

อุปกรณ์แจ้งเตือนอัตโนมัติผ่าน Line , SMS และ E-Mail

Automatically Warning Device via Line SMS and E-Mail

ณพวุฒิ โพธิ์หอม¹
ชัยพร เขมะภาคะพันธ์²

บทคัดย่อ

การวิจัยครั้งนี้ มีวัตถุประสงค์เพื่อพัฒนาอุปกรณ์แจ้งเตือนผ่าน Line SMS และ E-Mail โดยใช้ RaspberryPi ร่วมกับอุปกรณ์เซ็นเซอร์ต่าง ๆ พัฒนาด้วยภาษา Python และ Visual Basic.NET เมื่อระบบที่พัฒนาขึ้นสามารถตรวจจับสัญญาณต่าง ๆ ได้จากเซ็นเซอร์แล้วจะทำการส่งข้อความแจ้งเตือนไปยังเซิร์ฟเวอร์ซึ่งมีระบบประมวลผลเพื่อวิเคราะห์และแยกแยะประเภทการแจ้งเตือนก่อนตัดสินใจแจ้งเตือนผ่านช่องทางต่าง ๆ ตามกลุ่มผู้รับที่กำหนดไว้ก่อนหน้าได้

จากการทดลองจะพบว่าระบบสามารถส่งข้อมูลแจ้งเตือนไปยังกลุ่มผู้รับปลายทางได้ครบถ้วนทั้ง 3 ช่องทางตามที่ได้ตั้งค่าไว้ โดยพบว่าการส่งผ่าน Line มีความรวดเร็วมากที่สุด อย่างไรก็ตาม SMS สามารถเข้าถึงผู้รับที่ไม่ได้เชื่อมต่ออินเทอร์เน็ตได้ ในขณะที่การแจ้งเตือนผ่าน E-Mail มีการแสดงรายละเอียดข้อมูลได้มากกว่าวิธีอื่น ๆ

Abstract

This research aimed to develop automatically warning device via Line, SMS and E-Mail using RaspberryPi working with sensors which is coded by Python and Visual Basic.NET. When developed system can detect a signal from sensors, a warning message is directly sent to a computer server which runs a computer program, Analyzer, to analyze the warning message and decide to send the message via Line and/or SMS and/or E-Mail to a group of user which is already defined.

From testing results, the system can send warning messages via Line, SMS and E-Mail. All users can completely receive all messages. Moreover, warning via Line is fastest while SMS users can receive the messages without connection to Internet. However, warning via E-Mail can provide more detailed information.

¹ นักศึกษาหลักสูตรวิศวกรรมคอมพิวเตอร์และโทรคมนาคม มหาวิทยาลัยบูรพา

² อาจารย์ที่ปรึกษาวิทยานิพนธ์

บทนำ

เหตุผลที่ทางผู้จัดทำได้คิดการพัฒนาอุปกรณ์ IOT มาเพื่อแก้ไขกับปัญหากับงานปัจจุบันนั้น เพราะว่าได้สังเกตเห็นถึงระบบสิ่งอำนวยความสะดวกพื้นฐานที่ใช้งานอยู่นั้น ไม่สามารถตอบโจทยกับงานปัจจุบันได้ เช่น ระบบวัดอุณหภูมิในห้อง Server ที่มีตัวเลขดิจิทัลโชว์อยู่กลางห้อง , ระบบตรวจจับควันที่จะมี Alarm เสียงขึ้น ฯลฯ ซึ่งระบบดังกล่าวต้องใช้คนเดินเข้าไปตรวจสอบดู และถ้าไม่มีเจ้าหน้าที่ on-site อยู่จะไม่มีใครที่จะรับรู้ Alarm ได้เลย และที่สำคัญคือมีข้อจำกัดด้านสถานที่ติดตั้งไม่สามารถ integrate ระบบ Sensor ต่างๆ กับ ระบบเดิมที่ติดตั้งอยู่แล้วได้เลย ทำให้ผู้จัดทำจึงคิดโจทยที่ว่า “จะหาอุปกรณ์ที่ตรวจจับ Sensor ต่างๆ ที่ติดตั้งได้ง่าย ขนาดเล็ก , ประหยัดพลังงาน และพัฒนาต่อยอดได้ง่าย” จึงเป็นที่มาที่ทางผู้จัดทำจึงได้เลือกหยิบอุปกรณ์ IOT มาใช้ในการพัฒนานี้ร่วมกับการทำ Real-Time Analytics

