4.3 Environmental parameter acquisition program

The sensor measures the temperature, light intensity, and humidity of the environment as well as movement data of the vehicle itself, then sends the information to the ZigBee terminal nodes. After processing, the information is sent to the ZigBee coordinator wirelessly [9], then to the computer through the serial port, where it is displayed.

The program flow of the ZigBee coordinator for environmental parameter acquisition is shown in Fig. 9.The codes, head files declaration, macro definition, device descriptor and task initialization code subfunction [10-12] of the inspection vehicle are the same as the program controlled by buttons.

The following variable definition was added:

static void rxCB(uint8 port,uint8 event);

static uint8 SerialApp_Buf[SERIAL_APP_TX_MAX];

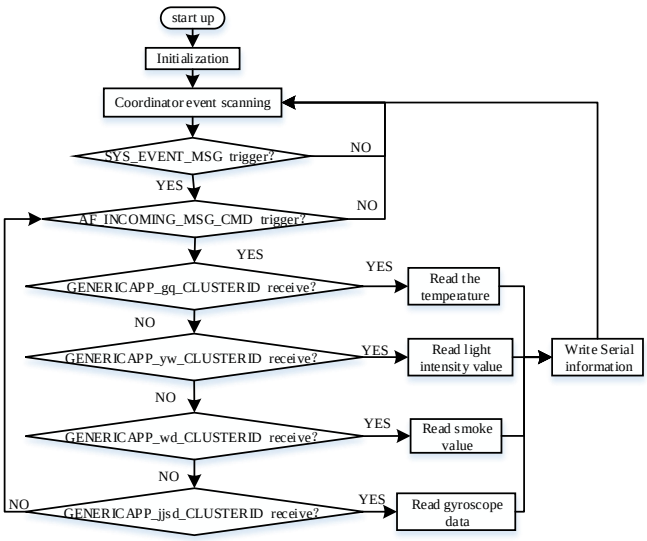

Figure 6. Coordinator software program flow for environmental parameter acquisition

Structure function:

typedef union w{ } TEMPERATURE;

typedef union y{ }SHIDU;

typedef union g{ }GUANGQIANG;

The task processing function has added:

*void GenericApp_MessageMSGCB (afIncomingMSGPacket_t *pkt){ }*

Serial port function:

static void rxCB(uint8 port,uint8 event){ }

The ZigBee terminal node program flow of environmental parameter acquisition is shown in Fig. 10.The head files declaration, macro definition, and device descriptor subfunction are consistent with corresponding subfunctions in the ZigBee terminal node program controlled by there mote buttons [10-13].

On the basis of the remotely controlled terminal node program, the following variable definition was added:

void GenericApp_Send_wd_Message(void);

void GenericApp_Send_gq_Message(void);

void GenericApp_Send_sd_Message(void);

void rxCB(uint8 port,uint8 event);

static uint8 SerialApp_Buf[SERIAL_APP_TX_MAX];

static uint8 SerialApp_Len;

uint16 myApp_ReadLightLevel(void);

uint16 myApp_ReadGasLevel(void);

int8 readTemp(void);

uint16 LightLevel;

afAddrType_t my_DstAddr;

Structure function:

typedef union w{ } TEMPERATURE;

typedef union s{ }YANWU;

typedef union g{ }GUANGQIANG;

On the basis of button control, task initialization includesthe serial port initialization function and wireless information transmission modes of the terminal nodes.

