

การปรับปรุงความแม่นยำของระบบค้นหาตำแหน่งภายในอาคาร AN ACCURACY IMPROVEMENT OF INDOOR LOCAL POSITIONING SYSTEM

สนธยา วิลัยจิตต์¹ และ เขมะทัต วิภาตะวนิช²

Sontaya Wilaijit and Kemathat Vibhatvanij

บทคัดย่อ

บทความนี้นำเสนอผลการพัฒนาชุดทดลองทางด้าน Wireless Sensor Network และระบบสมองกลฝังตัว (Embedded System) โดยผลสำเร็จหลักคือการพัฒนากระบวนการปฏิบัติการของ TinyOS ให้เหมาะสมกับอุปกรณ์ที่นำมาทดลองและพัฒนาอุปกรณ์ดังกล่าวเพื่อใช้สำหรับทดลองระบบค้นหาตำแหน่งของวัตถุภายในอาคารโดยใช้หลักการค้นหาตำแหน่งด้วยวิธี Time Difference Of Arrival (TDOA) ซึ่งวิธีดังกล่าวมีข้อดีคือไม่ต้องทำการ Clock Synchronize ระหว่างวัตถุกับอุปกรณ์ตัวรับสัญญาณซึ่งเหมาะกับระบบขนาดใหญ่และสามารถเพิ่มความแม่นยำของการระบุตำแหน่งของวัตถุ ให้มากขึ้นได้ด้วยการเพิ่มอุปกรณ์ตัวรับสัญญาณซึ่งติดตั้งเป็นแบบกริดบนเพดานแล้วทำการส่งข้อมูลดังกล่าวแบบ Multihop ประเภท Table Driven มาคำนวณที่ศูนย์กลาง ซึ่งผลงานนี้เหมาะกับงานในหลาย ๆ ด้าน เช่น ระบบนำทางของหุ่นยนต์ ระบบสำนักงานอัตโนมัติ หรือระบบรักษาความปลอดภัย เป็นต้น

ABSTRACT

This paper presents the result of the development of a new approach in using Wireless Sensor Network and embedded system. The major success of the development is developing the hardware of local positioning system by using Time Difference of Arrival (TDOA) method. The system uses TinyOS and Multihop Communication on the saving energy hardware which is appropriate for robot navigation, office automation and security system.

Key Words: TinyOS, nesC, radio stack, time difference of arrival (TDOA)

W Sontaya: g4665305@ku.ac.th, pop@ku.ac.th

^{1,2}ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์

Department of Computer, Faculty of Engineering, Kasetsart University.

คำนำ

ด้วยเทคโนโลยีทางด้าน Wireless Sensor Network ที่ถูกนำมาใช้ในหลายๆด้าน เช่นทางการทหาร การเก็บข้อมูล หรือทางการแพทย์ จึงนำเอามาประยุกต์ใช้ในการค้นหาตำแหน่งของวัตถุ ซึ่งปัจจุบันได้มีการพัฒนาทั้งทางด้านอุปกรณ์ซึ่งมีความประหยัดพลังงานมากขึ้นอีกทั้งมีขนาดที่เล็กลงและราคาถูกลงทางด้านระบบปฏิบัติการได้มีการพัฒนาในหลายๆด้าน เช่นความคล่องตัวในการใช้กับซีพียูตระกูลต่างๆสามารถทำได้ง่ายโดยไม่ต้องเขียนโปรแกรมขึ้นมาใหม่ทั้งหมดและความประหยัดพลังงานทั้งสองส่วนนี้ ทำให้ในปัจจุบันและในอนาคตระบบ Wireless Sensor Network จะมีบทบาทอย่างมากในชีวิตประจำวันของมนุษย์มากขึ้นซึ่งในปัจจุบันสังเกตได้จากอุปกรณ์รอบตัวเราซึ่งประกอบไปด้วยสมองกลขนาดเล็กฝังตัวอยู่เป็นอุปกรณ์ควบคุมการทำงานเช่นโทรศัพท์มือถือ, เตาไมโครเวฟ, เครื่องเล่นเอ็มพีสาม เป็นต้น และมีแนวโน้มว่าจะได้รับความนิยมมากขึ้นต่อไปในอนาคต