วัตถุประสงค์ของการศึกษา

1. เพื่อนำอุปกรณ์ IOT ติดตั้งไว้ในจุดที่ต้องการและส่งข้อมูลมายังอุปกรณ์ Analyzer ได้
2. เพื่อวิเคราะห์ Alarm ที่ส่งจากอุปกรณ์ IOT ได้ว่าต้องส่งไปที่ผู้รับผิดชอบคนไหน
3. เพื่อส่ง Alarm ออกไปยัง Line Group ที่กำหนดไว้ได้
4. เพื่อส่ง Alarm ออกไปยัง SMS ที่ผู้รับผิดชอบทำการลงทะเบียนเบอร์โทรศัพท์ไว้ได้
5. เพื่อส่ง Alarm ออกไปยัง E-Mail ที่ผู้รับผิดชอบทำการลงทะเบียนเบอร์โทรศัพท์ไว้ได้
6. เก็บรวบรวมประวัติการเกิด Alarm ย้อนหลังไว้ได้

วัสดุ อุปกรณ์ เครื่องมือหรือโปรแกรมที่ใช้ในการพัฒนา

1. เครื่องคอมพิวเตอร์ พร้อมการเชื่อมต่อเข้าสู่ระบบอินเทอร์เน็ต
2. โปรแกรม Visual Studio เพื่อใช้ในการพัฒนาโปรแกรม Analyzer
3. อุปกรณ์ Raspberry Pi II เพื่อใช้เป็นอุปกรณ์ IOT ในการ Detect Sensor
4. Sensor DHT11 ใช้เชื่อมต่อเข้ากับ Raspberry Pi เพื่อตรวจจับค่าความชื้นและอุณหภูมิ
5. Sensor SW-420 ใช้เชื่อมต่อเข้ากับ Raspberry Pi เพื่อตรวจจับการสั่นสะเทือน
6. Sensor MQ-7 ใช้เชื่อมต่อเข้ากับ Raspberry Pi เพื่อตรวจจับค่า CO
7. Sensor STM-32 ใช้เชื่อมต่อเข้ากับ Raspberry Pi เพื่อตรวจจับเปลวไฟ

ขั้นตอนการดำเนินการ

1. ศึกษาและค้นคว้าข้อมูลจากปัญหาที่เจอ ซึ่งนำมาซึ่งวิธีการนำอุปกรณ์ IOT เข้ามาช่วยแก้ไข ปัญหา และศึกษาช่องทางเพื่อนำมาประยุกต์ใช้ให้เหมาะสมกับการใช้ชีวิตประจำวัน
2. ศึกษารายละเอียดเกี่ยวกับ API ต่าง ๆ ที่ต้องนำมาใช้งาน เช่น LINE Notify API , SMS API , SMTP API และ Protocol ที่นำมาใช้ต้องมีความ Simple เพื่อง่ายต่อการพัฒนาต่อยอดในอนาคต
3. ศึกษาภาษาที่มีความยืดหยุ่นที่สามารถใช้เขียนลงอุปกรณ์ IOT เพื่อให้ตรงกับความต้องการ ใช้งานของระบบ ซึ่งในโครงการนี้ใช้ภาษา Python ในการพัฒนาร่วมกับระบบปฏิบัติการ Rasbian Detect ค่า Sensor จาก Port GPIO
4. ศึกษาภาษาที่สามารถเชื่อมต่อกับ API ต่าง ๆ และสามารถทำงานได้บนอุปกรณ์จำพวก Server และ Cloud ต่าง ๆ ได้ง่าย ซึ่งในโครงการนี้จะใช้ภาษา .NET ในการพัฒนา
5. จัดทำโครงร่างโครงการระบบ Alarm Monitoring with IOT เสนอผ่านอาจารย์ที่ปรึกษา
6. นำเสนอรายงานความก้าวหน้าเป็นระยะ
7. ดำเนินการทดสอบนำระบบมา Integrate ทำงานร่วมกันทั้งระบบ Sensor, Analyzer และ API ต่าง ๆ สามารถทำงานได้อย่างถูกต้อง
8. ทดสอบ Simulation ผลงานโดยการเชื่อมต่อ IOT เข้ากับระบบ Network ที่สามารถเชื่อมต่อเข้ากับระบบ Analyzer ได้ พร้อมทั้ง Allow Port ที่ต้องการใช้งาน และ สามารถเชื่อมต่ออินเทอร์เน็ตได้
9. นำผลที่ได้เสนออาจารย์ที่ปรึกษา

