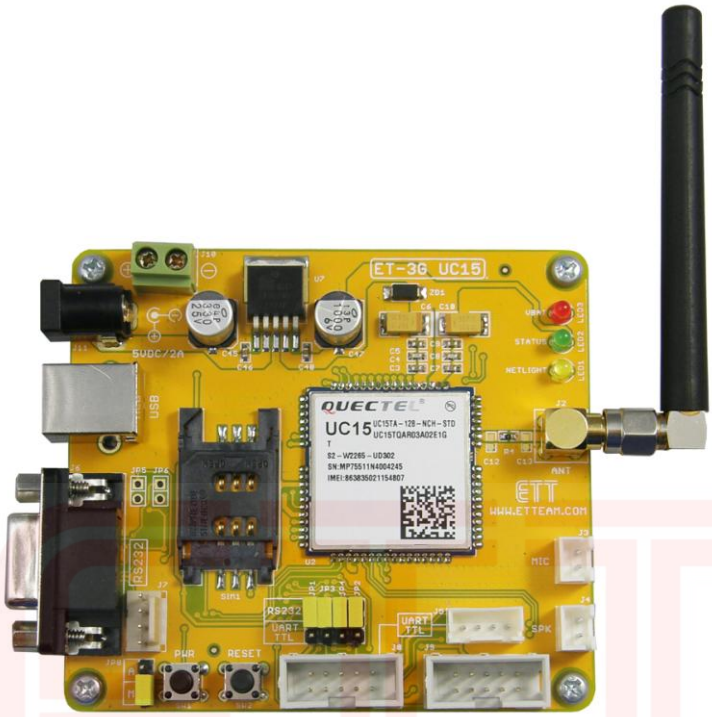
สารบัญ

1. คุณสมบัติของบอร์ด ET-3G UC15 3
2. ส่วนประกอบของบอร์ด ET-3G UC15 4
3. การสั่ง เปิด และ ปิด การทำงานของโมดูล UC15 8
4. การติดต่อสื่อสารกับโมดูล UC15 9
5. คุณสมบัติการทำงานของสัญญาณที่ควรรู้ 10
6. ตัวอย่างการใช้งาน AT Command เพื่อสั่งงานโมดูล UC15 11
7. การติดตั้งไดร์เวอร์ USB ของบอร์ด ET-3G UC15 12
8. การทดสอบการทำงานของบอร์ด ET-3G UC15 16
9. ตัวอย่างการใช้งานคำสั่งเกี่ยวกับ RS232 19
10. การ Setup และตรวจสอบค่า Configuration 21
11. การตรวจสอบข้อมูลต่างๆ ของโมดูล UC15 22
12. การโทรออก การรับสาย และ การยกเลิกการโทร 24
13. การตรวจสอบยอดเงินคงเหลือโดยใช้ USSD 25
14. การรับข้อความ SMS 26
15. การส่งข้อความ SMS ภาษาอังกฤษ 27
16. รหัสข้อความ SMS ภาษาไทย 28
17. หลักการถอดรหัสตัวอักษร Unicode 30
18. การส่งข้อความ SMS ภาษาไทย 33
19. การส่งข้อความ MMS (Multimedia Messaging Service) 35
20. การเชื่อมต่อบอร์ด ET-3G UC15 กับ บอร์ดไมโครคอนโทรลเลอร์ 39
21. ขนาดและวงจรของบอร์ด ET-3G UC15 41

www.etteam.com

ET-3G UC15

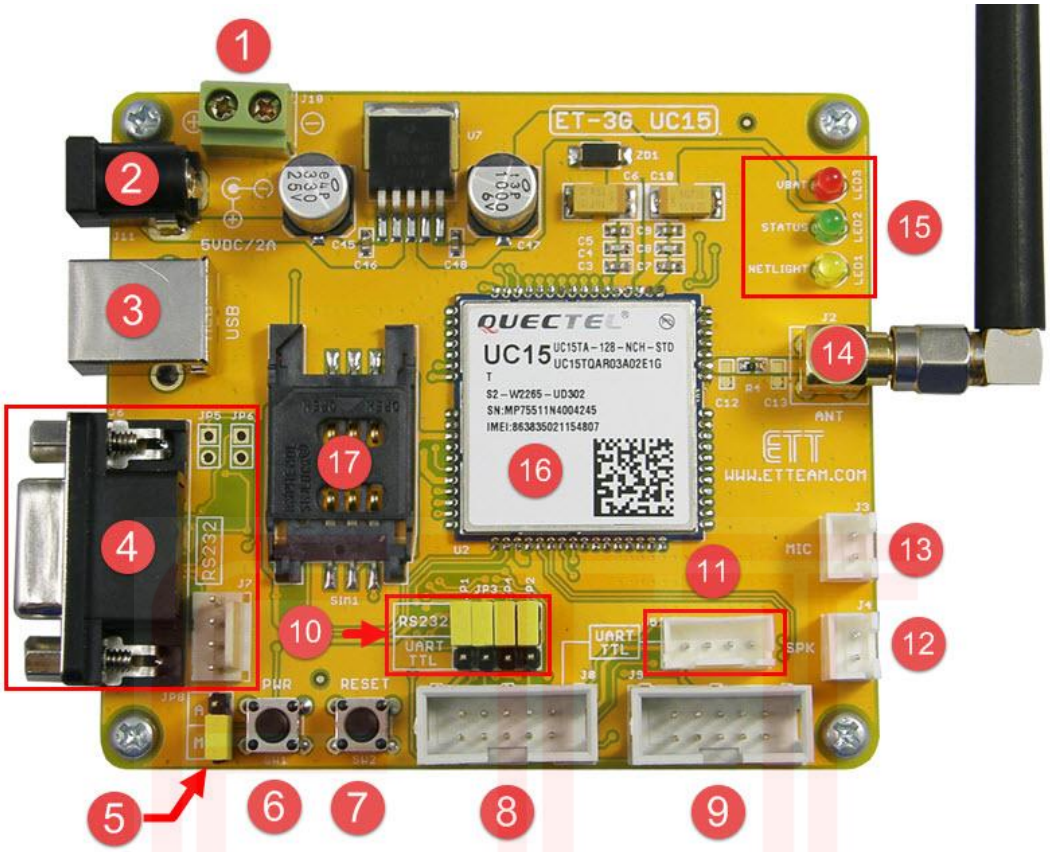


ET-3G UC15 เป็นชุดเรียนรู้และพัฒนาระบบการสื่อสารทางโทรศัพท์มือถือ โดยใช้โมดูล UMTS/HSDPA รุ่น UC15-T ของบริษัท Quectel เป็นอุปกรณ์หลัก ซึ่ง UC15 เป็นโมดูลสื่อสารระบบ UMTS/HSPA รองรับระบบสื่อสาร GSM ความถี่ 850/900/1800/1900MHz และ UMTS(3G) ความถี่ 850/2100MHz โดยการสั่งงานของโมดูล UC15 จะสั่งงานผ่านทางพอร์ตสื่อสารอนุกรม RS232 และพอร์ต USB ด้วยชุดคำสั่ง AT Command สามารถประยุกต์ใช้งานได้มากมายหลายรูปแบบ ไม่ว่าจะเป็นการรับส่งสัญญาณแบบ Voice, SMS, Data และยังรวมถึงการสื่อสารด้วย Protocol TCP/IP ด้วย ซึ่งตามปรกติแล้ว ถึงแม้ว่าโมดูล UC15 จะมีวงจร และ Firmware บรรจุไว้ภายในตัวเป็นที่เรียบร้อยแล้วก็ยังไม่สามารถนำไปใช้งานได้โดยตรงทันที เนื่องจากในการใช้งานจริงๆนั้น ผู้ใช้งานเองจำเป็นต้องออกแบบวงจรรอบนอกที่จำเป็นมาเชื่อมต่อกับขาสัญญาณของตัวโมดูลอีกในบางส่วน ไม่ว่าจะเป็นวงจรภาค Power Supply, วงจรเชื่อมต่อกับ SIM Card รวมไปถึงวงจร Line Driver ของ RS232 เป็นต้น ดังนั้นทางทีมงาน อีทีที จึงได้จัดสร้างบอร์ดสำหรับเป็นตัวกลางในการเชื่อมต่อระหว่างโมดูล UC15 กับอุปกรณ์ภายนอกเพื่อให้ผู้ใช้งานสามารถนำโมดูล UC15 ไปทำการทดลองและศึกษาเรียนรู้การสั่งงานต่างๆ ได้โดยสะดวก ก่อนที่จะนำเอาโมดูลตัวนี้ไปออกแบบดัดแปลงและประยุกต์ใช้งานในด้านต่างๆ ได้ต่อไปในอนาคต ซึ่งถึงแม้ว่าวงจรการเชื่อมต่อทั้งหมดที่ทางอีทีที ได้จัดทำขึ้นมานี้จะยังไม่สามารถรองรับการใช้งานทรัพยากรต่างๆ ที่มีอยู่ภายในโมดูลได้ครบถ้วนทั้งหมดก็ตามที แต่ในส่วนของการใช้งานโมดูลในส่วนที่เป็นความสามารถหลักๆ ที่จำเป็นนั้นมีไว้รองรับอย่างครบถ้วนเพียงพอแล้ว

