motherboard and 4G module are used for the gateway. The Arduion module and the ZigBee terminal are designated for data aggregation of the bottom nodes of the system [19]. Each environmental area node is divided into two types. The monitoring data of node 1 include ultraviolet, atmospheric pressure, raindrops, wind, PM, soil humidity and dust; the monitoring data of node 2 include temperature and humidity, smoke and flame, and node 2 is further divided into many sub-nodes to predict regional fires. In the water quality monitoring, the node monitoring data of each water quality area include: temperature, turbidity, conductivity, PH. The livestock monitoring is mainly divided into two sets of transmission modes. Firstly, the GPS module and the ZigBee terminal are employed for the bottom node data aggregation, and the monitoring data include: the real-time longitude and latitude of cattle and sheep and the altitude [7]; secondly, it is composed of the ZigBee terminal (hereinafter referred to as the “tag”) and the ZigBee coordinator (hereinafter referred to as the “card reader”), and the main data collected is the relative distance between cattle and sheep.

Deployment mode of environmental monitoring nodes: the second type of nodes is used for data collection in the surrounding area of the pasture, and the first type of nodes is deployed in the central area. Deployment mode of water quality monitoring node: the data in the place with water source in the pasture area are randomly selected, and are compared and analyzed with the water quality data in different areas. Livestock monitoring is deployed in the livestock group, and the gateway equipment is worn on the leading animal, the terminal node belt with the GPS positioning module is worn on some representative animals in the group, and the label is worn on each animal.

The 4G module and the card reader are connected with the raspberry motherboard through the USB interface, and serial ports serve to connect the Arduino motherboard with the ZigBee terminal and the ZigBee coordinator with the raspberry motherboard, so as to realize data exchange. In the environmental monitoring system, two kinds of nodes can transfer data to the coordinator through the ZigBee router, and the same is true in the water quality monitoring system. See Fig. 1 for the overall system structure.

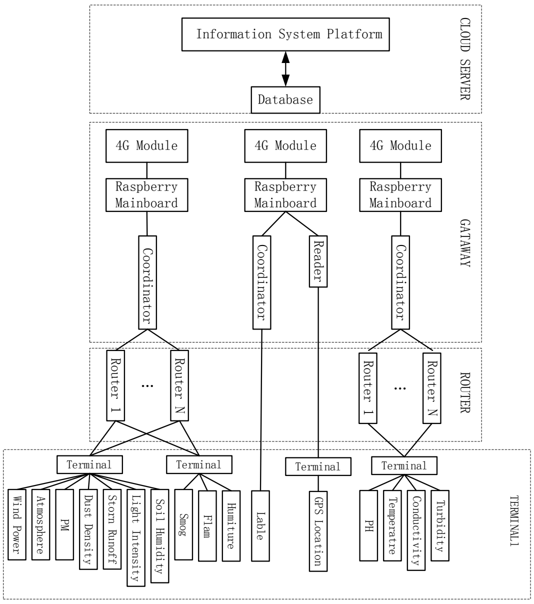

Figure 1. Overall system structure

3.2. Design and Implementation of Network Structure

Combined with the actual situation in the high and cold region, the sensor network of environmental monitoring and water quality monitoring is of the mesh network structure, and the sensor network of cattle and sheep GPS positioning is of the star network structure. Once a node fails, the data transmission of other terminal nodes will not be affected, so as to improve the reliability of data transmission. The design of sensor network include modification under the APP file of the Z-stack protocol stack to realize serial port transmission. Since the system is divided into three types of data for acquisition and transmission, the ZigBee network will inevitably appear in the same working area. In order to avoid mutual interference and reduction of communication efficiency, three different channels are selected for communication [14]. Gateway is designed for sensor network conversion, and 4G network for data transmission. The 4G network is used to transmit data to the cloud server, which provides mobile phone and computer terminal users to visit the website.

3.3. Design and Implementation of Hardware Architecture

The system hardware is mainly composed of the raspberry motherboard, the 4G module, the card reader, the tag, the ZigBee core module, the Arduion motherboard, the GPS positioning module and the sensor. See Fig. 2 for the hardware structure.