ระยะเวลาที่ใช้ในการดำเนินการ

1. ศึกษาหาความเป็นไปได้ของโครงการ 2 เดือน
2. พัฒนาระบบ IOT 1 เดือน
3. พัฒนาระบบ Analyzer 2 เดือน
4. ดำเนินการวัดผลและสรุปผลพร้อมจัดทำรายงาน 1 เดือน

อุปกรณ์ IOT จะทำหน้าที่อยู่ 2 หน้าที่คือ

1. คอยรับค่าจาก Sensor ต่าง ๆ เช่น CO₂ Sensor, Flame Sensor, Temp Sensor, Humidity Sensor, Vibration Sensor และ Monitoring Sensor
2. ประมวลผลค่าที่ได้รับและส่งค่าออกไปยังอุปกรณ์ Analyzer เพื่อที่อุปกรณ์ Analyzer จะได้ดำเนินการวิเคราะห์ Alarm และส่งต่อไปยังผู้ใช้

โดย Alarm ทั้งหมดจะถูกส่งโดย Protocol UDP ไปยังอุปกรณ์ Analyzer ผ่านระบบ Network ตามที่ได้ เช็ตไว้ 3G, 4G, LAN หรือ Wifi และ Analyzer จะวิเคราะห์ Alarm ว่า Alarm ดังต่อไปนี้จะดำเนินการส่งให้ ผู้รับคนไหนต่อไปตามช่องทางต่าง ๆ ที่ได้เตรียมไว้คือ LINE Notify , E-mail และ SMS

การวิเคราะห์ข้อมูล

ในการวิเคราะห์ข้อมูลผู้ศึกษาได้แบ่งการวิเคราะห์ข้อมูลแบ่ง 2 ช่วงด้วยกัน

1. ข้อมูลความถูกต้องและการทำงานของอุปกรณ์ IOT
2. ข้อมูลความถูกต้องและการทำงานของระบบ Analyzer โดยต้องทำงานได้ถูกต้องทุก Function และ ทุก Alarm Filter
3. เสถียรภาพของระบบที่สามารถทำงานได้ต่อเนื่องกับทุก ๆ API

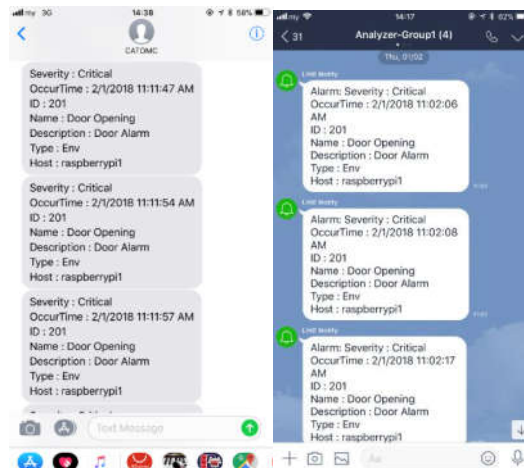
ผลการพัฒนาระบบ IOT

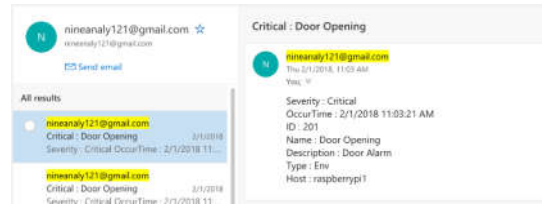
การพัฒนา ระบบ IOT ซึ่งทางผู้จัดทำได้ใช้อุปกรณ์ Raspberry Pi II พร้อมกับระบบปฏิบัติการ Raspbian นั้น ได้ดำเนินการตามขั้นตอนการดำเนินงานที่เสนอในบทที่ 3 แล้ว

โดยผู้ใช้งานจะต้องดำเนินการ Configuration Hostname , IP และ Port ของอุปกรณ์ Analyzer ปลายทางเพื่อให้ระบบ IOT สามารถส่งข้อมูลไปหา Analyzer ได้อย่างถูกต้อง

