The Designing and Porting of Temperature &Humidity Sensor Node Driver Based on ARM-Linux

He Jianfeng[1,2], Qu Jinhui[1]
1, Engineering Research Center of Nuclear Technology Application (East China Institute of Technology), Ministry of Education
Nanchang, 330013, China
e-mail: hjf_10@yeah.net

Wang Yuan[2], Pan Hengya[2]
2, College of Software engineering East China University of Technology
Nanchang, 330013, China
e-mail: hjf_10@yeah.net

***Abstract*—The places such as greenhouses, grain depot and nuclear radiation measurement require real-time monitoring of environmental temperature and humidity, so a low-cost, low-power wireless temperature and humidity sensor network nodes based on ARM-Linux platform is designed. The paper is analyzing the operating mechanism and timing sequence of DHT11 temperature and humidity sensors in detailed, and studying data format and the processing method of sensor.The DHT11 temperature and humidity sensor node corresponding Linux drivers are programmed, then the temperature and humidity acquisition program porting to the ARM9-Linux2.6.18 platform. Meanwhile, the use of open-source cross-platform QT graphics library, the collected data through the graphical user interface is intuitive feedback to the user. The system has good stability and scalability, and has good application prospects in radiation measurement.**

Keywords- ARM-Linux; Temperature&Humidity Sensor Node; DHT11 Driver; Porting;

I. INTRODUCTION

With the development of microprocessor technology, embedded operation system and sensor detecting technology, embedded systems are increasingly applied into environmental monitoring system of various indoor places. At same time, the environmental monitoring system in terms of accuracy, efficiency and intelligence continues to improve, the functions continue to strengthen, which makes up for the past artificially consuming a lot of resources and sensor performance deficiencies to some extent. As temperature and humidity in the environment for the crops growth, grain storage, nuclear radiation measurement and other occasions are two important factors, Therefore, detection of temperature and humidity is particularly important. However, on the temperature and humidity acquisition technologies, especially digital sensor acquisition involves many technical fields, even cover all the aspects of hardware, software, logic and protocol *etc*. This design is based on the S3C2440[1] microprocessor of ARM920T series, and use DHT11[2] temperature and humidity sensor as data acquisition module, to build hardware development platform of the system. With analyzing the boot process of temperature and humidity sensor, studying the temperature and humidity sensor device drive designing, compiling and transplanting in embedded Linux system, use of Qt/E application achieving the acquisition and transplanting of temperature and humidity. Finally, miniaturization, digitization and networking of temperature and humidity with real-time online monitoring technology are completed, then get rid of insurmountable problems which analog signal and traditional digital signal monitoring brings some precision, efficiency, networking *etc*.

II. THE HARDWARE DESIGN OF SENSOR NODE

The DHT11 is a temperature and humidity sensor which exports digital signal, it combines digital module acquisition technology with temperature-humidity sensor technology to collect temperature and humidity. This sensor is consisted of a resistance element which can respond to humidity, and a element which can measure temperature. It adopts the package of 4 pins in single row, also adopts serial interface communication. The first pin is a power pin, the second pin connects the micro-controller, and it should be used 5kΩ pull-up resistor when the length of connecting line is less than 20 meters, it should be used appropriate pull-up resistor depending on actual situation when the length is more than 20 meters. The third pin should be vacant. The fourth pin should be earthed. The mode of data communication is a single bus which is serial and bidirectional, 40bits data can be transferred in one time, the data includes both integer and fractional parts of humidity and temperature, the high bits of data should be output at first, and the humidity should be gathered before temperature. The format of data acquisition as shown in Table 1, the specific connection of pins as shown in Figure 1.