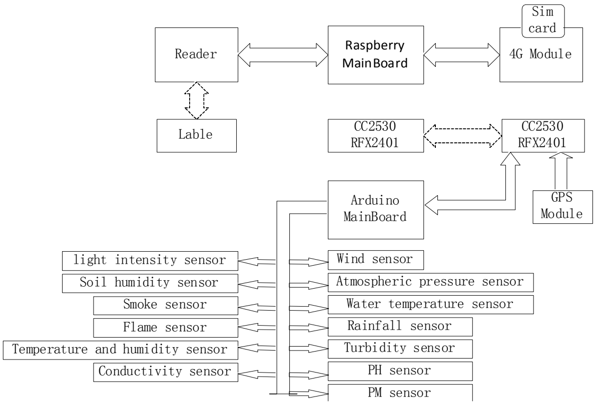

Figure 2. Hardware structure

The direction of system monitoring can be divided into three categories: sensor data, positioning data and ranging data. (1) The sensor data are mainly collected by various sensors, aggregated by the Arduino motherboard, and sent and received by the ZigBee radio frequency (RF) chip; (2) the positioning data are mainly sent and received by the GPS positioning module and the ZigBee RF chip; (3) the ranging data are mainly composed of the tag and the card reader module; each unit module gathers data through the raspberry motherboard and forwards it to the cloud server. See Fig. 3 for the hardware.

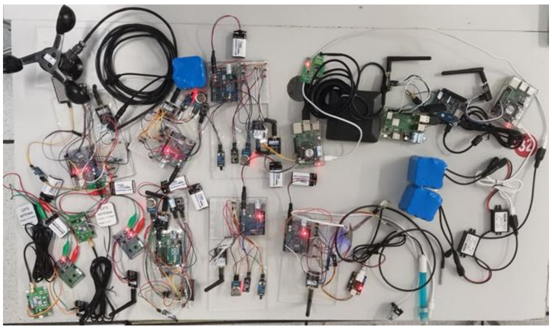

Figure 3. Hardware

The power supply devices vary for different monitoring modules of the system, mainly divided into Module 1 (the Arduino motherboard, the sensor and the CC2530+RF2401 node), Module 2 (the positioning module, and the CC2530+RF2401 node), Module 3 (the tag), and the gateway module (the raspberry, the 4G module, and the CC2530+RF2401 node). Considering the applicable scenarios of each module, the power supply modes are also different. Module 1 is supplied by the solar cell, with the output voltage of 12V. The wind sensor is directly connected to 12V, and the AMS1117 step-down module provides a stable 5.0V voltage for the Arduino motherboard [12]. Module 2 is supplied by the 9V 700mAh low self-charging/discharging battery, and the AMS1117 step-down module is used to provide stable 5.0V voltage for the positioning module and to provide stable 3.30 V voltage for the CC2530+RF2401 node. Module 3 is supplied by the 3.0 V button cell. The gateway module is supplied by the 12V rechargeable lithium battery, and the intelligent QC3.0 power converter is used to provide 5V 3A power supply for the raspberry motherboard. The reason why we choose to use the solar cell, the rechargeable battery and the button cell is that in addition to their long service life, they share the biggest advantage of saving energy and protecting environment, and they can be applied to different actual environments.

3.4. Design and Implementation of Communication Module

The Arduino module is connected with each sensor, and with the CC2530+RF2401 node through the serial port. The GPS positioning module is connected with the CC2530+RF2401 node. The 4G module and the raspberry motherboard can be connected by driver installation and startup script setting to the network automatically after start-up as long as the SIM card with GPRS is inserted into the card slot. The raspberry motherboard modifies the startup script to run the data collection code, and the bootable system can automatically collect and transmit data. To sum up, after the setting of the communication module of the whole system, the system will start automatically without manual intervention as long as the power supply is provided.

(1) GPS positioning module

The positioning system module is composed of the positioning data acquisition circuit, the positioning data sending circuit and the step-down circuit, which completes the function of positioning data terminal sending. See Fig. 4 for the hardware connection of the positioning module.