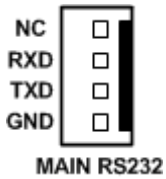
1. คุณสมบัติของบอร์ด ET-3G UC15

- ใช้โมดูล UMTS/HSDPA รุ่น UC15-T ของบริษัท Quectel
- รองรับความถี่ GSM 850/900/1800/1900MHz และ UMTS(3G) 850/2100MHz
- ความเร็วในการรับส่งข้อมูลสูงสุด HSPA Max 3.6 Mbps (DL) Max 384 Kbps (UL)
- รองรับโปรโตคอลเชื่อมต่อ PPP/ TCP/ UDP/ FTP/ HTTP/ FILE/ MMS/ SMTP/SSL/ PING
- รองรับคำสั่ง AT COMMAND (Compliant with 3GPP TS 27.007, 27.005 and Quectel enhanced AT)
- รองรับ SIM Card แบบ FULL SIZE (1.8V และ 3.3V) พร้อมวงจร ESD ป้องกัน SIM เสียหาย
- มีสวิตช์แบบ Push-Button สำหรับใช้สั่ง เปิด-ปิด การทำงานของโมดูลภายในบอร์ด
- มีสวิตช์แบบ Push-Button สำหรับใช้สั่ง RESET การทำงานของโมดูลภายในบอร์ด
- มีวงจร Regulate จำนวน 2 ชุด โดยสามารถใช้กับแหล่งจ่ายภายนอก 5 VDC กระแส 2 A ขึ้นไป ซึ่งสามารถจ่ายกระแสให้กับโมดูล UC15-T และอุปกรณ์เชื่อมต่อต่างๆ ได้อย่างเพียงพอ
    - ใช้วงจร Regulate ขนาด 3.88V / 3A สำหรับจ่ายให้กับโมดูล UC15-T ได้อย่างเพียงพอ สามารถใช้กับ SIM ของระบบต่างๆ ได้อย่างไม่เกิดปัญหา
    - ใช้วงจร Regulate ขนาด 3.3V / 200mA สำหรับจ่ายให้กับวงจร RS232
- มีวงจร Line Driver สำหรับแปลงระดับสัญญาณโลจิกจากโมดูล UC15-T ให้เป็น RS232 (9600 bps-921600 bps) สำหรับพอร์ตที่ใช้ในการสื่อสารและสั่งงานโมดูล โดยมีทั้งขั้ว DB9 ตัวเมีย มาตรฐาน และ แบบ 4 PINS (มาตรฐานอีทีที)
- มีวงจรแปลงระดับสัญญาณโลจิก TTL ระดับแรงดัน 3.3V-5V ทำให้สามารถเชื่อมต่อกับไมโครคอนโทรลเลอร์ได้โดยตรงโดยไม่ต้องผ่านวงจร Line Driver RS232
- มีพอร์ตเชื่อมต่อ USB 2.0 (TYPE B) สำหรับสื่อสารกับบอร์ด โดยรองการใช้งานบน Windows XP/Vista/7/8
- มี LED แสดงสถานะ สำหรับแสดงสถานะของแหล่งจ่ายไฟ สถานะพร้อมทำงานของโมดูล สถานะในการเชื่อมต่อกับ Network และ สถานะ Power-ON/Power-OFF ของโมดูล
- มีขั้วสำหรับเชื่อมต่อกับ ลำโพงและไมโครโฟน
- ขนาดบอร์ด 90 x 80 mm.

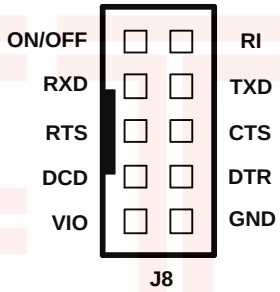
2. ส่วนประกอบของบอร์ด ET-3G UC15



- **หมายเลข** 1 ขั้วต่อไฟเลี้ยงเข้าบอร์ด แบบขั้วต่อขันน็อต โดยใช้แรงดันไฟเลี้ยง 5 VDC กระแสอย่างน้อย 2 A
- **หมายเลข** 2 ขั้วต่อไฟเลี้ยงเข้าบอร์ดแบบ DC JACK ซึ่งขั้วด้านนอกเป็นไฟบวก ด้านในเป็นลบ โดยใช้แรงดันไฟเลี้ยง 5 VDC กระแสอย่างน้อย 2 A
- **หมายเลข** 3 พอร์ตเชื่อมต่อ USB 2.0 (TYPE B) สำหรับสื่อสารกับบอร์ด
- **หมายเลข** 4 RS232 เป็นขั้วต่อ RS232 มาตรฐาน แบบ DB9 ตัวเมีย (รองรับ Hardware Flow Control) สำหรับใช้เชื่อมต่อกับสัญญาณ RS232 แบบ DB9 ตัวผู้ จากคอมพิวเตอร์ PC หรืออุปกรณ์ภายนอกอื่นๆ โดยใช้สาย 9 Pin แบบต่อตรง และขั้วต่อสัญญาณ RS232 แบบ 4 PINS (มาตรฐานอีทีที) ซึ่งพอร์ตนี้จะเป็นพอร์ตในการสั่งงานหลัก โดยการจัดตำแหน่งขาสัญญาณแบบ 4 PINS แสดงดังรูป

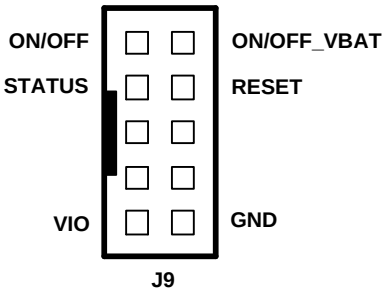


- **หมายเลข** 5 จั้มเปอร์เลือกเปิดการทำงานของโมดูล UC15 โดยถ้าเลือกไปที่ตำแหน่ง M จะเป็นการเลือกเปิดการทำงานของโมดูลโดยการกดสวิตช์ PWR หรือควบคุมจากภายนอก แต่ถ้าเลือกไปที่ตำแหน่ง A โมดูลจะทำงานทันทีเมื่อจ่ายไฟเลี้ยงเข้าบอร์ด
- **หมายเลข** 6 สวิตช์ PWR สำหรับเปิดปิดการทำงานของโมดูล โดยต้องกดปุ่มเป็นเวลาอย่างน้อย 100ms เพื่อเปิดการทำงานของโมดูล และกดปุ่มอีกอย่างน้อย 600ms เพื่อปิดการทำงานของโมดูล
- **หมายเลข** 7 สวิตช์ RESET สำหรับใช้รีเซ็ตการทำงานของตัวโมดูล
- **หมายเลข** 8 เป็นขั้วต่อสัญญาณระดับ TTL 3.3-5 V สำหรับเชื่อมต่อกับไมโครคอนโทรลเลอร์ได้โดยตรงโดยไม่ต้องผ่านวงจร Line Driver RS232 โดยการจัดตำแหน่งขาสัญญาณแสดงดังรูป