TABLE I. DHT11 DATA ACQUISITION FORMAT

Humidity parts(16 bits)		*Temperature parts(16 bits)*		*Checks(8 bits)*
Byte4	Byte3	Byte2	Byte1	Byte0
Humidity Integer	Humidity Decimal	Temperature Integer	Temperature Integer	Checksum

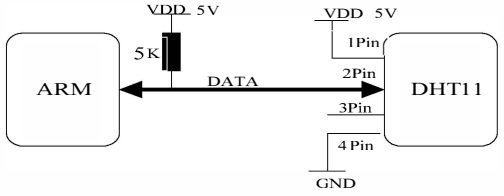

Figure 1. DHT11 Connection with the ARM Board

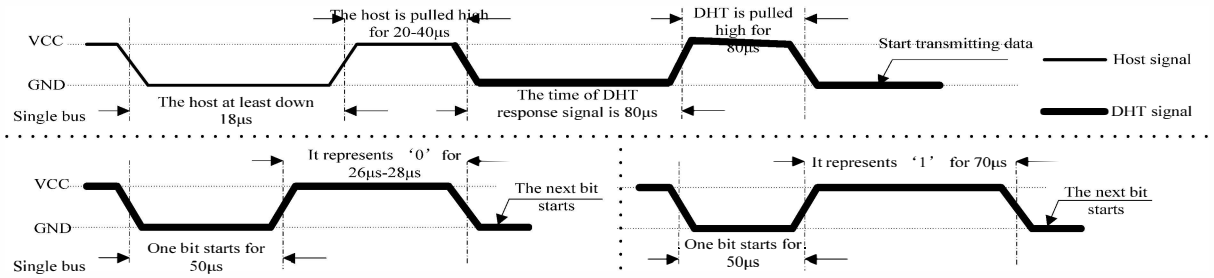

Figure 2. DHT communication processing and digital representation of 0,1

The bus operating mode as follows: Bus idle state is high level, then the host pulls down the level of bus to wait for DHT11 response, the low level should more than 18ms to ensure DHT11 can check start signal. When DHT11 receives the start signal of host, and waits for the end of host start signal, then sensor begins to send low level response signal, and the length of it's time is 80μs. When the host ends the start signal which it sends, and reads the response signal of DHT11 after delaying 20μs to 40μs, the host can switch pins to input mode, or output high level, and the bus level is pulled up by pull-up resistor. The DHT communication processing and digital representation are shown in Figure 2.

III. THE BASIC FRAME OF SOFTWARE DESIGN

The software design is consisted of three modules, they are user interaction module of GUI, sensor reading module, data analyzing and handing module *etc*. These modules interact and cooperate with each other to finish all works of the whole message monitoring analysis software, as is shown in figure 3.

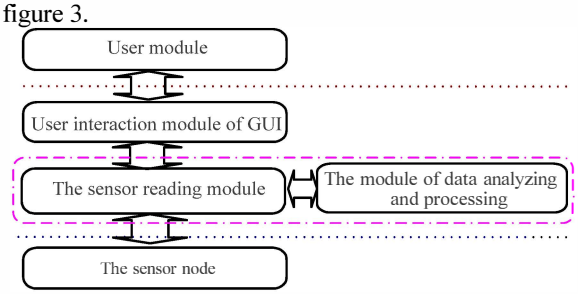

Figure 3. Architecture of temperature and humidity acquisition system

(1)User interaction module of GUI: It is responsible for interacting of users and draws graphical interfaces, disposes of users accidents, receives data that temperature and humidity sensor sends, and presents it by a visual form. When the sensor can't work normally, this module should inform users timely to deal with related anomaly. (2)Sensor reading module: this module is a key module of the whole system. It receives sensor data sent from the temperature and humidity, and the data is sent to the user interaction module is displayed. In addition, this module should monitor the working status of the sensor, when it finds abnormality, it should pass on messages to user interaction module and inform the user. (3)The module of data producing and analyzing: this module is the most important module of the whole system. It is responsible for communication with sensor according to time sequence, the level of the original signal is parsed into a data bit signals that ultimately becomes the other modules inside the machine recognizable data. Among these three modules, the user interaction module and sensor reading module are implemented in Qt/E environment. The module of data processing and analyzing is implemented by device driver.