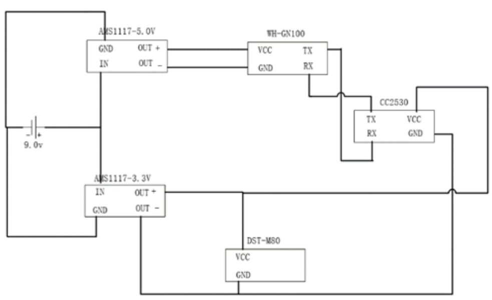

Figure 4. Hardware connection of the positioning module

Since the data measured by the positioning module is too large at one time, serious packet blocking and packet loss occur in the process of ZigBee data transmission. Therefore, the ZigBee terminal is used to screen out the required part of data transmission. The node number is added to the terminal (take node L0003 as an example, where L represents the positioning data, 0003 represents the node number carried by the data, and the content can be modified according to the actual situation) to determine the terminal node identification, so that the gateway can cut the data according to the node number and transmit the data. Modification of the terminal Z-stack protocol stack is shown in Fig. 5 (partially).

```
void SampleApp_SerialCMD(mtOSALSerialData_t *cmdMsg)
{
  uint8 i, j, k, len, len_cp, *str=NULL, *str_a="NULL", *str_cp="NULL";

  str=cmdMsg->msg;
  str_a=str;
  len=*str;
  for(i=1;i<=len;i++)
  {
    HalUARTWrite(0,str+i,1);
  }
   for(j=1;j<=len;j++)
  {
    if( *(str_a+j)=='G' && *(str_a+j+1)=='N' && *(str_a+j+2)=='G' && *(str_a+j+3)=='G' && *(str_a+j+4)=='A')
    {
      str_cp = str;
      len_cp=*str_cp;
       for(k=1;k<=len_cp;k++)
        {
          HalUARTWrite(0,str_cp+k,1);
        }
    }
  }
 *(str_cp+1)='L';
 *(str_cp+2)='0';
 *(str_cp+3)='0';
 *(str_cp+4)='0';
 *(str_cp+5)='3';
  if(AF_DataRequest(&SampleApp_Periodic_DstAddr, &SampleApp_epDesc,
                    SAMPLEAPP_COM_CLUSTERID,
                    len_cp+1,
                    str_cp,
                    &SampleApp_TransID,
                    AF_DISCV_ROUTE,
                    AF_DEFAULT_RADIUS)== afStatus_SUCCESS)
  {
  }
  else
  {
  }
```

Figure 5. Partial program diagram of terminal Z-stack protocol stack

(2) Gateway module

The gateway data processing is mainly completed by the raspberry module with the help of the Python grammar. The data is received by the serial port, and uploaded by connection to the server. The Serial library is used to read the serial data, transform the data coding format, and filter and cut the data. Environment data processing: a list is defined; the common elements in the list are determined by the data type of the terminal node (number of list elements = data type of the terminal node + 1); the initial value is defined for each element, the first element to determine the name of the node, the other elements to represent the data monitored by the current node and each element to represent a kind of sensor data, and each terminal node is named with 4 bits (for example, W1TC21.75: W1 represents the node name, TC represents the temperature, and the remaining bits represent the sensor data). After determining the node number and data type, the data bits of the remaining bits are put in the corresponding positions of the list elements. The list judges every data entry to determine whether all the elements in the list are the initial values defined initially. When all the elements are not the initial values, the whole data is sent to the database. The above operation is wrapped by “try”, and a “pass” is put in the “except” to solve the problem of abnormal code stop caused by data receiving error in the test, and “try” is put into the dead loop to ensure high availability of the module. The screenshot of data processing code of the livestock positioning information gateway is shown in Fig. 6.

```
data = serial.readline()
if (('L' in data.decode('UTF-8')) == True and ('A' in data.decode('UTF-8')) == True and (
        'N' in data.decode('UTF-8')) == True and ('E' in data.decode('UTF-8')) == True):
    a = data.decode('UTF-8')
    if (data[a.index('A') - 5] == 76):
        # print(a.index('L'))
        number1 = chr(data[a.index('L') + 1])
        number2 = chr(data[a.index('L') + 2])
        number3 = chr(data[a.index('L') + 3])
        number4 = chr(data[a.index('L') + 4])
        number = number1 + number2 + number3 + number4
        # print(a.index('A')-5)
        aa = data[a.index('A') - 5:].decode('UTF-8')  # find data
        print(aa)  # find data
        print(number)  # ID

        # find E
        print(aa.index('E'))
        print(aa[aa.index('E') - 1])
        print(ord(aa[aa.index('E') - 1]))
        i = j = m = k = 0
        # find E
        while True:

            if (ord(aa[aa.index('E') - 2 - i]) == 44):
                print(aa.index('E') - 2 - i)
                break
            i += 1
        E = aa[aa.index('E') - 1 - i: aa.index('E') - 1]

        print(E)
        E1 = int(float(E)//100)
        E2 = '%.6f'%((float(E)%100)/100)
        E2 = float(E2)/60*100
        E2 = '%.6f'%(float(E2))
        E = float(E1)+float(E2)
        print(E)
        # print(aa[aa.index('E')-3-i])
```

```
            # find N
            while True:

                if (ord(aa[aa.index('E') - 5 - i - j]) == 44):
                    print(aa.index('E') - 5 - i - j)
                    break
                j += 1
            N = aa[aa.index('E') - 4 - i - j: aa.index('E') - 4 - i]

            print(N)
            N1 = int(float(N)//100)
            N2 = '%.6f'%((float(N)%100)/100)
            N2 = float(N2)/60*100
            N2 = '%.6f'%(float(N2))
            N = float(N1)+float(N2)
            print(N)

            # find elevation
            while True:

                if (ord(aa[aa.index('E') + m]) == 77):
                    break
                m += 1
            # print(aa[aa.index('E')+m])  # find M
            while True:

                if (ord(aa[aa.index('E') + m - k - 2]) == 44):
                    break
                k += 1
            elevation = aa[aa.index('E') + m - k - 1: aa.index('E') + m - 1]
            print(elevation)
        cursor.execute(sql, [number, E, N, elevation])
        conn.commit()
    print()
except:
    pass
```