การทำงานของระบบ IOT เมื่อผู้ใช้งานเปิดอุปกรณ์ IOT และได้ตั้งค่าอุปกรณ์ทั้งหมดแล้ว Process ทั้งหมดจะ Active ขึ้นมาทำงานโดยอัตโนมัติจาก Crontab Script ที่ตั้งไว้เมื่อทำการ Reboot ระบบ แต่ถ้าระบบทำงานผิดพลาด ผู้ใช้งานสามารถเข้าไปดำเนินการ Stop และ Start จาก Process Script ที่ได้สร้างไว้ (launcher_start.sh , launcher_stop.sh)

ระบบสามารถส่งข้อมูลไปยังปลายทางได้อย่างถูกต้อง





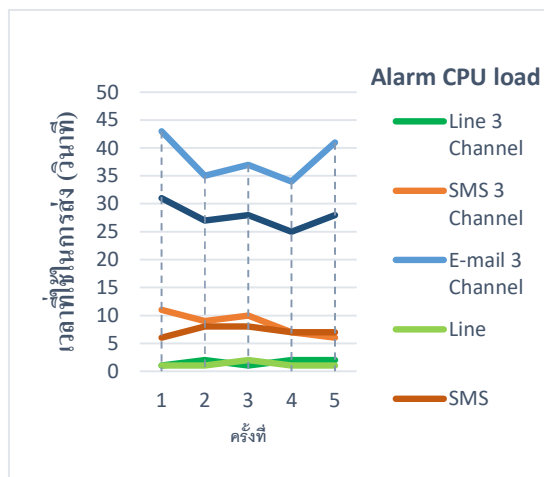
ภาพผลการส่งข้อมูล Alarm ไปยังปลายทาง

Sensor	Local	DHT11	SW-420	MQ-7	STM-32
Alarm					
Line	✓	✓	✓	✓	✓
SMS	✓	✓	✓	✓	✓
E-Mail	✓	✓	✓	✓	✓

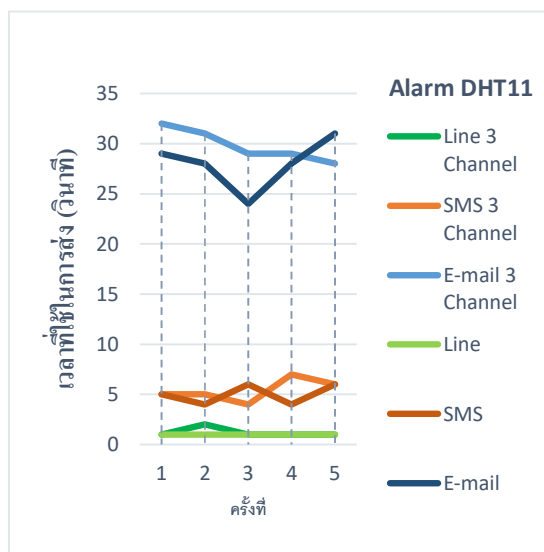
ตารางแสดงผลการส่งข้อมูลออกไปยังปลายทางของแต่ละ Sensor

จากผลการทดลองเบื้องต้นพบว่าระบบ Alarm Monitoring สามารถรับค่าจาก Sensor ต่าง ๆ (Local Environment, DHT11, SW-420, MQ-7 และ STM-32) และสามารถส่งข้อมูลไปยังปลายทางตามที่ตั้งค่าไว้ได้ และจากผลการทดลองเบื้องต้น ทางผู้จัดทำจึงจำแนกการทดลองไปที่แต่ละตัว Sensor โดยทำการทดลองดังนี้