IV. THE DRIVERS DESIGN OF DHT11 TEMPERATURE AND HUMIDITY SENSOR

A. The structure of DHT11 function

As the DHT11 sensor function is simple, so it is only needed to achieve read() function, the others, such as write(), Iseek() function has not realistic meaning[3]. When the GUI through system calls function read(), the driver uses DHT11_read() function performs the actual read operation of sensor.The driver file_operations structure is defined as follows:

```
static struct file_operations dev_fops = {
        .owner = THIS_MODULE,
        .read = DHT11_read,
        };
```

According to the sensor working sequence requests, call s3c2440_gpio_cfgpin( ) to set GPF0 port with low level to export firstly, and keep it for 18ms. Next, keep high level for 40μs, starting to read data when sensor responses to ARM controller. Then, set GPF0 port as input state, being prepare for receiving data. These code are shown as follows:

```
static ssize_t DHT11_read ( struct file* filp, char __user* buf, size_t count, loff_t* f_pos ) {
    … … …
     s3c2440_gpio_cfgpin(S3C2440_GPF(0),S3C2440_GPIO_OUTPUT);
     s3c2440_gpio_setpin ( S3C2440_GPF(0) , 0 );
     msleep (18);
     s3c2440_gpio_setpin ( S3C2440_GPF(0) , 1 );
     udelay ( 40 );
     s3c2440_gpio_cfgpin(S3C2440_GPF(0),S3C2440_GPIO_INPUT);
     …. …. ….
```

After the above-mentioned task is accomplished, the sensor will send a 80μs response signal which is low level, then pull up level for 80μs to be ready for transmitting temperature and humidity data to ARM. The loop should be passed through in driver, and delay 10μs for every time, if the sensor responses normally, then DataTemp won't be zero

after the end of loop. If DataTemp was zero, the function will end and return errors. It's same with the way that the sensor pulls up the level.

```
DataTemp = 10 ; //processing the 80μs response signal
while (!(s3c2440_gpio_getpin(S3C2440_GPF(0)) ) && DataTemp ) {
  DataTemp --;
  udelay ( 10 );  }
  if ( !DataTemp ){
  .... .... ....
  }
```

It will start transmitting data after the sensor sends all the response signal, the specific reading details is achieved by DHT11_read_byte() function. The data is read in to a character array by calling DHT11_read_byte() function ,and checking the data is lost or not during transmission by check code. When the reading ends, call copy_to_user() and send it into user space variable. Set the GPF0 port with high level output again to be ready for the next time reading when the all process is over.

B. The DHT11_read_byte function

The DHT11_read_byte() function processes bits which the sensor sends. The data format as shown in Table 1, and they are all 8 bits data. So, every time data needs to be processed with only 8bits. From the sensor timing sequence we know that before every bit begins to be transmitted there will be 50μs low level output, then the high level will appear, the length of timing sequence determines the bit it transmits is 0 or 1 for every time. Therefore, the key of this module is deciding time interval between high and low level, the specific implementation procedure is shown as follows:

```
static char DHT11_read_byte ( void ){
  .... .... ....
  while(!(s3c2440_gpio_getpin(S3C2440_GPF(0))))
        //processing the low level that every bit starts
  .... .... ....
  while(s3c2440_gpio_getpin(S3C2440_GPF(0)))
        //processing the high level that every bit starts
  .... .... ....
 if (temp >6)  //high level exceeds 30μs,then the data is 1
   { DHT11_byte <<= 1 ;
     DHT11_byte |= 1 ;}
   else    //transmitting the data which is 0
    { DHT11_byte <<= 1 ;
      DHT11_byte |= 0 ;}
  .... .... ....
  }
```

Obviously, from the function above, we know when temp is more than 6,it has been delayed 30μs, so the sensor transmits the data is 1.On the contrary, when the temp is less than 6, the data it transmits is 0. Change 8 bits into character and achieve it by shifting, the lowest order of the data is 0, so the lowest order can be changed into the transmitting data by OR operation with 1 or 0, and it needs to DHT11_byte through OR operation with 0 when reading in 0.This process finishes receiving and processing a bit, as a character type contains 8 bits, so it's only needed to repeat the above process with 8 times.

