Advanced Science and Technology Letters
Vol.51 (CES-CUBE 2014), pp.81-85
http://dx.doi.org/10.14257/astl.2014.51.19

Real time synchronization for a small group of wireless sensor network system implemented with a DSP

JunJung, Jeong Bae Cho, Chol Hee Kwon, Sang Myung Lee

LIGNex1 Co.,Ltd.333,Pangyo-ro,Bundang-gu,Seongnam-City,Gyeonggi-do, 463-400,Korea
jun.jung@lignex1.com,jeongbae.cho@lignex1.com,
kwoncholhee369a@lignex1.com,
sangmyung.lee@lignex1.com

Abstract. The Wireless Sensor Networks (WSNs) consist of multiple sensor nodes and one gateway or one server which can gather each sensor node's data. The server makes information using the sensing data. Therefore it is important for the server to know whether the each sensor node's data is worthy to agglomerate or not. One of the agglomeration criteria is the time data of the detection message. When the sensor node placed in a shadow area, the GPS module is not available and if the RTC (Real Time Clock) module battery is week, time synchronization using the dsp cpu timer is effective.

Keywords: Sensor Network, Time Synchronization

1 Introduction

In recent, Wireless Sensor Networks (WSNs) is applied to various fields such as agricultural industry, machine measurement, and surveillance and reconnaissance system. The wireless sensor network system has several sensor nodes which can send a detection message to a gateway or a server. However if the server and the sensor node do not have a common time scale, the detection signal or message of the sensor node may be ignored by the server or the gateway for the reliability of gating or tracing a target. [4] Therefore many wireless sensor network system uses a GPS(Global Positioning System) or a RTC(Real Time Clock) module for the time synchronization.[6][7]

When the sensor node is placed in a shaded area or the battery of a RTC (Real Time Clock) is weak, it is difficult for a sensor node to synchronize a time. Therefore this paper proposes a solution for time synchronization without using a GPS (Global Positioning System) or a RTC (Real Time Clock) module.

2 Conceptual Model

ISSN: 2287-1233 ASTL

Wireless Sensor Networks (WSNs) consist of multiple sensor nodes and one gateway or one server which can gather each sensor node's data. The server makes information using the sensing data. Therefore it is important for the server to know whether the each sensor node's data is worthy to agglomerate or not. One of the agglomeration criteria is the time data of the detection message. If a sensing data is not matched to common time scale, the server will not use it. The Figure 1 shows the sequence of the time synchronization between a server and a sensor node. Because the RF transmission is critically related to the power consumption, it occurs only when a target is detected. Furthermore the server gives the server local time S1 only one time when the user wants to set the common time scale.

2.1. Chart Description

According to the conceptual model above, the timing synchronization process is designed as below.

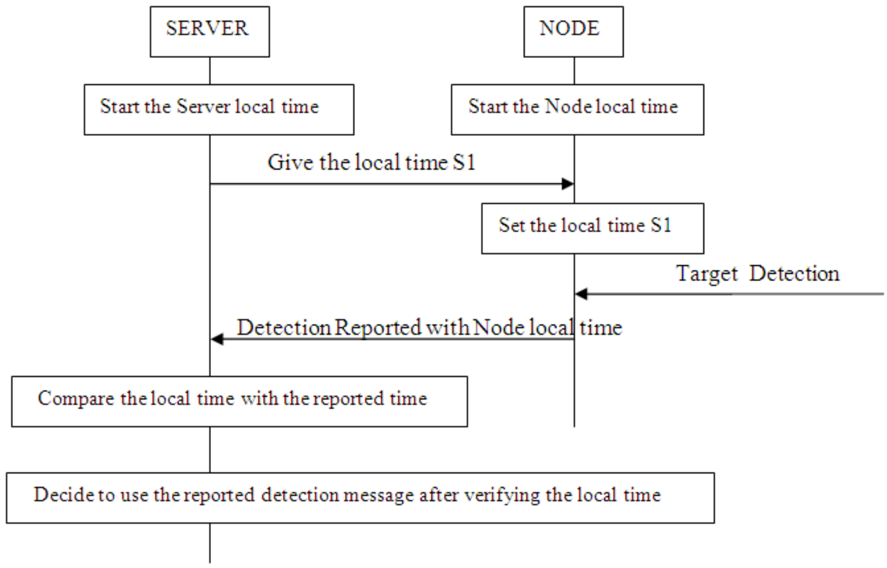

Fig. 1. Timing Synchronization Process Chart

2.2. Source Code

When a node starts the node local time, it uses the DSP (Digital Signal Processor) cpu timer. The following source code performs setting the node local time, and getting the server local time through the SCI (Serial Communication Interface) with MSP430 that controls the RF module. Compare to using the GPS timer or RTC (Real Time Clock) module, it can help to save the power consumption and complexity of the system.

```
#define StartCpuTimer0()   CpuTimer0Regs.TCR.bit.TSS = 0
void SetLocalClock()
{
        g_time = (Uint32)*(scib_rx_data) & 0x000000ff;
        g_time = g_time << 8;
        g_time = g_time | (Uint32)*(scib_rx_data+1) & 0x000000ff;
        g_time = g_time << 8;
        g_time = g_time | (Uint32)*(scib_rx_data+2)& 0x000000ff;
        g_time = g_time << 8;
        g_time = g_time | (Uint32)*(scib_rx_data+3) & 0x000000ff;
}
void SCIRXINTB_ISR()
{
        char i,rsr,len,firstbyte;
        rsr = ScibRegs.SCIRXST.all & 0xff; read
        if((rsr & 0x9c) != 0)
        {
         ScibRegs.SCICTL1.bit.SWRESET=0;
                ScibRegs.SCICTL1.bit.SWRESET=1;
        }
        else
        {
        len = (ScibRegs.SCIFFRX.all >> 8) & 0x1f;
        firstbyte = ScibRegs.SCIRXBUF.all;
        if(firstbyte == 0x07)          //time sync message header
        {
                for(i = 0; i < 4; i++)
                {
                        scib_rx_data[i] = ScibRegs.SCIRXBUF.all;
                }
                SetLocalClock();
}
```