เทคนิคที่ใช้ในการค้นหาตำแหน่งของวัตถุ มีหลายวิธีโดยแบ่งเป็นกลุ่มใหญ่ได้แก่ TDOA [1] คือการค้นหาตำแหน่งของวัตถุโดยใช้ความแตกต่างของสัญญาณที่ส่งมาถึงจุดรับตั้งแต่ 3 จุดขึ้นไป, Time Of Arrival (TOA) [9] คือการวัดค่าเวลาที่เดินทางจากวัตถุถึงจุดรับสัญญาณตั้งแต่ 3 จุดขึ้นไป, Received Signal Strength Indicator (RSSI) คือการค้นหาตำแหน่งของวัตถุ (MS) โดยการตรวจวัดจากความเข้มของสัญญาณที่วัดได้, Angle Of Arrival (AOA) [5][9] คือการค้นหาตำแหน่งของวัตถุ จากทิศทางของสัญญาณที่ส่งออกมาจากวัตถุ ถึงจุดรับสัญญาณจำนวนตั้งแต่ 3 จุดขึ้นไป, Finger Print คือการนำค่าความเข้มของสัญญาณที่วัดได้มาทำการหาตำแหน่งของวัตถุจากฐานข้อมูลที่ได้เก็บไว้ก่อนหน้านี้ ซึ่งวิธีดังที่กล่าวมานี้มีข้อดีและข้อเสียอีกทั้งความแม่นยำที่แตกต่างกันออกไป

TinyOS

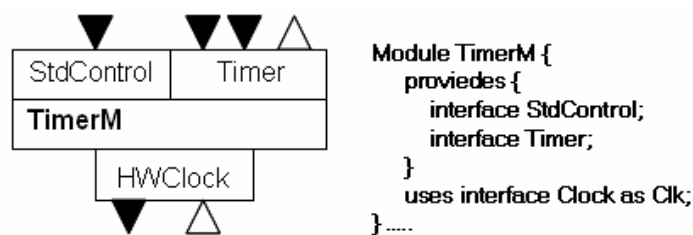
TinyOS เป็น Operating System แบบ Event Base Operating System ซึ่งเริ่มต้นถูกพัฒนาจากมหาวิทยาลัย Berkeley โดยภาษาที่ใช้เรียกว่า NesC [11] พัฒนาสำหรับ Wireless Sensor Network ทำงานได้กับ Hardware หลาย Platform

TinyOS Scheduler

TinyOS Scheduler ประกอบด้วย Scheduler 2 Level ได้แก่ Event และ Command คือ Task ที่ทำให้สำเร็จได้ในช่วงเวลานั้นๆ เช่นกลุ่มของคำสั่งเพื่อหยุดการทำงานของภาค Analog to Digital Converter (ADC) ซึ่งประกอบด้วยคำสั่งจำนวนไม่มากโดยถูกเรียกใช้จาก Interrupt ของ Hardware เช่น Timer Interrupt หรือ ADC Interrupt เป็นต้นและใน Scheduler ส่วนที่ 2 คือ Task ที่อยู่ในคิวที่ต้องกระทำแบบ First In First Out (FIFO) และถูก Interrupt จาก Hardware ได้ และเมื่อ CPU ไม่มี Task อยู่ในคิวก็จะเข้าสู่โหมด Sleep จนกระทั่งถูก Interrupt อีกครั้งหนึ่ง