ชื่อขาสัญญาณ	ทิศทาง	รายละเอียด
ON/OFF	INPUT	ใช้ควบคุมการเปิดปิดโมดูล UC15 โดยจะทำงานที่ลอจิก 1
RI	OUTPUT	Ring Indicator
RXD	INPUT	Receive data
TXD	OUTPUT	Transmit data
RTS	INPUT	Request to send
CTS	OUTPUT	Clear to send
DCD	OUTPUT	Data carrier detect
DTR	INPUT	Data terminal ready
VIO	INPUT	ใช้รับแรงดันจากวงจรที่ทำการเชื่อมต่อด้วย (3.3V-5V) เพื่ออ้างอิงกับวงจร Level shifter ทำให้สามารถเชื่อมต่อกับวงจร ที่มีระดับลอจิกแตกต่างกันได้
GND	-	กราวด์

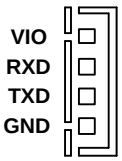
- **หมายเลข** 9 เป็นขั้วต่อสัญญาณระดับ TTL 3.3-5 V สำหรับเชื่อมต่อกับไมโครคอนโทรลเลอร์ เพื่อใช้ควบคุมและตรวจสอบสภาวะต่างของบอร์ด โดยการจัดตำแหน่งขาสัญญาณแสดงดังรูป



ชื่อขาสัญญาณ	ทิศทาง	รายละเอียด
ON/OFF	INPUT	ใช้ควบคุมการเปิดปิดโมดูล UC15 โดยจะทำงานที่ลอจิก 1
ON/OFF_VBAT	INPUT	ใช้ปิดการจ่ายไฟเลี้ยงของบอร์ด โดยจะทำงานที่ลอจิก 1
STATUS	OUTPUT	ใช้บอกสถานะว่าโมดูล UC15 เปิดการทำงานอยู่ ถ้าเป็นลอจิก 0 แสดงว่าโมดูลเปิดอยู่ ถ้าเป็นลอจิก 1 แสดงว่าแสดงว่าโมดูลปิดอยู่หรืออยู่สภาวะ Power Down Mode
RESET	INPUT	ใช้สำหรับรีเซ็ตการทำงานของโมดูล UC15 โดยจะทำงานที่ลอจิก 1
VIO	INPUT	ใช้รับแรงดันจากวงจรที่ทำการเชื่อมต่อด้วย (3.3V-5V) เพื่ออ้างอิงกับวงจร Level shifter ทำให้สามารถเชื่อมต่อกับวงจร ที่มีระดับลอจิกแตกต่างกันได้
GND	-	กราวด์

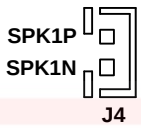
www.etteam.com

- **หมายเลข** 10 เป็นจั้มเปอร์เลือกว่าจะต่อขาสัญญาณ UART ของโมดูลผ่านวงจร Line Driver RS232 หรือไม่ ถ้าผู้ใช้ต้องการเชื่อมต่อผ่านขั้ว RS232 ก็ให้เลือกจั้มเปอร์ JP1-JP4 ไปที่ตำแหน่ง RS232 แต่ถ้าต้องการเชื่อมต่อทางขั้ว TTL (J5 , J8) ก็ให้เลือกจั้มเปอร์ JP1-JP4 ไปที่ตำแหน่ง UART TTL
- **หมายเลข** 11 เป็นขั้วต่อสัญญาณระดับ TTL 3.3-5 V สำหรับเชื่อมต่อกับไมโครคอนโทรลเลอร์ได้โดยตรงโดยไม่ต้องผ่านวงจร Line Driver RS232 โดยการจัดตำแหน่งขาสัญญาณเหลือเพียง 4 เส้น ซึ่งแสดงดังรูป



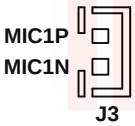
ชื่อขาสัญญาณ	ทิศทาง	รายละเอียด
VIO	INPUT	ใช้รับแรงดันจากวงจรที่ทำการเชื่อมต่อด้วย (3.3V-5V) เพื่ออ้างอิงกับวงจร Level shifter ทำให้สามารถเชื่อมต่อกับวงจร ที่มีระดับลอจิกแตกต่างกันได้
RXD	INPUT	Receive data
TXD	OUTPUT	Transmit data
GND	-	กราวด์

- **หมายเลข** 12 ขั้วต่อสำหรับเชื่อมต่อลำโพง (32 Ω) โดยมีขาสัญญาณดังรูป



  - SPK1P ขั้วต่อลำโพงขั้วบวก
  - SPK1N ขั้วต่อลำโพงขั้วลบ

- **หมายเลข** 13 ขั้วต่อสำหรับเชื่อมต่อคอนเดนเซอร์ไมโครโฟน โดยมีขาสัญญาณดังรูป



  - MIC1P ขั้วต่อไมโครโฟนขั้วบวก
  - MIC1N ขั้วต่อไมโครโฟนขั้วลบ

- **หมายเลข** 14 ขั้วต่อแบบ SMA สำหรับเชื่อมต่อกับเสาอากาศหลักของระบบ GSM 850/900/1800/1900MHz และ UMTS(3G) 850/2100MHz

- **หมายเลข** 15 LED แสดงสถานการณ์ทำงานของบอร์ดโดยมีรายละเอียดดังนี้
  - VBAT ใช้ทำหน้าที่แสดงสถานะของแหล่งจ่ายไฟจากภายนอกที่ต่อมาให้กับบอร์ด โดย LED นี้จะติดสว่างก็ต่อเมื่อมีการจ่ายไฟให้กับบอร์ดเป็นที่เรียบร้อยแล้ว
  - STATUS ใช้แสดงสถานะของโมดูล UC15 ว่าทำงานอยู่หรือเปล่า ถ้า LED ติดแสดงว่าโมดูลทำงานอยู่ ถ้า LED ไม่ติดแสดงว่าโมดูลไม่ทำงาน หรืออยู่ในสภาวะ Power Down Mode
  - NETLIGHT ใช้แสดงสถานะของโมดูล ในขณะทำการเชื่อมต่อกับเครือข่ายอยู่ โดยเมื่อโมดูลอยู่ในสถานะพร้อมทำงาน LED นี้จะติดกระพริบด้วยค่าความ

www.etteam.com

เร็วต่างๆ ซึ่งมีความหมายดังนี้

- ติด 200ms ดับ 1800ms แสดงว่าโมดูลกำลังหาเครือข่าย
- ติด 1800ms ดับ 200ms แสดงว่าโมดูลอยู่ในสภาวะปกติ
- ติด 125ms ดับ 125ms แสดงว่าโมดูลกำลังรับส่งข้อมูล
- ติดตลอด แสดงว่าโมดูลกำลังอยู่ในโหมดการใช้งาน Voice Call
- ดับตลอด แสดงว่าโมดูลอยู่ในสภาวะ sleep

- **หมายเลข 16** โมดูล UC15
- **หมายเลข 17** คอนเน็คเตอร์ SIM Card แบบ FULL SIZE

3. การสั่ง เปิด และ ปิด การทำงานของโมดูล UC15

ตามปรกติแล้ว โมดูล UC15 จะมีโหมดการทำงานอยู่หลายโหมด เราสามารถทำงานสั่ง เปิด และปิดการทำงานของโมดูลได้ โดยใช้วิธีดังต่อไปนี้