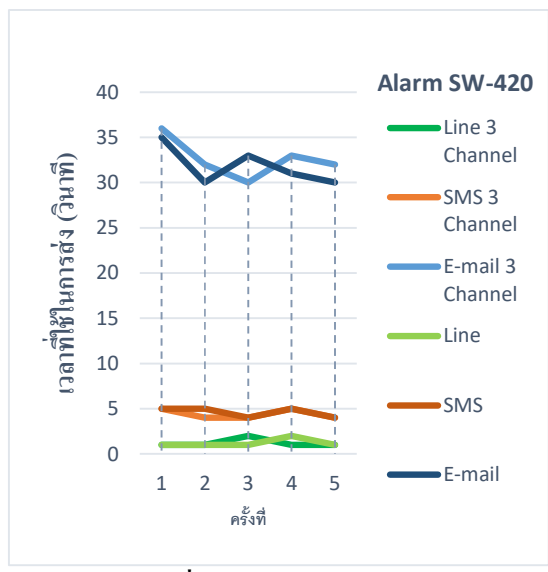
1. ทดลองส่ง Alarm CPU Load ของอุปกรณ์ Raspberry Pi ไปยังปลายทางทั้ง 3 ช่องพร้อมกันและจำแนกที่ละช่องทาง
2. ทดลองส่ง Alarm จาก Sensor DHT11 ของอุปกรณ์ Raspberry Pi ไปยังปลายทางทั้ง 3 ช่องพร้อมกันและจำแนกที่ละช่องทาง
3. ทดลองส่ง Alarm จาก Sensor SW-420 ของอุปกรณ์ Raspberry Pi ไปยังปลายทางทั้ง 3 ช่องพร้อมกันและจำแนกที่ละช่องทาง
4. ทดลองส่ง Alarm จาก Sensor MQ-7 ของอุปกรณ์ Raspberry Pi ไปยังปลายทางทั้ง 3 ช่องพร้อมกันและจำแนกที่ละช่องทาง
5. ทดลองส่ง Alarm จาก Sensor STM-32 ของอุปกรณ์ Raspberry Pi ไปยังปลายทางทั้ง 3 ช่องพร้อมกันและจำแนกที่ละช่องทาง



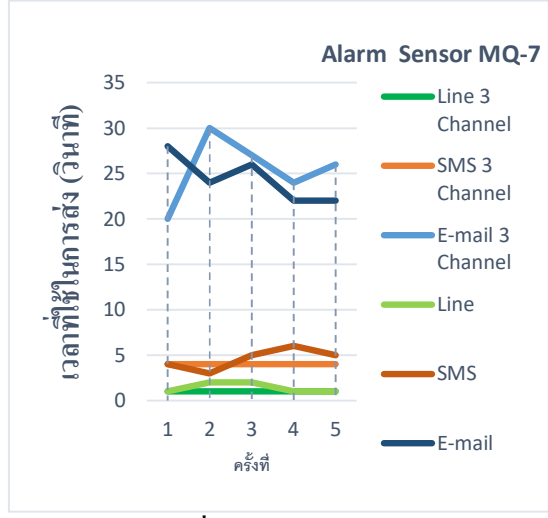
กราฟแสดงเวลาที่ใช้ในการส่งของ Alarm CPU



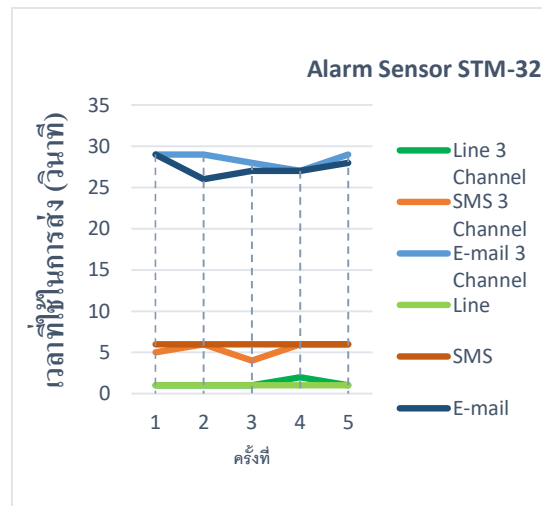
กราฟแสดงเวลาที่ใช้ในการส่งของ Alarm DHT11



กราฟแสดงเวลาที่ใช้ในการส่งของ Alarm SW-420



กราฟแสดงเวลาที่ใช้ในการส่งของ Alarm MQ-7



กราฟแสดงเวลาที่ใช้ในการส่งของ Alarm STM-32

สรุปผลการทดลอง

จะเห็นได้ว่าทั้ง 5 การทดลองข้างต้นช่องทางในการส่งข้อมูล Alarm ทั้งหมดช่อง Line เป็นช่องทางที่ถึงปลายทางเร็วที่สุดและตามมาด้วย SMS และ E-mail แต่ถ้าตรวจสอบดูค่านัยสำคัญแล้วต่างกันอยู่แค่ประมาณ 4-6 วินาที และ 25-40 วินาทีตามลำดับ ซึ่งไม่น่าจะมีผลกับการที่เมื่อปลายทางได้รับ Alarm แล้วลงมือดำเนินการตามหน้าที่ที่รับผิดชอบ