NesC

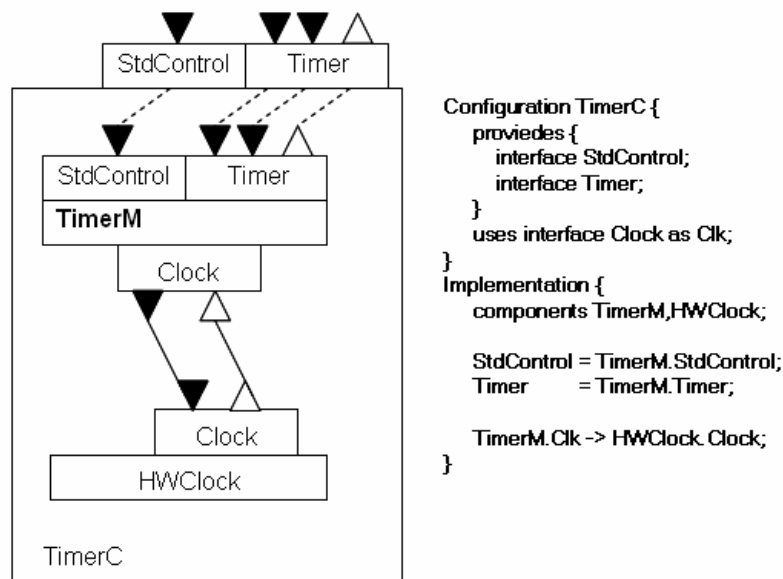
NesC คือชื่อภาษาสำหรับการเขียนโปรแกรมบนระบบปฏิบัติการ TinyOS [3][11] ซึ่งมีความคล้ายกับภาษาซีมาตรฐานซึ่งรูปแบบการทำงานของ TinyOS นั้นเป็นแบบ Event-Driven คือโปรแกรมจะทำงานเมื่อเกิดเหตุการณ์ต่างๆมา Interrupt โดย Component ของ TinyOS แบ่งออกเป็น 3 กลุ่ม[12] ดังนี้ 1. Component ใน กลุ่มของ Thin Abstract เช่น Clock , Bit Level Radio Component 2. กลุ่มของ Component ที่ทำหน้าที่เป็นตัวแทนในส่วนที่ Hardware ไม่สามารถให้บริการได้โดยตรงเช่น Byte Level Radio Component ที่ระดับบนสุดของ Radio Component 3. Component ที่ทำหน้าที่อยู่ระดับบนสุดของ Application โดยการเขียน NesC แบ่งกลุ่มของ Component เป็น 2 กลุ่มหลักประกอบด้วย Module และ Configuration



รูปที่ 1 แสดง Module ของ TinyOS

จากรูปที่ 1 เป็นตัวอย่างของ Module Timer โดย Module จะมีคำสั่งอยู่ภายในเช่น Module สำหรับ Timer จะประกอบด้วยคำสั่งต่างๆเช่นการกำหนดค่าเริ่มต้นในรีจิสเตอร์ หรือ Module ควบคุมการสื่อสารแบบ Multihop จะประกอบไปด้วยคำสั่งสำหรับการ

พิจารณาหา Parent Node Address เป็นต้น แต่ละ Module จะมี Interface สำหรับเชื่อมต่อกับ Module อื่นๆ ให้เรียกใช้งานได้สะดวก เช่น Module Timer จะมี Interface สำหรับกำหนดช่วงเวลาหรือ Module ควบคุมการรับ-ส่งแบบ Multihop จะมี Interface สำหรับ Send และ Receive เป็นต้น โดย Interface นั้นแบ่งเป็น 2 กลุ่มคือ Command ซึ่งเป็น Interface สำหรับเรียกโปรแกรมย่อยที่อยู่ใน Module นั้นๆ และ Event ซึ่งเป็น Interface สำหรับเรียกเมื่อเกิด Interrupt จาก Module โดยเราสามารถเรียกใช้ Module และ Interface ได้หลายครั้ง



รูปที่ 2 แสดง Configuration ของ TinyOS

รูปที่ 2 แสดงส่วนของ Configuration ซึ่งเป็นการรวมเอา Module ต่างๆ ที่เชื่อมต่อกันผ่านทาง Interface มารวมกันเพื่อสะดวกในการใช้งานซึ่งการแบ่งกลุ่มของ Component ดังกล่าวมีข้อดี เช่นเมื่อต้องการพัฒนาระบบปฏิบัติการให้สามารถเข้ากับชิพตระกูลอื่นๆ สามารถทำได้กับ Component ในกลุ่มที่ 1 และ 2 โดยที่ Component ในกลุ่มที่ 3 สามารถนำมาใช้ได้โดยไม่ต้องเขียนใหม่ซึ่งในปัจจุบันนี้ TinyOS สามารถนำไปใช้ได้กับชิพตระกูลต่างๆ เช่น ATMEGA108, MSP430 หรือ PIC เป็นต้น

อุปกรณ์และวิธีการ

Node ที่ใช้ในการทดสอบระบบนี้ใช้ TinyOS ซึ่งเป็นระบบปฏิบัติการที่นิยมใช้อย่างแพร่หลายในงานด้าน Wireless Sensor Network ซึ่งบอร์ดนั้นประกอบด้วย CPU MSP430, CC1000 [7] ทำหน้าที่รับส่งคลื่นวิทยุที่มีความถี่ 433.005 MHz ที่ Baud rate 9600 BPS, SRF05 ทำหน้าที่รับส่งสัญญาณอัลตราโซนิกที่มีความถี่ 40 KHz และ FT232BM เป็นส่วนควบคุมการรับส่งข้อมูลกับอุปกรณ์ภายนอกผ่านทาง Serial port(RS232) หรือ USB Port