C. The compiling and loading of driver

The Linux kernel compilation process is done by the kernel code root directory and subdirectories in the Makefile hierarchical management[5].Owing to the Linux operation system often has static and dynamic compiling mode for drive compiling, in terms of a compiled drive function, it is often compiled into the core directly and can be cut out freely when configuring the kernel, that is static compiling mode. Before compiling the drive, configure the core of compiling Linux, and then write the Makefile as follows:

```
ifeq ($(KERNELRELEASE),)
KERNELDIR ?=/home/peter/linux-2.6.32.2
PWD: = $(shell pwd)
modules: $(MAKE) -C $(KERNELDIR) M=$(PWD) modules
modules_install: $(MAKE) -C $(KERNELDIR) M=$(PWD) modules_install
clean:  rm -rf *.o *~ core .depend .*.cmd *.ko *.order *.markers *.mod.c *.mod.o *.symvers. tmp_versions
.PHONY: modules modules_install clean
else
  obj-m := dht11.o
endif
```

Input the order of "make", call dht11.ko module file which arm-linux-gcc generates, burn this file into ARM board, and use the order of insmod to load this module, the Dht11 module can start working by GUI interface module after loading.

D. The design of GUI user interface module

(1) GUI interface module construction and layout: on the basis of requirement analysis, the GUI module needs to supply visual data and recent varied situation of temperature and humidity. The GUI is based on embedded Linux, thus it starts carrying out immediately when ARM starts completely, and ceaselessly collecting the temperature and humidity data surrounding[6].The GUI user interface module displays the data on screen by chart forms. The interface display-class is defined as follows:

```
#include "sensorstat.h" //header file of custom sensor data
 .... .... ....
class SensorPlot : public QwtPlot
{  Q_OBJECT
public:
  ……  //defining temperature and humidity
  const QwtPlotCurve *sensorCurve( int id ) const
protected:
   void timerEvent( QTimerEvent *e );
private Q_SLOTS:
   void showCurve( QwtPlotItem *, bool on );
private:
   …… //temperature&humidity curve displays
   double timeData[HISTORY];
   SensorStat sensorStat;  //reading the sensor's data
};
```

The timerEvent(QTimerEvent *e) is a time accident function, the system calls it in a defined time interval, the time interval is determined by startTimer() function. The function of showCurve() is a slot function, being used to

forbid or allow displaying the curve chart of temperature and humidity. The timeData[HISTORY] is displayed on scales of abscissa.

(2)The initialization of GUI module: the GUI module is initialized in main function, through stating a QWidget object which is vBox, and other components are all added to vBox, then it is displayed on screen after completing the layout of window. After the finish of initialization, the reading module will read the sensor value. If the sensor can't work normally, it will remind users about the errors.

As Chinese acquiescently displayed in Qt is messy code, therefore, designated code is needed. Put in two header file that are QTranslator and QTextCodec in main function, besides, put in statements as follows:

QTextCodec::setCodecForTr(QTextCodec::codecForName("GB2312")); //When using Tr byte stream into encode of QString string, and designated as GB23 12.

QTextCodec::setCodecForCStrings(QTextCodec::codecForName("UTF-8")); //Specify QString class and code of byte stream, and is designated as UTF-8.

(3)The sensor reading module: The temperature and humidity passed to display module by void statistic (double & hum, double & tem) function. In lookUp() function, Qt opens the device of /dev/dht11,and Linux returns a device file descriptor at this time, then call the open() function to read data which the sensor transmits by this file descriptor.

As the preamble said, the sensor transmits 40bits data in one time, and they are putted in character array of buf after being all read, the temperature and humidity integers are stored in buf[0] and buf[2],other decimal parts are 0 forever. The check codes are putted in buf[5].The function of statistic() calls lookup() function to read data, and transfers it to GUI by parameter quote.

Because GUI has used Qwt, the Qwt header and library file directories are joined in the project file.