3.1 **สวิตช์ PWRKEY (SW1)** เป็นการสั่ง เปิด และ ปิด การทำงานของโมดูล UC15 ด้วยการกดสวิตช์ โดยสวิตช์ตัวนี้ จะเป็นแบบ Push-Button Switch (สวิตช์ กดติด-ปล่อยดับ) โดยเป็นการกำหนดสถานะทางโลจิกให้กับขาสัญญาณ PWRKEY(PIN 18) ของโมดูล โดยเมื่อกดสวิตช์จะเป็นโลจิก “0” เมื่อปล่อยสวิตช์จะเป็นโลจิก “1” โดยการทำงานของสวิตช์จะต้องทำการกดสวิตช์ต่อเนื่องกันเป็นเวลามากกว่า 100mS (0.1 วินาที) จึงจะมีผลต่อการทำงานของโมดูล โดยลักษณะการทำงานของสวิตช์ จะเป็นแบบ Toggle กล่าวคือ ถ้าโมดูลอยู่ในสถานะของ Power OFF อยู่ แล้วทำการกดสวิตช์ เป็นเวลามากกว่า 100mS (0.1 วินาที) จะเป็นการสั่งให้โมดูลกลับเข้าสู่ Power ON หรือพร้อมทำงาน แต่ถ้าหากว่าโมดูลอยู่ในสถานะของ Power ON อยู่ แล้วทำการกดสวิตช์ เป็นเวลามากกว่า 600mS (0.6 วินาที) แล้วปล่อยจะเป็นการสั่งให้โมดูลหยุดทำงานและกลับเข้าสู่สถานะของ Power OFF (หยุดทำงาน) การสั่งงานด้วยวิธีนี้ต้องเลือกจั้มเปอร์ JP8 ไปที่ตำแหน่ง M

3.2 **ควบคุมการเปิดปิดทางขา PWRKEY** การสั่งเปิดปิดการทำงานของโมดูลแบบนี้จะใช้สัญญาณควบคุมจากภายนอก เช่น จากไมโครคอนโทรลเลอร์ โดยผ่านทางขา ON/OFF (คอนเน็คเตอร์ J8 หรือ J9) โดยลักษณะการทำงานจะเป็นแบบ Toggle กล่าวคือ ถ้าโมดูลอยู่ในสถานะของ Power OFF อยู่ แล้วทำการส่งลอจิก “1” เป็นเวลามากกว่า 100mS (0.1 วินาที) แล้วปล่อยเป็นลอจิก “0” จะเป็นการสั่งให้โมดูลกลับเข้าสู่ Power ON หรือพร้อมทำงาน แต่ถ้าหากว่าโมดูลอยู่ในสถานะของ Power ON อยู่ แล้วทำการการส่งลอจิก “1” เป็นเวลามากกว่า 600mS (0.6 วินาที) แล้วปล่อยเป็นลอจิก “0” จะเป็นการสั่งให้โมดูลหยุดทำงานและกลับเข้าสู่สถานะของ Power OFF (หยุดทำงาน) การสั่งงานด้วยวิธีนี้ต้องเลือกจั้มเปอร์ JP8 ไปที่ตำแหน่ง M

3.3 **เปิดการทำงานแบบอัตโนมัติ** การทำงานแบบนี้จะเปิดการทำงานโมดูล UC15 ทันทีเมื่อจ่ายไฟเลี้ยงเข้าบอร์ด ET-3G UC15 ซึ่งสามารถทำได้โดยให้เลือกจั้มเปอร์ JP8 ไปที่ตำแหน่ง A

4. การติดต่อสื่อสารกับโมดูล UC15

การติดต่อสื่อสารกับโมดูล UC15 ของบอร์ด ET-3G UC15 นั้นสามารถทำได้ 2 แบบ คือ เชื่อมต่อผ่านพอร์ตสื่อสารอนุกรม RS232 โดยใช้ขั้วต่อแบบ 4 PIN จัดเรียงสัญญาณตามมาตรฐานของบริษัท ETT หรือขั้วต่อ RS232 มาตรฐาน แบบ DB9 ตัวเมีย ซึ่งสามารถนำไปเชื่อมต่อกับสัญญาณ RS232 มาตรฐาน เช่น คอมพิวเตอร์ RS232(Com Port) หรือ บอร์ดไมโครคอนโทรลเลอร์ของบริษัท ETT ที่มีขั้ว RS232 แบบ 4 PIN ได้ทันที นอกจากนี้บอร์ด ET-3G UC15 ยังได้เตรียมขั้วต่อสัญญาณอนุกรมระดับสัญญาณ TTL 3.3-5V (J5,J8,J9) สำหรับเชื่อมต่อกับไมโครคอนโทรลเลอร์โดยตรง โดยไม่ต้องมีวงจรแปลงระดับสัญญาณเป็น RS232 โดยสัญญาณการเชื่อมต่ออนุกรมของโมดูล UC15 จะมีดังนี้

- DCD (Data Carrier Detect) ของโมดูล UC15 ซึ่งเป็น Output จาก UC15 ซึ่งตามปรกติจะต่อเข้ากับ DCD Input ของอุปกรณ์ด้าน Host
- TXD (Transmit Data) ของโมดูล UC15 ซึ่งเป็น Output จาก UC15 ซึ่งตามปรกติจะต่อเข้ากับ RXD (Receive Data) ของอุปกรณ์ด้าน Host
- RXD (Receive Data) ของโมดูล UC15 ซึ่งเป็น Input ของ UC15 ซึ่งตามปรกติจะต่อเข้ากับ TXD (Transmit Data) จากอุปกรณ์ด้าน Host
- DTR (Data Terminal Ready) ของโมดูล UC15 ซึ่งเป็น Input ของ UC15 ซึ่งตามปรกติจะต่อเข้ากับ DTR จากอุปกรณ์ด้าน Host
- RTS (Request To Send) ของโมดูล UC15 ซึ่งเป็น Input ของ UC15 ซึ่งตามปรกติจะต่อเข้ากับ RTS ของอุปกรณ์ด้าน Host
- CTS (Clear To Send) ของโมดูล UC15 ซึ่งเป็น Output จาก UC15 ซึ่งตามปรกติจะต่อเข้ากับ CTS ของอุปกรณ์ด้าน Host
- RI (Ring Indicator) ของโมดูล UC15 ซึ่งเป็น Output จาก UC15 ซึ่งตามปรกติจะต่อเข้ากับ RI ของอุปกรณ์ด้าน Host
- GND ของโมดูล UC15 ต้องต่อเข้ากับ GND ของอุปกรณ์ด้าน Host

แสดงการต่อสายสัญญาณระหว่าง ET-3G UC15 กับ ไมโครคอนโทรลเลอร์

UC15	Signal Direction	MCU
DCD	→	DCD
TXD	→	RXD
RXD	←	TXD
DTR	←	DTR
RTS	←	RTS
CTS	→	CTS
RI	→	RI
GND	___	GND

ตารางการเชื่อมต่อสัญญาณแบบเต็ม

UC15	Signal Direction	MCU
TXD	→	RXD
RXD	←	TXD
GND	___	GND

ตารางการเชื่อมต่อสัญญาณแบบ 3 เส้น

www.etteam.com

<u>คำแนะนำ</u> ในกรณีที่ใช้การเชื่อมต่อสัญญาณแบบ 3 เส้น (RXD,TXD,GND) ต้องทำการปิด Hardware Flow Control ของโมดูล UC15 โดยใช้คำสั่ง AT+IFC=0,0 แต่ถ้าต้องการต่อแบบครบทุกเส้น ก็ให้ทำการเปิด Hardware Flow Control โดยใช้คำสั่ง AT+IFC=2,2

5. คุณสมบัติการทำงานของสัญญาณที่ควรรู้

- RI(Ring Indicator) เป็น Output จากโมดูล UC15 ตามปรกติจะเป็น High แต่เมื่อมีสัญญาณเรียกเข้าจะ Active เป็น Low ตามเงื่อนไขต่อไปนี้
  - เมื่อมีสัญญาณเรียกเข้า Voice Calling สัญญาณ RI จะ Active เป็น LOW ค้างอยู่จนกว่าจะมีการตอบรับ(ATA) หรือ ได้รับคำสั่งยกเลิกการเชื่อมต่อ(ATH) หรือผู้เรียกสายทำการวางสายก่อนจะมีการตอบรับ
  - เมื่อมีสัญญาณเรียกเข้า Data Calling สัญญาณ RI จะ Active เป็น LOW ค้างอยู่จนกว่าจะมีการตอบรับ(ATA) หรือ ได้รับคำสั่งยกเลิกการเชื่อมต่อ (ATH)

- เมื่อมีสัญญาณเรียกเข้า SMS สัญญาณ RI จะ Active เป็น LOW ประมาณ 120mS และกลับเป็น HIGH โดยอัตโนมัติ