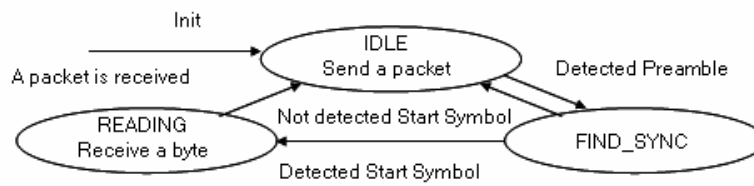


รูปที่ 3 บอร์ดทดสอบการทำงาน

จากรูปที่ 3 แสดงบอร์ดที่ใช้ในการทดสอบการทำงานของบทความนี้ซึ่งได้ออกแบบโดยศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (NECTEC)

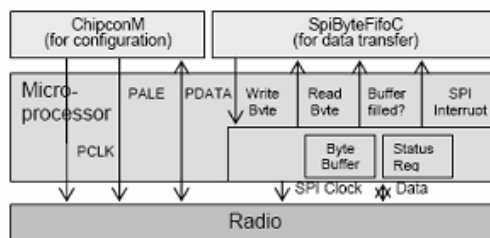
Network Stack ของ TinyOS

ส่วนที่สำคัญส่วนหนึ่งของ Wireless Sensor Network คือส่วนการรับ-ส่งคลื่นวิทยุ บอร์ดการทดลองที่ใช้ในบทความนี้ระหว่าง CPU กับ CC1000 นั้นติดต่อกันผ่านทาง Serial Peripheral Interface (SPI) ที่มีอยู่ใน CPU โดย กำหนดให้ ซีพียูอยู่ในโหมด Slave ซึ่ง CC1000 จะทำการ Interrupt ซีพียู ทุกๆ 8 Clock Cycle (1 ไบท์)

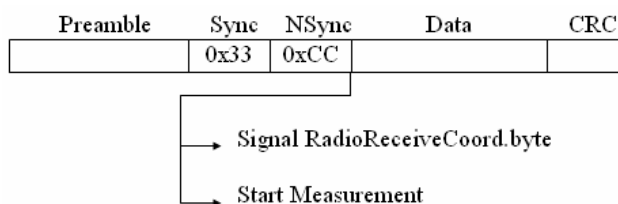


รูปที่ 4 State Transition ของ Packet

รูปที่ 4 แสดงการ Decomposition and Reassembly ข้อมูลที่อ่านได้ซึ่งจะกระทำเมื่อซีพียูถูก Interrupt ทุกๆ 1 Byte ก็จะทำนำข้อมูลที่อ่านได้นั้นไปทำการหา Preamble (0x55) และ Sync Byte (0x33) จนเมื่อพบ Sync Byte แล้วก็จะนำข้อมูลที่ได้อ่านมานั้นเก็บเป็น Packet เพื่อส่งให้ส่วนอื่นทำหน้าที่ต่อไปหากยังไม่พบก็จะอยู่ใน State IDLE ต่อไป ซึ่ง Module ทั้งหมดจะทำงานร่วมกัน[14] โดย GenericComm เป็น Configuration ทำหน้าที่รวบรวมเอา Module ต่างๆเข้าไว้ด้วยกัน , ActiveMessage(AM) ทำหน้าที่คัดเลือกว่า Packet นั้นเป็น Packet ของส่วนรับส่งคลื่นวิทยุหรือเป็น Packet ในส่วนของการรับส่งผ่านทาง Serial port(RS232) หรือ USB, CC1000RadioC Configuration ทำหน้าที่สำหรับรวบรวม Module ที่เกี่ยวข้องกับ CC1000 ทั้งหมด, CC1000RadioIntM เป็น Module ที่กำหนดค่าเริ่มต้นของ CC1000 เช่น Baudrate หรือ Data Format เป็นต้น, SpiByteFIFO เป็น Module ที่กำหนดค่าเริ่มต้นของ SPI เช่นการกำหนด Baudrate ของซีพียูกับ CC1000, RandomLFSR เป็น Module ที่สร้างค่าสุ่มแบบ Linear Feedback Shift Registers (LFSR) เพื่อใช้ในการ Backoff เมื่อต้องการส่งข้อมูลแล้วช่องสัญญาณไม่ว่าง, ADCM เป็น Module ที่ควบคุมการทำงานของภาค ADC เพื่อใช้ในการวัดค่าต่างๆที่ต่อเชื่อมกับบอร์ดทดลองและเพื่อวัดค่าของช่องสัญญาณวิทยุว่าว่างหรือไม่เป็นต้น ซึ่งจากการทดลองพบว่า การรับส่งข้อมูลจะเกิดความผิดพลาดอยู่ที่ประมาณ 5-10%