INCLUDEPATH += /usr/local/qwt-6.0.2/include

LIBS += -L"/usr/local/qwt-6.0.2/lib/" -lqwt

In this way, through the order of qmake to generate the file of Makefile, then it can contain the library of Qwt to pass through gcc compiling correctly.

V. THE TEST RESULTS AND CONCLUSION

The sensors node works in the platform of ARM9-Linux, and carries on capturing analysis and displaying of the surrounding temperature and humidity. Start the software and load DHT11 sensor's drive. The "#insmod dht11.ko" carries out "./senserplot -qws" starting software in catalogue. When the drive is running, we can through the touch tablet or mouse to decide showing temperature curve or humidity curve or not. The showing results of temperature and humidity is shown in Figure 4.

The thesis is based on S3C2440 microprocessor as the core and achieves DHT11 sensor digital temperature and humidity intelligent system. Construct cross-compiling environment which exploitation needs, study development of embedded Linux driver and Qt graphical interfaces, complete the connection between DHT11 sensor and ARM9, and the design of drive function. The system uses the micro-controller technology and embedded Linux operating system to process the system by modularity, realizes modularization of the temperature and humidity data acquisition, data processing module and user interface module. Through the practical application of specific, verifying the collaborative work between each module, temperature and humidity data acquisition and display is normal, the interaction with the user is normal, the acquisition process is continuous, stable and reliable. The data acquisition of temperature and humidity compared with the ordinary temperature and humidity meter, the error is smaller. The hardware of the system is low power consumption and simple connection, which is convenient assembly and carrying, software and hardware work concertedly and stably, through the modular design, which has good maintainability. In combination with the other functions of nuclear instrument has been applied in nuclear radiation field of online real-time monitoring of temperature and humidity has made a very good result.

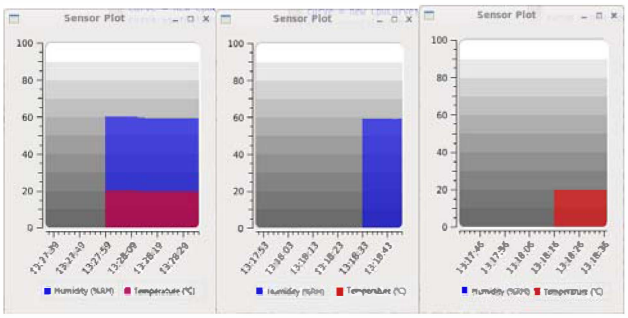

Figure 4. Displaying results of temperature and humidity

ACKNOWLEDGMENT

This work was supported by National Natural Science Fundation of China(No.11365001), and Natural Science Fundation from Jiangxi Province(No.20114BAB211026), This work was supported by Jiangxi Province Department of Education Fund for scientific and technological research(No. GJJ13464) and Open Science Fund from Engineering Research Center of Nuclear Technology Application(East China Institute of Technology), Ministry of Education (No.HJSJYB2011-08).

REFERENCES

[1] Samsung Corp. User Manual S3C2440A[EB/OL](2004-12-1)[2005-10-31]. http://www.samsungsemi.com.

[2] Resistance type digital sensor of temperature and humidity module DHT11 manual [EB/OL](2011-12-1)[2012-09-15]. http://www.aoson-g.com

[3] Zhang Guangjian, Liu Zheng. Embedded Linux driver development tutorial examples [M], Tsinghua University press, 2011

[4] Xu Min, Zhuge Zhenrong, SongJiaren.Based on 1Wire technology of digital temperature instrument[J]. Mechanical and electrical engineering, 2007, 24 (4). Pp. 26-29

[5] He Jianfeng, He Yueshun, Ye Zhixiang. Design of ARM/DSP communication interface in embedded Linux and driver development of [J]. instrument technique and sensor, 2009.5 (5): pp:47-49

[6] Lu Rongjian ,Li Pin,Sun Zhou .Application of SHT10 sensor in the temperature and humidity monitoring system.Sensor and micro system.2012.09.pp.136-145.