The task processing function would judge the scanned event, and call for scanned executable event function:

UINT16 GenericApp_ProcessEvent(byte task_id,UINT16 events){ }

After acquiring information from the sensor, the corresponding values are calculated and sent to the program of the coordinator:

void GenericApp_Send_wd_Message(void){ }

void GenericApp_Send_sd_Message(void){ }

void GenericApp_Send_gq_Message(void){ }

Serial communication subfunction:

static void rxCB(uint8 port,uint8 event)

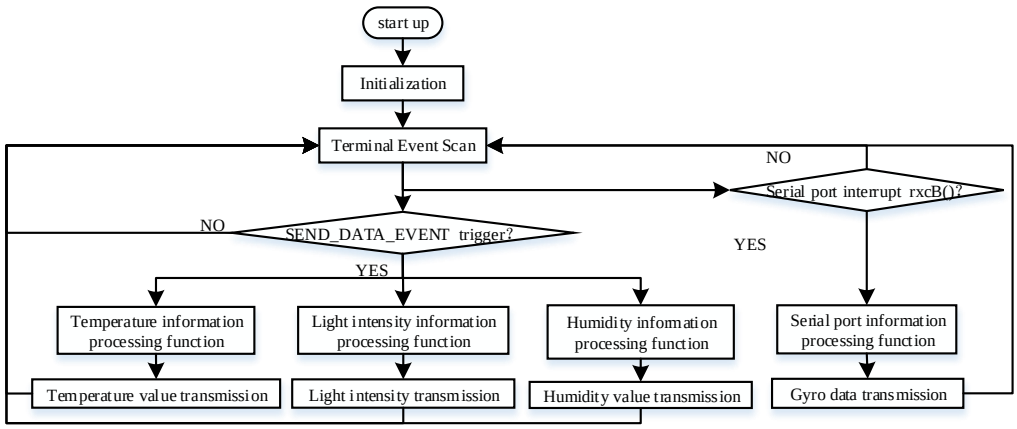

Figure 10. Terminal node program flow of environmental parameter acquisition

Based on the codes of coordinator h.file of the button-controlled inspection vehicle, the ZigBee coordinator h.file acquired via environmental parameters was added [13]:

```
#define GENERICAPP_wd_CLUSTERID2
#define GENERICAPP_sd_CLUSTERID 3
#define GENERICAPP_gq_CLUSTERID4
#define GENERICAPP_aj_CLUSTERID5
#define GENERICAPP_ck_CLUSTERID 6
#define GENERICAPP_jjsd_CLUSTERID  7
#define GENERICAPP_jsd_CLUSTERID   8
#define GENERICAPP_jd_CLUSTERID    9
#define SEND_DATA_EVENT              0x0001
#if !defined( SERIAL_APP_TX_MAX )
#define SERIAL_APP_TX_MAX  11
#endif
```

4.4 Program design ofserial port remotely controlled inspection vehicle

At the PC operation interface, the user presses the button to send serial port information to the ZigBee coordinator which then analyzes the serial port information and wirelessly sends corresponding information to the ZigBee terminal nodes. After being processed by the terminal nodes, the data is sent to the master controller of the inspection vehicle, then the vehicle moves accordingly.

The program flow of the ZigBee coordinator of the inspection vehicle is shown in Fig. 11.The head file declaration, macro definition, device descriptor, task initialization, and task processing subfunctions [10-12] are consistent with the subprograms of the ZigBee coordinator program of the remotely controlled inspection vehicle.

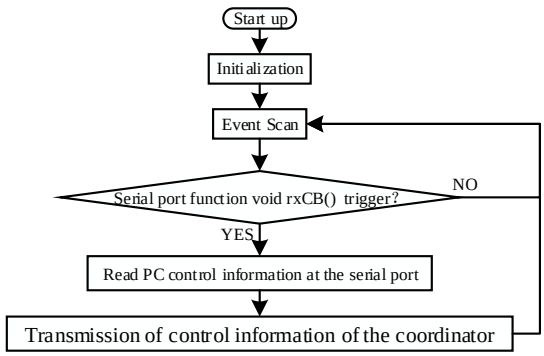

Figure 11. Coordinator software program flow of inspection vehicle remotely controlled via serial port

Variable definition:

static void rxCB(uint8 port,uint8 event);

Serial port function:

static void rxCB(uint8 port,uint8 event){ }

The program flow of the ZigBee terminal nodes of theremotely controlled inspection vehicle is shown in Fig. 12.The head file declaration, macro definition, device descriptor, variable definition, and task initialization subfunctions are consistent with corresponding subfunctions in the ZigBee terminal nodes of the button-controlled vehicle. Based on the subfunctions in the ZigBee terminal nodes, the task processing subfunction was given a code function controlled via serial port [10-13].