รูปที่ 5 แสดงการเชื่อมต่อระหว่างซีพียูและ CC1000

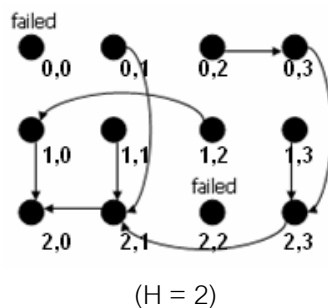


รูปที่ 6 แสดงจุดเริ่มต้นของการนับ Internal Clock เมื่อได้รับสัญญาณ Beacon

Routing protocol

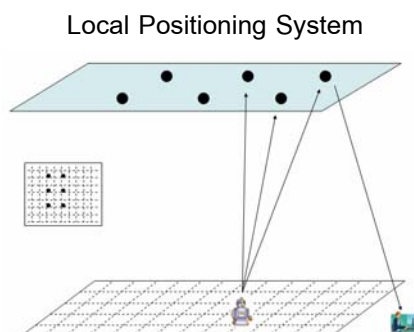
ในการทดสอบนี้ได้ใช้ Routing Protocol แบบ Table Driven โดยใช้ชื่อว่า GridRouting [2] ซึ่งเหมาะสมกับลักษณะการวางตัวของ Node (BS) ซึ่งมีลักษณะเป็นกริด โดยมีขั้นตอนการทำงานเริ่มจากการทำ Spanning Tree เพื่อหา Neighbors Node เริ่มจาก Root Node จะทำการส่ง Connected Message ทุกๆ n วินาที ให้กับ Node ที่อยู่ใกล้เคียงโดย Node ทั้งหมดจัดเรียงเป็น Grid ขนาด M*N โดยมี Root Node อยู่ที่ (0,0) แต่ละ Node จะมีจุดอ้างอิงเป็น (i,j) โดย $i=0..M-1$ และ $j=0..N-1$ ซึ่งแต่ละ Node จะมี Node ข้างเคียง (Neighbor) คือ Low-Neighbor (i-H, j) and (i, j-H) และ High-neighbors (i+H, j) and (i, j+H) ถ้า Node ได้รับ Packet ที่

ไม่ใช่ Neighbors Node ก็จะไม่สนใจ Packet ดังกล่าวแต่หากเป็น Packet จาก Node ข้างเคียงก็จะทำการส่ง Packet ดังกล่าวต่อไป
ยัง Low-Neighbor หรือ High-Neighbor ที่เป็น Parent Node ของตัวเองไปเรื่อยๆ จนถึง Root Node หาก Low-Neighbor ไม่มีก็จะ
เลือก High-Neighbor มาเป็น Parent ซึ่งเรียกว่า Inversion (H)



รูปที่ 7 แสดงการทิศทางการส่งข้อมูลของ Grid Routing เมื่อ Node ข้างเคียง Fail

ตัวอย่างเช่นเมื่อ Node (2,2) Fail Node(1,2) ก็ทำการส่งต่อ Packet ไปยัง (1,0) เป็นต้นในแต่ละ Node นั้นจะทำการรักษา
ข้อมูล 2 อย่างได้แก่ Parent และค่า Inversion Count ถ้าแต่ละ Node มี Parent Node นั้นก็จะทำการส่ง Connected Message ไปให้
Parent Node ทุกๆ n วินาที