- **DTR(Data Terminal Ready)** เป็น Input ของโมดูล UC15 เมื่อต้องการให้โมดูลทำงานต้องให้ขาสัญญาณนี้ได้รับโลจิก 0 ถ้าขา DTR ได้รับโลจิก 1 โมดูลจะหยุดทำงานและเข้าสู่ Sleep Mode โดยอัตโนมัติ (ถ้ามีการสั่ง Enable Sleep Mode ด้วยคำสั่ง **AT+QSCLK=1** ไว้) ดังนั้นถ้าต้องการให้โมดูลทำงานตลอดเวลาต้องควบคุมให้ขาสัญญาณ DTR ด้านโมดูลได้รับโลจิก 0 หรือสั่งปิดการทำงานของ Sleep Mode โดยใช้คำสั่ง **AT+QSCLK=0**

6. ตัวอย่างการใช้งาน AT Command เพื่อสั่งงานโมดูล UC15

โมดูล UMTS/HSDPA รุ่น UC15 ถูกออกแบบให้ทำหน้าที่เหมือน Modem โดยจะใช้การติดต่อสั่งงานและสื่อสารกับโมดูล ผ่านทางพอร์ตสื่อสาร RS232 รองรับ Baudrate ตั้งแต่ 9600-921600 bps โดยใช้ชุดคำสั่งแบบ AT Command ซึ่งจะมีรูปแบบการใช้งานเหมือนกับ Modem มาตรฐานทั่วๆไป เพียงแต่จะมีการเพิ่มเติม Option และคำสั่งพิเศษอื่นๆเพิ่มเติมขึ้นมาอีก เพื่อให้เหมาะสมและสอดคล้องกับความสามารถในการทำงานของโมดูลได้อย่างครบถ้วน

สำหรับรายละเอียดการใช้คำสั่ง AT Command ที่จะใช้ สำหรับติดต่อสั่งงานโมดูล UC20G ไม่ว่าจะเป็น รูปแบบคำสั่ง และ หน้าที่การทำงานของแต่ละคำสั่ง ผู้ใช้สามารถศึกษารายละเอียดต่างๆได้จากคู่มือคำสั่ง AT Command (**ไฟล์เอกสารชื่อ** Quectel_UC15_AT_Commands_Manual_V1.1.pdf) ในแผ่น CD-ROM ซึ่งในที่นี้จะขอแนะนำถึงวิธีการและรูปแบบการใช้งานคำสั่งแบบย่อๆ แบบพอสังเขป เพื่อเป็นแนวทางให้ผู้เริ่มต้นได้ใช้เป็นแนวทางและประกอบความเข้าใจในการศึกษาการทำงานของคำสั่งต่างๆต่อไป โดยรูปแบบของคำสั่งต่างๆที่เป็น AT Command นั้น จะเริ่มต้นคำสั่งด้วยรหัส ASCII ของตัวอักษร 2 ตัว คือ “A” และ “T” ซึ่งจะใช้ตัวอักษรแบบพิมพ์เล็ก หรือ พิมพ์ใหญ่ก็ได้ มีความหมายเหมือนกัน จากนั้นก็จะตามด้วยรหัสคำสั่ง และ Option ต่างๆของคำสั่ง(ถ้ามี) โดยทุกๆคำสั่งจะต้องจบด้วยรหัส Enter หรือ 0DH (13) เสมอ เช่นคำสั่ง รีเซ็ต จะใช้รูปแบบคำสั่งเป็น “ATZ” หรือ “atz” ก็สามารถใช้งานได้ถูกต้องเหมือนกัน โดยรูปแบบคำสั่งทั้งหมดจะแบ่งออกเป็น 4 แบบด้วยกัน คือ

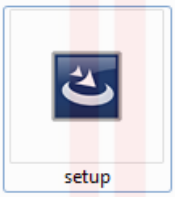
การใช้งาน	รูปแบบคำสั่ง	รายละเอียด
ทดสอบคำสั่ง	**AT+<x>=?**	รูปแบบการใช้คำสั่งแบบนี้ จะใช้สำหรับสั่งอ่านค่ารูปแบบและพารามิเตอร์ต่างๆของคำสั่ง โดยถ้าคำสั่งนั้นมีอยู่จริง โมดูลจะตอบรับด้วยการพิมพ์ค่าของพารามิเตอร์ต่างๆของคำสั่งที่มีอยู่ทั้งหมดให้ทราบ
อ่านค่าพารามิเตอร์	**AT+<x>?**	รูปแบบการใช้คำสั่งแบบนี้ จะใช้สำหรับสั่งอ่านค่าพารามิเตอร์

		ที่กำหนดไว้แล้วของคำสั่งนั้นๆ โดยโมดูลจะตอบรับด้วยการพิมพ์ค่าพารามิเตอร์ปัจจุบันที่กำหนดไว้แล้วให้ทราบ
กำหนดค่าการทำงาน	**AT+<x>=<...>**	รูปแบบการใช้คำสั่งแบบนี้ จะใช้สำหรับสั่งเขียนหรือกำหนดค่าพารามิเตอร์ให้กับคำสั่ง เช่น การกำหนดค่า Baudrate
สั่งให้ทำงาน	**AT+<x>**	รูปแบบการใช้คำสั่งแบบนี้ จะใช้สำหรับสั่งงานให้โมดูลปฏิบัติตามคำสั่งที่ต้องการ เช่น การสั่งรีเซ็ต (ATZ)

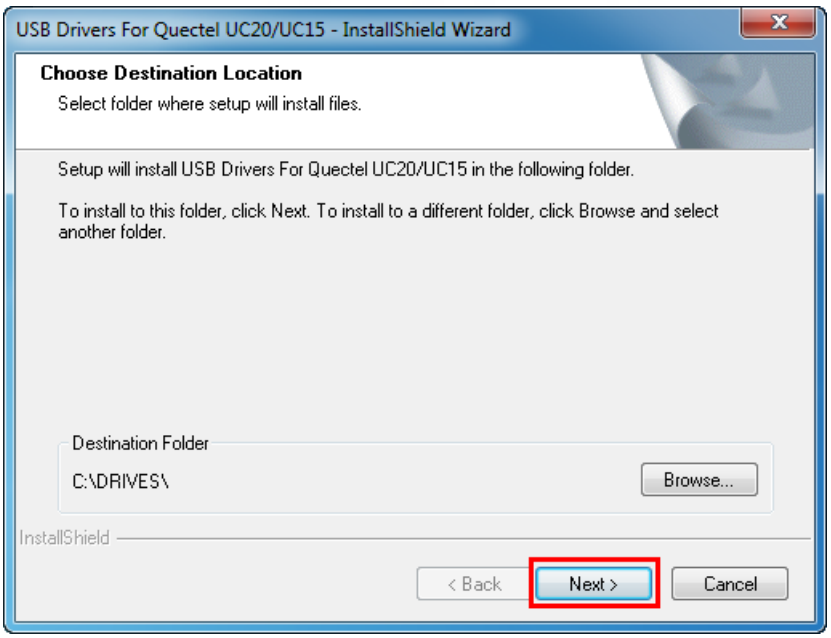
ตารางแสดง รูปแบบการใช้งาน AT Command (เมื่อ <x> คือ รหัสคำสั่ง)