On the basis of the .h file of the vehicle's coordinator, the ZigBee coordinator .h file of the vehicle was added as follows:

```
#define GENERICAPP_ck_CLUSTERID     6
```

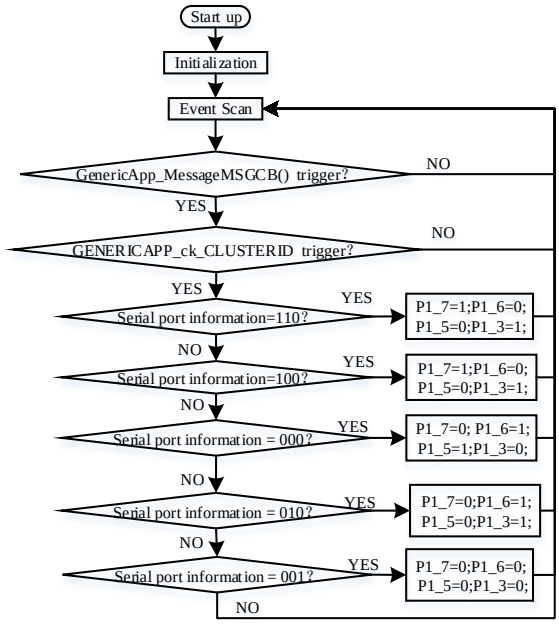

Figure 12. Terminal node program flow ofinspection vehicle remotely controlled via serial port

4.5 Upper computer software for remotely controlled inspection vehicle

The serial-port-controlled inspection vehicle interface, as built in Visual Studio 2010, is used to inspect button information and send it to the ZigBee coordinator through the serial port, then to the ZigBee terminal nodes. After receiving task data, the terminal nodes send it to the master controller to control the vehicle's movement. The program flow of the vehicle remotely controlled via serial port is shown in Fig. 13.

The environmental parameters acquired by the ZigBee terminal nodes, after being wirelessly sent to the coordinator, are written into the serial port. The data is processed in the upper computer software and displayed on the console in terms of certain norms and frequency [2, 3, 10, 12]. The program flow of the data acquisition center of the upper computer display interface is shown in Fig. 14.

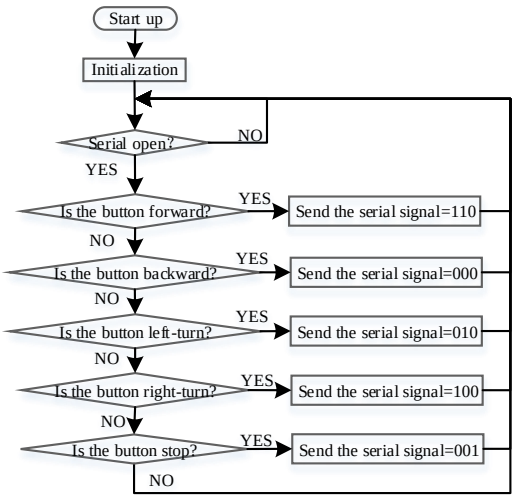

Figure 13. Program flow of upper computer software of remotely controlled inspection vehicle