รูปที่ 8 หลักการทำงานของระบบ

รูปที่ 8 แสดงหลักการทำงานของระบบเริ่มจากวัตถุส่งสัญญาณคลื่นวิทยุ (Beacon) ที่ประกอบด้วยรหัสของวัตถุ (ID) นั้น
พร้อมกับการส่งสัญญาณอัลตราโซนิก เมื่อ BS ที่อยู่บนเพดานจะรับสัญญาณ Beacon ได้ก่อน (เนื่องจากคลื่นวิทยุเดินทางได้เร็วกว่า
สัญญาณอัลตราโซนิก) ก็จะเริ่มนับสัญญาณนาฬิกาจากฐานเวลาภายในซีพียูไปจนกว่าจะรับสัญญาณอัลตราโซนิกได้ก็จะทำการหยุด
นับสัญญาณนาฬิกาแล้วส่งสัญญาณที่นับได้นำไปคำนวณที่เครื่อง PC ซึ่งเชื่อมต่ออยู่กับ Root Node ซึ่งจะทำเช่นนี้กับทุก Node ที่รับ
สัญญาณ Beacon ได้ซึ่งโดยหลักการนี้จะสามารถคำนวณหาพิกัดตำแหน่งของวัตถุได้นั้นจำเป็นต้องได้ค่าจากการนับตั้งแต่ 3 Node
ขึ้นไป

ปัจจัยที่มีผลต่อความแม่นยำในการระบุตำแหน่งของวัตถุนั้นได้แก่ความถี่และอุณหภูมิซึ่งมีผลต่อความเร็วในการเดินทางของ
คลื่นอัลตราโซนิก

Time of Arrival (TOA)

ด้วยวิธีการของ TOA นั้นใช้หลักการของการวัดเวลาที่ใช้ในการเดินทางระหว่างวัตถุ หรือ Mobile Station (MS) และ Base
Station(BS) ซึ่งความเร็วในการเดินทางของคลื่นวิทยุ (C) จึงเขียนสมการของระยะห่างระหว่าง MS_i และ BS_j ได้เป็น

$$r_i = (t_i - t_0)C$$

โดย t_0 คือ เวลาที่เริ่มต้นที่ MS และ t_i คือเวลาที่คลื่นวิทยุเดินทางมาถึง BS ด้วยวิธีการดังกล่าวต้องใช้ BS อย่างน้อย 3 BS จึงเขียน
สมการได้เป็นดังนี้

$$r_1^2 = x_0^2 + y_0^2 \quad (1)$$

$$r_2^2 = (x_2 - x_0)^2 + (y_2 - y_0)^2 \quad (2)$$

$$r_3^2 = (x_3 - x_0)^2 + (y_3 - y_0)^2 \quad (3)$$

เมื่อนำสมการ (1)-(2) จะได้

$$r_2^2 - r_1^2 = x_2^2 - 2x_2x_0 + y_2^2 - 2y_2y_0$$

เมื่อนำสมการ (1)-(3) จะได้

$$r_3^2 - r_1^2 = x_3^2 - 2x_3x_0 + y_3^2 - 2y_3y_0$$

เมื่อ กำหนด

$$K_i^2 = x_i^2 + y_i^2$$

จะได้สมการในรูปของ

$$H = \begin{bmatrix} x_2 & y_2 \\ x_3 & y_3 \end{bmatrix}, x = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}, b = \frac{1}{2} \begin{bmatrix} K_2^2 - r_2^2 + r_1^2 \\ K_3^2 - r_3^2 + r_1^2 \end{bmatrix}$$

ซึ่งอยู่ในรูปของ

$$Hx = b$$

ดังนั้น x หาได้จาก

$$x = H^{-1}b$$

จากสมการของวิธี TOA นั้นจะเห็นว่าต้องทำการ Clock Synchronize ระหว่าง MS และ BS ซึ่งหาก Clock ของทั้งสองผิดพลาดไปเพียง $1 \mu s$ อาจทำให้การคำนวณตำแหน่งของวัตถุ ผิดพลาดไปถึง 100 เมตร

Time Difference of Arrival (TDOA)

กำหนด

$$r_{i,1} = r_i - r_1$$

$$r_{i,1} = (t_i - t_0)C - (t_1 - t_0)C = (t_i - t_1)C$$

เมื่อ $r_{i,1}$ คือความแตกต่างของระยะทางระหว่าง Node₁ กับ Node_i ซึ่งจะเห็นว่าวิธี TDOA ไม่จำเป็นต้อง Clock Synchronization ระหว่าง MS และ BS

$$(r_{2,1} + r_1)^2 = K^2 - 2x_2x_0 - 2y_2y_0 + r_1^2$$
$$-x_2x_0 - y_2y_0 = r_{2,1}r_1 + \frac{1}{2}(r_{2,1}^2 - K_2^2)$$