Figure 6. Screenshot of data processing code of livestock positioning information gateway

4. Design and Implementation of Software

4.1. System Data Flow Diagram

The administrator needs to call the user name and password information of the administrator when logging in. The data visualization processing needs to call all kinds of data in the database to draw or display in the form of chart. The administrator can only browse and query the information in the browsing interface after logging in. All the data information files are called unilaterally, and no data addition or modification is possible, so as to ensure the practicability and security of the data. The data flow is shown in Fig. 7.

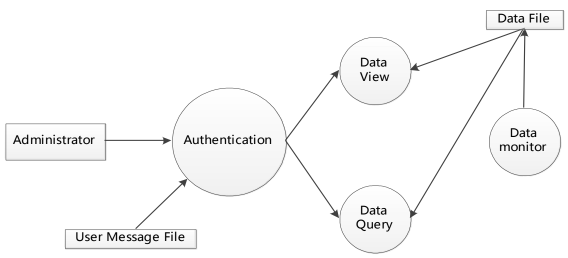

Figure 7. Data flow diagram

4.2. Making of Data Line Chart

The echart library and Ajax (asynchronous transmission) are used to make the line chart. In order to accurately reflect the change of data, the line chart is adopted to display data, and the data display arrays are imported into the horizontal and vertical axis arrays respectively. Important codes for making line chart is shown in Fig. 8 below:

```
xAxis: [{

        type: 'category',
        position: 'bottom',
        data: arr77,

        },
yAxis: {
      type: 'value'
  },
  series: [
      {
        name: 'temp'
        type: 'line'
        data: arr2,
      },
```

Figure 8. Important codes of line chart

4.3. Interface Integration and Editing

The graph is integrated into a div module by the iframe tag to display the graph and detailed table data for the web page. The line chart and data table are imported into the web page through the iframe tag, a table is defined in the web page, the line chart is imported after simple merging of rows and columns, and the corresponding relationship between the button and line chart is determined through "name" and "target" definitions of the iframe. The integrated display interface is shown in Fig. 9.

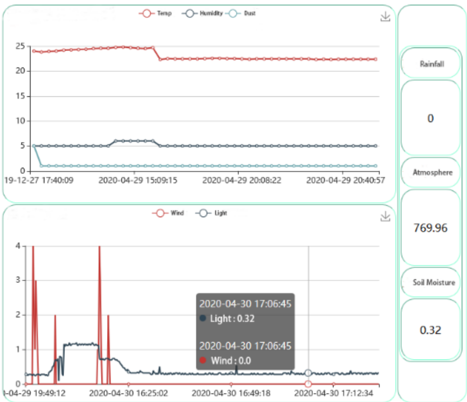

Figure 9. Integrated display interface

4.4. Graphic Display

The jQuery module of JavaScript is used in web page making for simple data exception warning. When the data item is abnormal, the bottom color of the table changes, and a Popup jumps out directly when the fire occurs to make timely reminder.