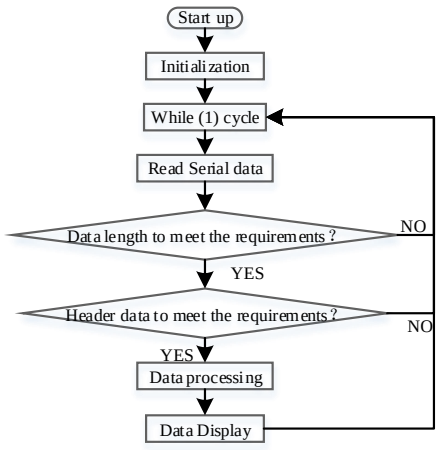

Figure 14. PC software program flow for data acquisition and display

5. CONCLUSIONS

In this study, we designed a greenhouse environment inspection control system based on ZigBee. The proposed system could be utilized in small greenhouses and laboratory greenhouses to allow remote inspection of the environmental information without necessitating that wireless sensor networks be established. The system is altogether quite economic and practical.

ACKNOWLEDGMENT

This research is supported by Water Science and Technology Project of Shaanxi Province in China (No. 2015slkj-11); Teaching reform project of Northwest A&F University (No. JY1302053); Students' Innovative Research Plan of Northwest A&F University (1201510712120)

REFERENCES

[1] J. Ma, L. Lin and X. Liu, "A fuzzy comprehensive evaluation of the applicability of intelligent greenhouse control systems," *Sensor Letters*, vol. 11, no. 6, pp. 1396-1402, 2013.

[2] D.H. Park, B.J. Kang and K.R. Cho, "A Study on greenhouse automatic control system based on wireless sensor network," *Wireless Personal Communications*, vol. 56, no. 1, pp. 117-130, 2011.

[3] J. Lin, Li Kong and F. Wang, "Wireless sensor network-based viticulture environmental monitoring system," *Journal of Drainage and Irrigation Machinery Engineering*, vol. 32, no. 7, pp. 637-644, 2014. (in Chinese with English abstract)

[4] S. Park, M. Choi and B. Kang, "Design and implementation of smart energy management system for reducing power consumption using ZigBee wireless communication module," *Procedia Computer Science*, vol. 19, pp. 662-668, 2013.

[5] Q. Zhang, X. Yang and Y. Zhou, "A wireless solution for greenhouse monitoring and control system based on ZigBee technology," *Journal of Zhejiang University: Science A*, vol. 8, no. 10, pp. 1584-1587, 2007.

[6] W. Yang, K. Lv and M. Li, "The wireless intelligent controller of greenhouse based on zigbee," *Sensor Letters*, vol. 11, no. 6, pp. 1321-1325, 2013.

[7] L. Gao, M. Cheng and J. Tang, "A wireless greenhouse monitoring system based on solar energy," *Telkomnika - Indonesian Journal of Electrical Engineering*, vol. 11, no. 9, pp. 5448-5454, 2013.

[8] F. Shariff, N.A. Rahim and W.P. Hew, "Zigbee-based data acquisition system for online monitoring of grid-connected photovoltaic system," *Expert Systems with Applications*, vol. 42, no. 3, pp. 1730–1742, 2015.

[9] X. Yan, "The design and research of greenhouse environmental monitoring system based on wireless network," *International Journal of Applied Environmental Sciences*, vol. 8, no. 13, pp. 1737-1746, 2013.

[10] H. Luo, P. Yang and Y. Li, "An intelligent controlling system for greenhouse environment based on the architecture of the internet of things," *Sensor Letters*, vol. 10, no. 1, pp. 514-522, 2012.

[11] T. Li, M. Zhang and Y. Ji, "Management of CO_2 in a tomato greenhouse using WSN and BPNN techniques," *International Journal of Agricultural and Biological Engineering*, vol. 8, no. 4, pp. 43-51, 2015.

[12] X. Li, X. Cheng and K. Yan, "A monitoring system for vegetable greenhouses based on a wireless sensor network," *Sensors* (Switzerland), vol. 10, no. 10, pp. 8963-8980, 2010.

[13] D.H. Park, B.J. Kang and K.R. Cho, "A study on greenhouse automatic control system based on wireless sensor network," *Wireless Personal Communications*, vol. 56, no. 1, pp. 117-130, 2011.