7. การติดตั้งไดร์เวอร์ USB ของบอร์ด ET-3G UC15

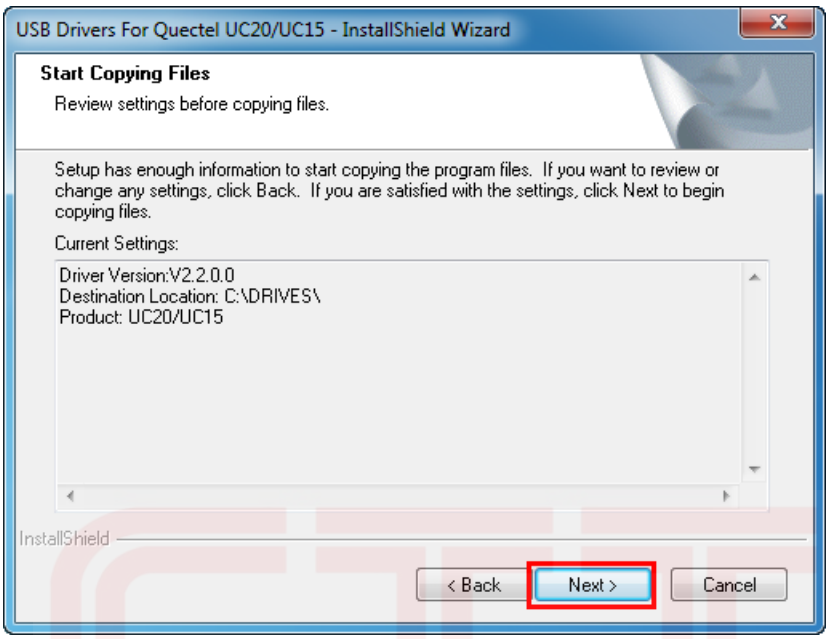
7.1 ไปที่ไดเรคทอรี่ USB Driver ใน CD ROM และคลิกที่ setup เพื่อทำการติดตั้ง



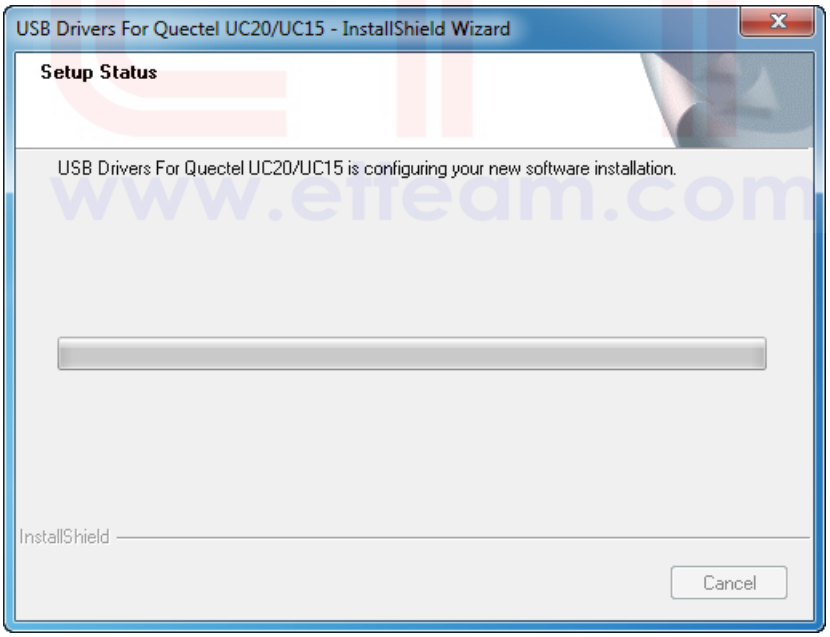
7.2 คลิก Next > เพื่อเริ่มติดตั้ง โดย Destination Folder จะใช้ค่าเริ่มต้น