The abnormal display data items are as follows:

(1) Environmental data monitoring interface (as shown in Fig. 10)

Temperature: the temperature in the high and cold region is generally lower than that in other regions at the same time, so attention shall be paid to the fire risk of the region when the temperature in a certain region is higher than 40 degrees.

Smoke: after testing, the value of smoke sensor can be known to be between 100 and 500. After grading, the smoke is divided into 1 to 8 levels. When the smoke concentration reaches level 2, alarm has to be sent that the smoke concentration becomes higher to alert the herdsmen.

Flame: the sensor value of flame is divided into 0 and 1. When the value is 0, there is no fire risk. When the value is 1, the alarm box will pop up directly to inform the fire, which is convenient for herdsmen to make timely response to the fire.

Wind force: the wind force value is between 0 and 20. When the wind force value reaches 6, the bottom color of the table will turn blue. At the same time, the alarm box will be sent out to alert managers to pay attention to the impact of strong wind on the rise of grassland grass.

PH, turbidity, conductivity: the drinking water of living standard in our country is taken as the standard. When the value is lower or higher than the standard value, the warning prompt will be made through the color change of the table background.

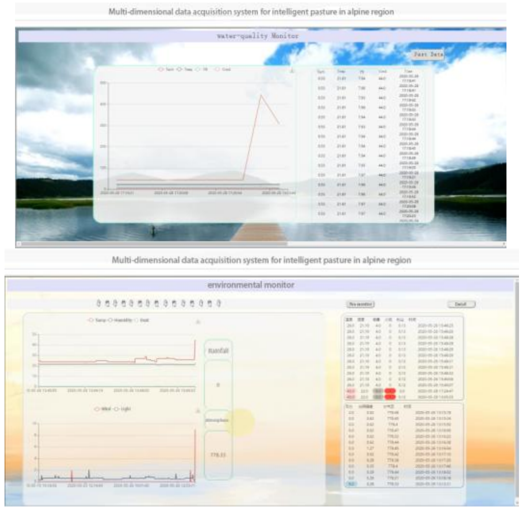

Figure 10. Environment data interface

(2) Monitoring interface of livestock positioning data

After the field test, a reasonable data upload interval is set. The interval for the GPS positioning data to be uploaded to the server is 1min. The tag sends the data to the card reader through the ZigBee network every 10s. The card reader transmits the data to the gateway and finally to the server. When the animal with the GPS positioning module leaves the delineated grassland area, the interface alarm will be processed. The livestock positioning is shown in Fig. 11. The daily activity track of each animal with the GPS positioning module is displayed, as shown in Fig. 12. When the animals with labels leave the transmission distance for a long time, the warning processing is made for the lost labels, and the number of warning times is counted. When a label is in the warning object for a long time, the possibility of loss is analyzed, and the more the times of continuous warning, the greater the possibility. Relative distance of livestock is shown in Fig. 13.

Figure 11. Livestock positioning

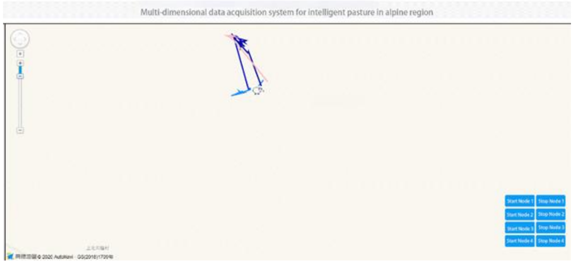

Figure 12. Livestock activity track

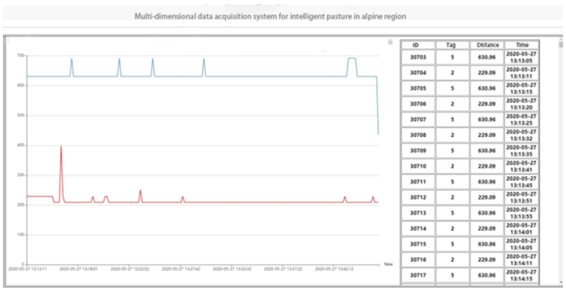

Figure 13. Relative distance of livestock

5. Conclusion

According to the unique characteristics of pastures in the high and cold region, the system has been successfully applied in actual pastures to monitor and alarm the fire information in the pastures, to locate the livestock travelling outside, and to perform automatic real-time monitoring of the environment surrounding the pastures. It is characterized by low cost, low power consumption and small size, which can be applied to remote areas with relatively backward network. The system can be applied to the actual environment, so as to provide data support for herdsmen's intelligent management of pastures, which contributes to the development of pastures in high and cold regions, promotes the process of intelligent animal husbandry, and is of long-term and important significance for economic development in such regions.