ในทำนองเดียวกัน

$$-x_3x_0 - y_3y_0 = r_{3,1}r_1 + \frac{1}{2}(r_{3,1}^2 - K_3^2)$$

ซึ่งเขียนให้อยู่ในรูปของ

$$Hx = cr_1 + d$$

โดยให้

$$c = \begin{bmatrix} -r_{2,1} \\ -r_{3,1} \end{bmatrix}, d = \frac{1}{2} \begin{bmatrix} K_2^2 - r_{2,1}^2 \\ K_3^2 - r_{3,1}^2 \end{bmatrix}$$

$$x = H^{-1}cr_1 + H^{-1}d$$

เมื่อ

$$H = \begin{bmatrix} x_2 & y_2 \\ x_3 & y_3 \\ x_4 & y_4 \\ \dots & \dots \\ \dots & \dots \end{bmatrix}, c = \begin{bmatrix} -r_{2,1} \\ -r_{3,1} \\ -r_{4,1} \\ \dots \\ \dots \end{bmatrix}, d = \frac{1}{2} \begin{bmatrix} K_2^2 - r_{2,1}^2 \\ K_3^2 - r_{3,1}^2 \\ K_4^2 - r_{4,1}^2 \\ \dots \\ \dots \end{bmatrix}$$

หา x ได้จาก

$$x = (H^T H)^{-1} H^T (cr_1 + d)$$

ด้วยข้อดีของวิธีการดังกล่าวพบทความนี้จึงใช้วิธีการค้นหาตำแหน่งของวัตถุด้วยวิธี TDOA ซึ่งมีข้อดีคือไม่ต้องทำ Clock Synchronize ระหว่างวัตถุ และ BS และสามารถปรับปรุงความแม่นยำของระบบให้เพิ่มขึ้นได้โดยการเพิ่ม BS [1]

ผลการทดลอง

จากการทดสอบเมื่อนำค่า Time Of Flight (TOF) ที่ได้จากระยะต่างๆมาทำการคำนวณจะได้สมการเส้นตรงด้วยวิธี Least Square Method (LSM) ได้สมการดังนี้

$$\text{Value} = 1.306 D + 9.88$$

ทำการทดลองโดยวาง BS 3 จุด ได้แก่ (700,100), (200,100) และ (200,500), โดยวางวัตถุไว้ที่ตำแหน่ง (450,300) ผลลัพธ์จากการทดสอบเป็นจำนวน 10 ครั้ง ได้ผลดังนี้

ครั้งที่	ตำแหน่ง X	ตำแหน่ง Y	ผิดพลาด X	ผิดพลาด Y
1	449.9610	299.9513	3.9000	4.8700
2	449.9200	299.9513	8.0000	4.8700
3	450.0000	300.0512	0.0000	5.1200
4	449.9610	300.0000	3.9000	0.0000
5	450.0000	300.0000	0.0000	0.0000
6	449.9610	299.9513	3.9000	4.8700
7	449.9610	299.9000	3.9000	10.0000
8	449.9610	299.9000	3.9000	10.0000
9	450.0000	299.9000	0.0000	10.0000
10	449.9590	300.0000	4.1000	0.0000

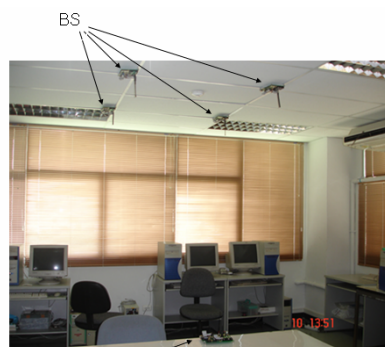
ตารางที่ 1 แสดงตำแหน่งของวัตถุ

จากตารางที่ 1 แสดงตำแหน่งของวัตถุซึ่งค่าความผิดพลาดที่ได้อยู่ที่ประมาณ 8 เซนติเมตรในแกน X และ 10 เซนติเมตรในแกน Y จากการทดลองยังได้ทดสอบในส่วนของการเพิ่มลระยะเวลาในการ Spanning Tree ว่ามีผลกับความแม่นยำที่ได้หรือไม่ซึ่งได้ผลดังตารางด้านล่าง