แต่ยังมีอีก 1 ประเด็นที่น่าสนใจคือ ความรวดเร็วในการตรวจจับของ Sensor ต่าง ซึ่งจากการทดลองพบว่า Sensor STM-32 (Flame Sensor) เป็น Sensor ที่มีความรวดเร็วในการตรวจจับสูงมากเพราะเมื่อมีเปลวไฟเกิดขึ้นตัว Sensor STM-32 จะมี Output ส่งไปยัง Raspberry Pi ทันทีในเวลาไม่ถึง 1 - 2 วินาที ต่างกับ Sensor อื่นซึ่งจะมี Delay ระดับหนึ่งกว่าจะส่ง Output ออกไปยัง Raspberry Pi โดยเรียงลำดับตามนี้ STM-32 > SW-420 > DHT11 > MQ-7

อภิปรายผล

จากการทำโครงการเรื่อง Alarm Monitoring with IOT ผลการศึกษาและการทดลองอยู่ในเกณฑ์ที่ดี และมีความสำเร็จและยืดหยุ่นพอที่จะนำไปต่อยอดเพื่อใช้แก้ปัญหาตามผู้จัดทำต้องการได้ และเมื่อพิจารณารายละเอียดหลายๆ ด้านแล้วคิดว่า ในอนาคตนั้นอุปกรณ์ IOT จะเข้ามามีบทบาทกับชีวิตประจำวันมากขึ้นและสามารถช่วยอำนวยความสะดวกต่างๆ และสามารถทำงานแทนบุคคลได้เลยทีเดียว และโครงการนี้ได้รับคำแนะนำจากอาจารย์และผู้มีประสบการณ์หลายๆ ท่านว่าจะเป็นการที่เป็นประโยชน์ สามารถนำไปประยุกต์ใช้ได้ทันที พร้อมทั้งเป็นอุปกรณ์ IOT ที่มีคนนิยมใช้มากจึงง่ายต่อการพัฒนาเป็นอย่างมาก

ข้อเสนอแนะ

1. ปัจจุบันผู้จัดทำได้ใช้อุปกรณ์ IOT (Raspberry Pi) ที่ยังต้องอาศัยการเชื่อมต่อกับ Wi-Fi หรือ RJ45 ทำให้ยังมีข้อจำกัดในการใช้งานในจุดที่ไม่มี Wi-Fi หรือ RJ45 ซึ่งสามารถพัฒนาต่อยอดไปใช้อุปกรณ์ 3G , 4G หรือ สัญญาณ LoRaWAN หรือ NB-IOT ในอนาคตได้
2. ในโครงการนี้ยังไม่ได้ดำเนินการออกแบบในส่วนของ Prototype ซึ่งสามารถพัฒนาใส่อุปกรณ์ Battery + 3G เพื่อสามารถทำงานได้ในเวลาที่ไม่มี Power Source จ่ายเข้ามาในระบบ
3. โครงการนี้ยังขาดการสื่อสารแบบ 2 ทาง โดยผู้ใช้งานสามารถสั่งการกลับไปยังอุปกรณ์ IOT ให้ปฏิบัติงานตามต้องการได้
4. ประเด็นเรื่อง Cyber Security ของอุปกรณ์ IOT, อุปกรณ์ Analyzer เพื่อให้ระบบเกิดความมั่นคงปลอดภัย ต่อการบุกรุกจากผู้ไม่ประสงค์ดีภายนอก

บรรณานุกรม

- อ.นพ มหิษานนท์, (2560). **Raspberry Pi PROJECTS. (พิมพ์ครั้งที่ 1).** นนทบุรี: ผู้แต่ง.
สถาบัน Home of Maker. การเรียนรู้ **Raspberry Pi** . สืบค้นจาก <http://www.homeofmaker.com/?p=891>
Lluís Casals ID , Bernat Mir ID , Rafael Vidal ID and Carles Gomez * ID. (2017). **Modeling the Energy. Performance of LoRaWAN.**
Chutchavan Suksutthi. การพัฒนาโปรแกรมบน **Raspberry Pi ด้วย Qt.** สืบค้นจาก
<http://www2.crma.ac.th/itd/Know/RBPI/index.asp>
Kaushik Pal ,(July 20, 2016). The Advantages of Real-Time Analytics for Enterprise. Presented by
Sponsor: Bloor Group.
<https://www.techopedia.com/2/31433/trends/big-data/advantages-of-real-time-analytics-for-enterprise>