According to the source code, the DSP (Digital Signal Processor) cpu timer starts when the StartCpuTimer is called. The main function calls the StartCpuTimer when the sensor node power up. And the DSP timer is configured before the StartCpuTimer is called by the configuration setting part. When the server gives its local time, the DSP gets the time via the sci interface and the interrupt service routine will set the time to the DSP local time. Because the server doesn't check the current time but its own time and the node detection message time, the relative time synchronization is more important to make information.

3 Experimental Measurements

To verify the proposed method, the four sensor node is deployed. Since a sensor node has a seismic sensor, it can populate a detection message when a man walks around it. First, the server and four sensor node has started their own local time, then the server give its local time to the each sensor node. Then, a man makes seismic signal near the sensor node to populate a detection message. The server gets a detection message of a sensor node with the its local time, and verify the time scale which is already set at first time between the server and sensor node. If the time scale is not matched, the message is ignored.

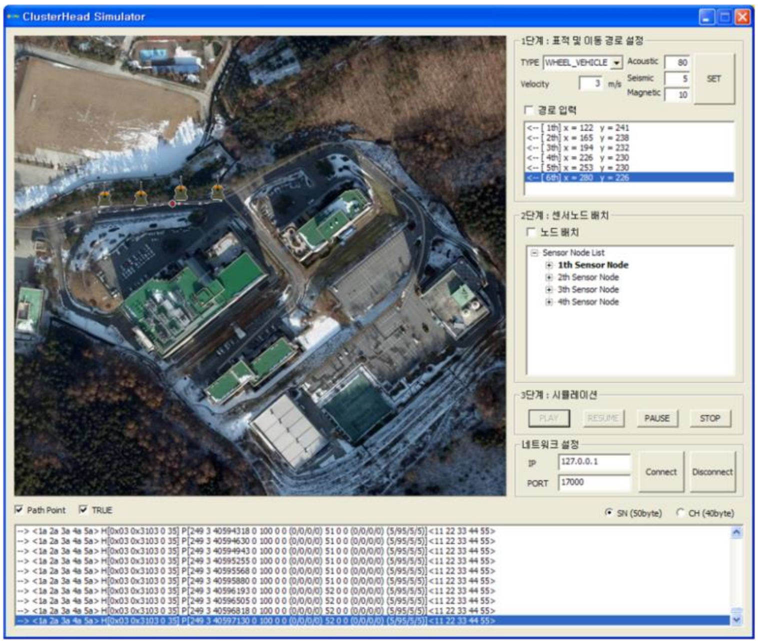

Fig. 2. Time Synchronization Status

4 Experimental Result

The Figure 2 shows the time synchronization status of each sensor node. And the Figure 3 shows the result of populating detection message with having the common time scale. Each node has been tested 10 times, and depending on k value it has three cases. If the k value is meet to the condition, it counts to success.

k (duration criteria)	Node #1	Node #2	Node #3	Node #4	Average
k < ∣ 1s ∣	60%	50%	60%	60%	57.5%
∣ 1s ∣ < k < ∣ 2s ∣	70%	80%	70%	70%	72.5%
∣ 2s ∣ < k < ∣ 3s ∣	80%	90%	80%	80%	82.5%

Fig. 3. Time Synchronization results depend on the k value.

The k value is the server's duration criteria of the agglomeration.
k = S1(refer to the figure 1.) – Node's local time.

5 Conclusion

When the sensor node placed in a shadow area, the GPS module is not available and if the RTC (Real Time Clock) module battery is week, time synchronization using the dsp cpu timer is effective. As seen above, a relatively simple method of time synchronization has succeeded in a more precise time synchronization which is cost-effective in terms of the reference of the future can be used as a reference point.

References

1. Scott, C. D. and Smalley, R. E.: J. Nanosci. A Programming Model for Time-Synchronized Distributed Real-Time Systems (2007)
2. Ratnaparkhe, T., Natekar, S., Chandan, S., Sadaphal, V.: Selection of Time Synchronizing Nodes in WirelessSensor Network(2010)
3. KHEDIRI, S. E., Nasri, N., Kachouri, A., Wei, A. : Synchronization in Wireless Sensors NetworksUsing balanced clusters (2013)
4. Ikram, W., Dr. Stoianov, I., Thornhill, N.: Towards a Radio-Controlled Time Synchronized Wireless Sensor Network: AWork in-Progress Paper (2010)
5. Sundararaman, B., Buy, U., Kshemkalyani, A. D.: ClockSynchronisation for Wireless Sensor Networks (2005)
6. Yoon, J. H.: A Synchronization of Secure Sensor terminal by GPS time information (2007)
7. Yoon, D.: Sylvester, D., Blaauw, D.: A 5.58nW 32.768kHz DLL-assisted XO for real-time clocks in wireless sensing applications (2012)
8. Salim, EL., NASRI, N., WEI, A. and KACHOURI, A.: Survey of Synchronization Algorithms: Wireless Sensor Networks CaseStudy. The second international conference on communicationand information Technology (ICCIT 2012)