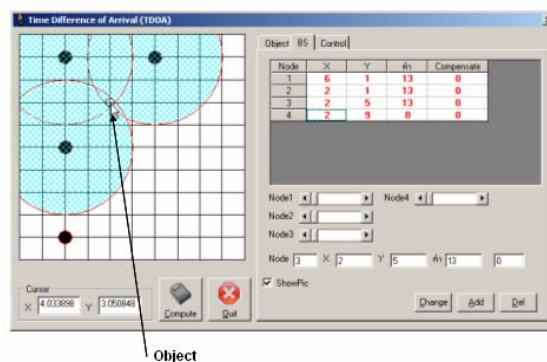
ความถี่ในการ Spanning Tree (วินาที)					
ระยะทาง(cm)	1	2	3	4	5
105	12.56	12.60	12.63	12.72	12.71
110	13.23	13.12	13.17	13.10	13.08
115	13.56	13.57	13.19	13.24	13.21
120	14.34	14.30	14.33	14.33	14.33
125	14.83	14.78	14.79	14.81	14.79
130	15.28	15.35	15.24	15.28	15.23
180	20.47	20.55	20.54	20.46	20.54
185	20.94	21.01	21.25	21.18	21.05
190	21.44	21.45	21.47	21.46	21.43

ตารางที่ 2 ค่าเฉลี่ยที่ได้จากการวัด 100 ครั้ง

จากผลการทดลองสรุปได้ว่าค่าที่วัดได้จาก BS ไม่มีผลต่อความถี่ในการทำ Spanning tree เนื่องจากค่าที่วัดได้ไม่แปรผันหรือแปรผันกับความถี่ในการทำ Spanning tree โดยค่าที่ได้เพิ่มขึ้นตามระยะห่างระหว่างวัตถุกับ BS



รูปที่ 9 แสดงภาพของการทดลอง



Object

รูปที่ 10 หน้าจอของระบบ

รูปที่ 10 แสดงตัวอย่างของหน้าจอซึ่งเชื่อมต่อกับ Root node ซึ่งนำค่าที่วัดได้จาก BS มาทำการคำนวณหาตำแหน่งของวัตถุ

สรุป

จากการทดลองได้ความแม่นยำของการระบุตำแหน่งของวัตถุที่ประมาณ 8 เซนติเมตรตามแกน X และ 10 เซนติเมตรตามแกน Y และจากการจำลองโดยการคำนวณ (Simulation) จะได้ความแม่นยำของการระบุตำแหน่งของวัตถุที่ประมาณ 3 เซนติเมตรตามแกน X และ 4 เซนติเมตรตามแกน Y ซึ่งผลที่ได้ก็นับว่าดีโดยพิจารณาถึงตัวแปรที่สำคัญได้แก่อุณหภูมิตั้งแต่ 25-30 องศาเซลเซียส

เอกสารอ้างอิง

- [1] Y.T. Chan, K.C. Ho, 1994. "A Simple and Efficient Estimator for Hyperbolic Location"
- [2] Mohamed G. Gouda, Young-ri Choi, "The Logical Grid Routing Protocol in Sensor Networks"
- [3] David Gay, Phil Levis, David Culler, 2005. "Software Design Patterns for TinyOS"
- [4] Murugavel R, 2001. "Ultrasonic Distance Measurement With the MSP430"
- [5] Roy Want, Andy Hopper, Veronica Falcao, Jonathan Gibbons, 1992. "The Active Badge Location System"

- [6] Texas Instrument, 2006. "MSP430X1XX Family User's Guide"
- [7] Texas Instrument, 2006. "Single Chip Very Low Power RF Transceiver"
- [8] J.Hill, R.Szewczyk, A.Woo, S.Hollar, D.Culler, K.Pister, "System Architecture Directions for Networked Sensors"
- [9] Jeffrey Hightower , Roy Want, 2000. "SpotON : An Indoor 3D Location Sensing Technology Based on RF Signal Strength"
- [10] Cambridge University, "THE ACTIVE BADGE SYSTEM"
- [11] David Gay, Philip Levis, 2004. "The *nesC* Language: A Holistic Approach to Networked Embedded Systems"
- [12] Vlado H, Joseph P, Jan-Henrich H, Cory S, Adam W, 2005. "Flexible Hardware Abstraction for Wireless Sensor Networks"
- [13] Ali H. Sayed, Nabil R. Yousef, **1999**. "Wireless Location"
- [14] Jaein Jeong, Sukun Kim, "CS262A Advanced Topics in Computer Systems DOT3 Radio Stack"