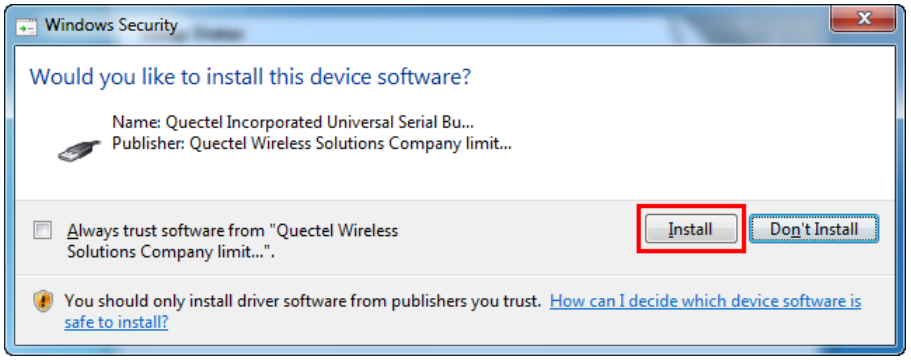
7.3 จากนั้นคลิก Next >

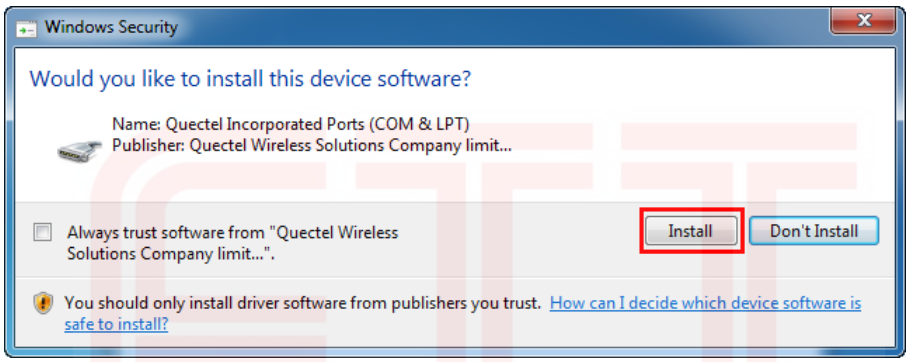


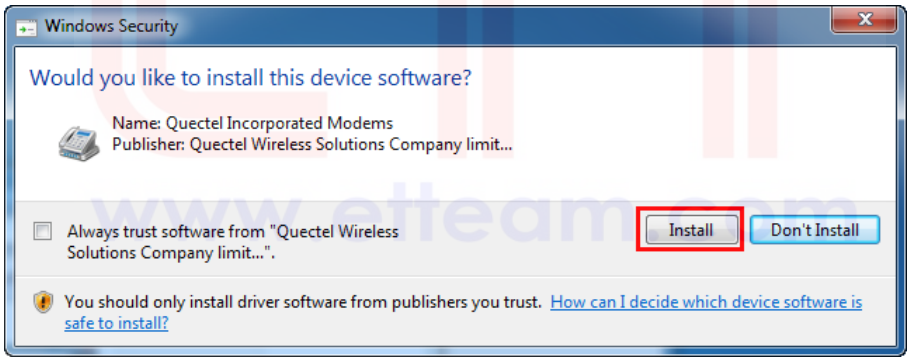
7.4 จากนั้นโปรแกรมจะเริ่มติดตั้ง Driver

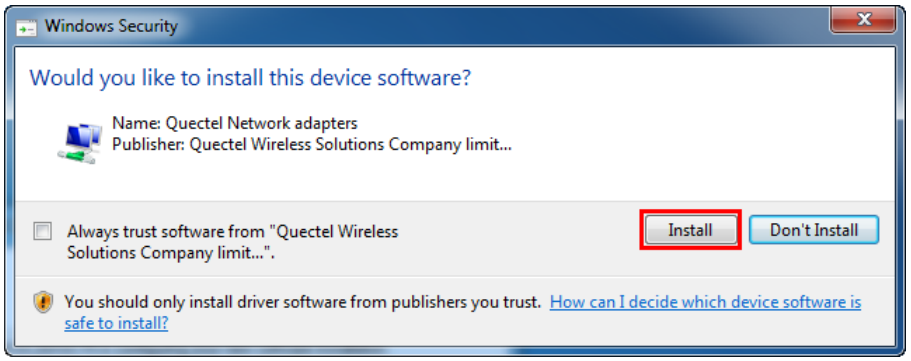


7.5 จากนั้นจะมีหน้าต่าง Windows Securrity ฟ้องขึ้นมาให้เลือก Install

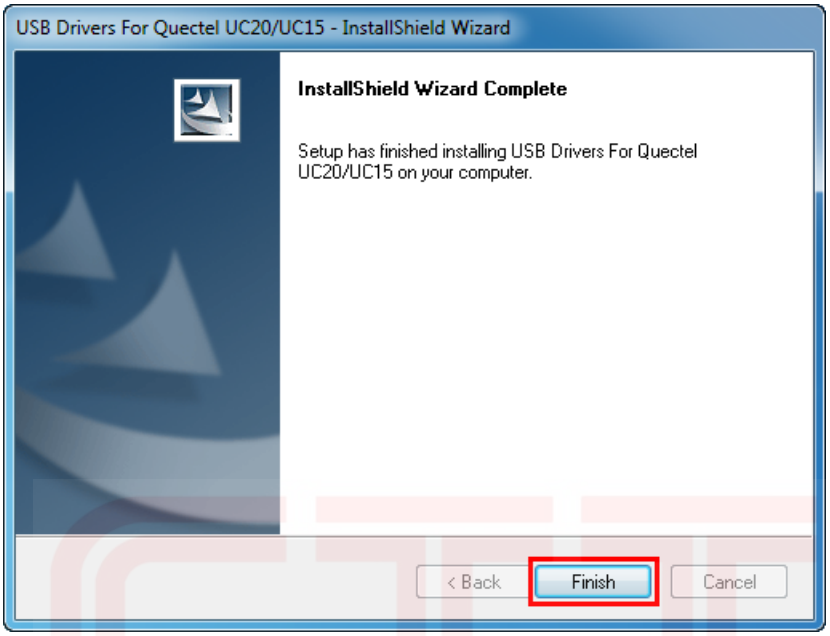




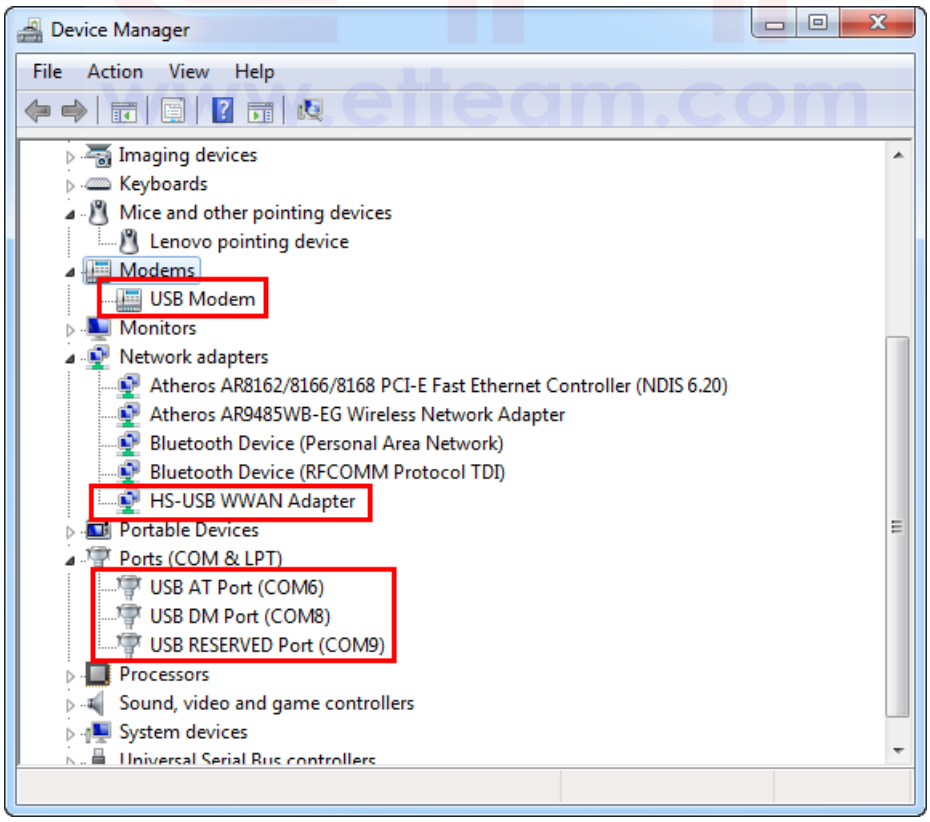




7.6 เมื่อการติดตั้งไดร์เวอร์เสร็จเรียบร้อย จะมีหน้าต่างดังรูป ให้คลิกที่ปุ่ม Finish



7.7 ให้ทำการต่อไฟเลี้ยงเข้าบอร์ด ET-3G UC15 ต่อสาย USB เข้ากับคอมพิวเตอร์ จากนั้นให้เปิดการทำงานของโมดูล จากนั้น Windows จะตรวจพบฮาร์ดแวร์ใหม่โดยอัตโนมัติ รอจนเรียบร้อยจะเห็นฮาร์ดแวร์ใหม่เพิ่มขึ้นมาดังรูป



จะเห็นว่าในรายการ Port (COM & LPT) จะมีพอร์ต เพิ่มขึ้นมา 3 พอร์ต คือ

- USB AT Port ใช้สำหรับส่ง AT Commands เพื่อควบคุมโมดูล UC15
- USB DM Port ใช้สำหรับอัพเกรดเฟริมแวร์ของโมดูล UC15
- USB RESERVED Port จองไว้ไม่ได้ใช้งาน

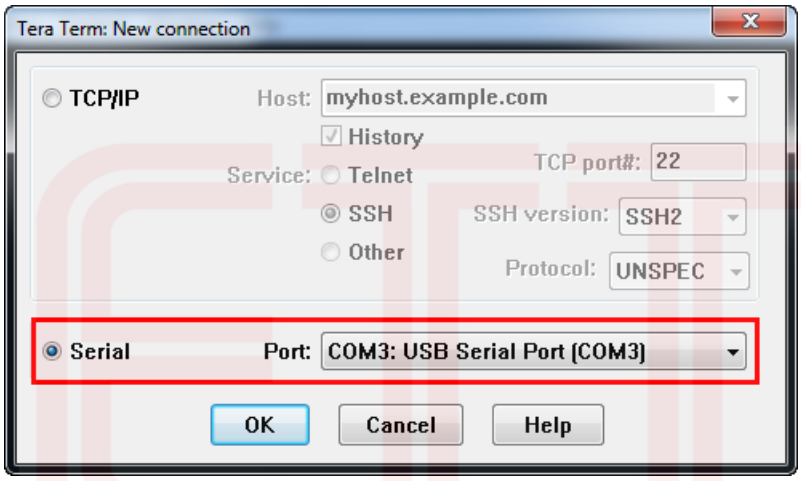
8. การทดสอบการทำงานของบอร์ด ET-3G UC15

ดังได้ทราบแล้วว่าในการสั่งงานโมดูล UC15 นั้น จะใช้วิธีการส่งค่ารหัสคำสั่งในรูปแบบของ AT Command ผ่านทางพอร์ตสื่อสารอนุกรมไปให้กับโมดูล ซึ่งตามปรกติจะต้องเขียนโปรแกรมเพื่อส่งรหัสคำสั่งต่างๆไปให้กับโมดูลเอง ขึ้นอยู่กับว่าจะใช้อุปกรณ์ใดเป็นตัวควบคุมการทำงานของโมดูล ซึ่งไม่ได้จำกัดว่าเป็นอุปกรณ์แบบใด อาจจะเป็นคอมพิวเตอร์ PC หรือ ไมโครคอนโทรลเลอร์ตระกูลใดๆก็ได้ ขอให้มีพอร์ตสื่อสารอนุกรม RS232 อยู่ก็สามารถนำมาเชื่อมต่อเพื่อสั่งงานโมดูล UC15 ได้แล้ว ส่วนที่ว่าจะเขียนโปรแกรมอย่างไร และจะใช้ภาษาใดในการเขียนนั้น ขึ้นอยู่กับผู้พัฒนาโปรแกรมว่า มีความถนัดอย่างไรและมีพื้นฐานอะไรอยู่บ้าง ซึ่งหลักสำคัญก็คือ ผู้พัฒนาต้องหาคำตอบให้ได้ว่า การจะเขียนโปรแกรมสั่งงานอุปกรณ์ทำการส่ง และ รับข้อมูล ด้วยพอร์ตสื่อสารอนุกรม RS232 นั้นจะต้องทำอย่างไร ซึ่งจะไม่ขอกล่าวถึงในที่นี้ด้วย