Acknowledgement

This work is supported by Qinghai Science and Technology Project (2018-ZJ-753).

Reference

[1] Hu, D. Design of intelligent ranch management system based on Nb IOT. *Information technology and informatization*, **2018** (07): 31-33.

[2] Xu, Y. Design of intelligent ranch monitoring center. Baotou, *Inner Mongolia University of science and technology*, **2016**: 1.

[3] Chen, H.; Qin, X. Design of intelligent environmental control system based on cloud ranch. *Industrial control computer*, **2016**, 29(01): 21-23.

[4] Ouyang, C.X. Intelligent ranch based on Internet of things. *Computer products and circulation*, **2019** (04): 69.

[5] Mo, S.L.; Wang, K.; Zhang, H.C.; Zhou, B.Z.; Lin, X.; Ren, Y.L. Design of environmental monitoring training platform based on Internet of things. *Laboratory research and exploration*, **2016**, 35(08): 235-237.

[6] Ran, Y.Z.; Tang, M.X.; Yang, K.Y.; Wang, Y.W.; Cao, J.H. Design and implementation of Agricultural Internet of things system based on Arduino and raspberry. *Journal of Chongqing University of Technology (NATURAL SCIENCE)*, **2019**, 33(08): 138-143.

[7] Li, H.Q.; Mao, Z.F.; Shi, N.C. Design of cowshed environmental monitoring system based on ZigBee technology. *Computer and Telecom*, **2020** (z1): 31-33.

[8] Jia, H.; Chi X.Y. Research on smart home telemedicine monitoring technology based on ZigBee technology. *Automation application*, **2019** (11): 138-141.

[9] Chen X.M.; Liu, S.H. Design of agricultural environmental data collection system based on Internet of things. *Laboratory research and exploration*, **2018**, 37(07): 66-68+105.

[10] Du, Y.X.; Zhan, B.; Li, B.S.; Qin, L. Design of grassland cattle monitoring system based on ZigBee and GPS. *Modern electronic technology*, **2016**, 39(08): 138-141+146.

[11] Li, H.Q.; Mao, Z.F.; Shi N.C. Design of cowshed environmental monitoring system based on ZigBee technology. *Computer and Telecom*, **2020** (z1): 31-33.

[12] Xiao, T.; Cui, Z.W.; Zhao, Y.; Sang, H.W. Design of tea ecological environment monitoring system based on Internet of things. *Internet of things technology*, **2020**, 10(03): 31-32.

[13] Hou, R.; Huang, Y.X.; Zhang, K.; Lu, Y.; Gong, L. Design of field environmental monitoring system based on ZigBee and energy sensing. *Laboratory research and exploration*, **2017**, 36(05): 57-60.

[14] Wei Y.F. Design of wireless network terminal based on zstack protocol stack. *Electronic world*, **2018** (16): 143-144.

[15] Tang W.L.; Zeng, X.Y.; Cao, Y.Q. Forest fire monitoring system based on ZigBee wireless sensor network. *Laboratory research and exploration*, **2010**, 29(06): 49-53.

[16] Duan, K. Research on BDS / GPS / GLONASS multi system fusion positioning method. Shanghai Jiaotong University, **2018.**

[17] Chen, J.S.; Peng, L.F. Design and implementation of big data visualization system based on Python + ecarts. *Journal of Anhui Vocational College of electronic information technology*. **2019**. 1671-802x (2019) 04-0006-04.

[18] Bu, Q.Y. Design and implementation of web application system based on spring MVC framework. *Digital world*, **2017** (4): 21-22.

[19] Shi, L.J. Design of greenhouse environment monitoring system based on Arduino technology. *Electrical engineering*, **2019** (09): 72-73.