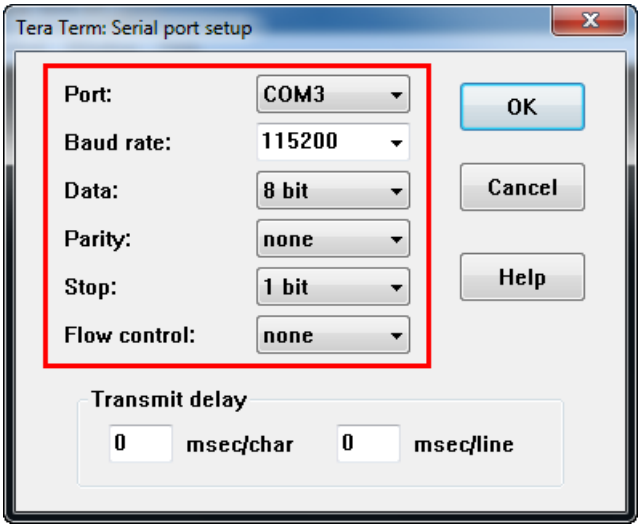
สำหรับในการศึกษาเบื้องต้นนั้น ยังไม่จำเป็นต้องใช้วิธีการเขียนโปรแกรมก็ได้ แต่สามารถใช้โปรแกรมสำเร็จรูปจำพวก Serial Terminal ต่างๆ ของคอมพิวเตอร์เป็นตัวทดสอบการทำงานเพื่อทำความเข้าใจกับรูปแบบคำสั่งและผลของการทำงานต่างๆให้เข้าใจเสียก่อน ตัวอย่างเช่น ถ้าต้องการจะสั่งให้โมดูล UC15 โทรออกไปยังโทรศัพท์มือถือหมายเลข 0811234567 นั้น ในอันดับแรกจะต้องศึกษารูปแบบการทำงานของคำสั่งให้เข้าใจเสียก่อน จนสามารถเข้าใจแล้วว่าจะต้องใช้คำสั่ง “ATD0811234567;” เพื่อสั่งให้โทรออก จากนั้นจึงค่อยปรับเปลี่ยนไปเป็นการเขียนโปรแกรมในภายหลัง ซึ่งผู้ใช้ก็จะต้องไปศึกษาหาคำตอบต่อไปอีกว่าการที่จะเขียนโปรแกรมเพื่อสั่งให้อุปกรณ์ส่งค่ารหัส “ATD0811234567;” ออกไปทางพอร์ตสื่อสารอนุกรมนั้นต้องทำอย่างไรบ้าง ซึ่งในที่นี้จะขอแนะนำให้ใช้โปรแกรม HyperTerminal ของ Windows เป็นเครื่องมือในการทดลองในเบื้องต้นไปก่อน โดย HyperTerminal เป็นโปรแกรม Terminal สำเร็จรูป ซึ่งแถมมาพร้อมกับระบบปฏิบัติการ Windows อยู่แล้ว โดยความสามารถของโปรแกรมตัวนี้จะมีอยู่มากมายหลายส่วน ซึ่งในที่นี้เราจะใช้ประโยชน์เฉพาะในส่วนของการทำหน้าที่เป็น Serial Terminal ใน Text Mode เท่านั้น โดยหลังจากสั่ง Run โปรแกรมแล้ว ข้อมูลใดๆที่รับได้จากสัญญาณด้านรับ (RXD) ของพอร์ตสื่อสารอนุกรม ในย่านที่เป็นรหัส ASCII Code (20H..FFH) จะถูกนำมาแปลงเป็นตัวอักษรและแสดงผลที่หน้าจอของโปรแกรมให้เห็นทันที ส่วนรหัสของข้อมูลที่มีค่าต่ำกว่า 20H (00H-1FH) จะไม่ถูกนำมาแสดงผล แต่จะถือว่าเป็นคำสั่ง เช่น เมื่อได้รับ รหัส 0DH โปรแกรม Hyper Terminal จะถือว่าเป็นคำสั่งให้เลื่อน Cursor ของการแสดงผลไว้ในตำแหน่งเริ่มต้นของบรรทัด หรือเมื่อได้รับรหัส 0AH ก็จะทำการ

เลื่อน Cursor ของการแสดงผลให้ขึ้นบรรทัดใหม่แทนดังนี้เป็นต้น และ ในทางตรงกันข้าม เมื่อเราทำการกดคีย์ใดๆ โปรแกรมก็จะแปลค่าการกดคีย์นั้นให้เป็นรหัส ASCII ของตัวอักษรของตำแหน่งคีย์นั้นๆส่งออกไปยังขา TXD ของพอร์ตสื่อสารอนุกรมโดยอัตโนมัติ แต่ถ้าถ้าเครื่องผู้ใช้เป็น Windows7, Windows8 จะไม่มีโปรแกรม HyperTerminal ติดมา ดังนั้นในตัวอย่างนี้จะใช้โปรแกรม Tera Term แทน ซึ่งการใช้งานมีดังนี้

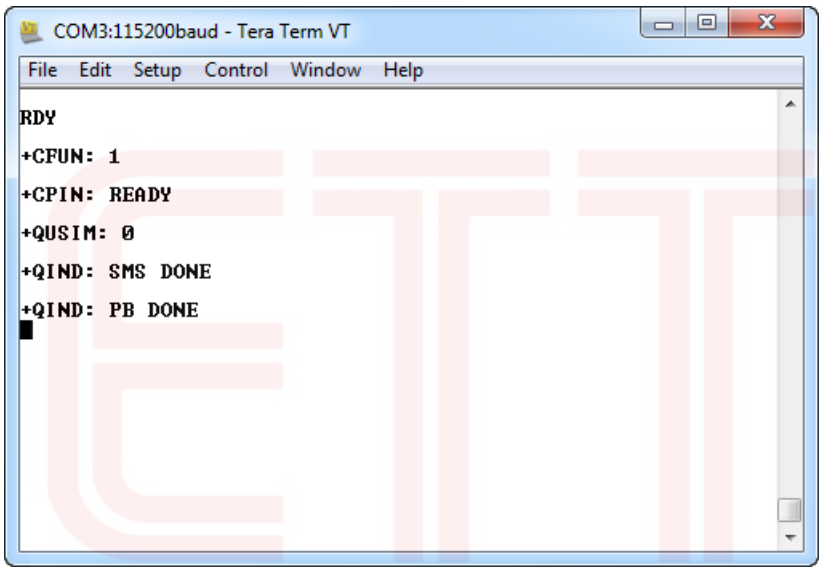
8.1 ทำการติดตั้งโปรแกรม Tera Term จากนั้นให้เปิดโปรแกรม เลือกการเชื่อมต่อเป็น Serial และเลือกพอร์ตที่จะเชื่อมต่อกับ ET-3G UC15 จากนั้นคลิก OK ดังรูป



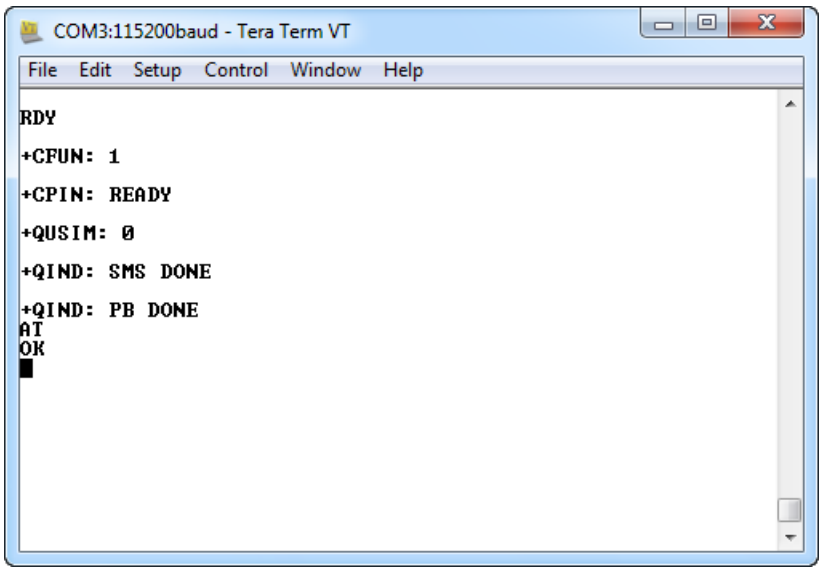
8.2 เลือกที่เมนู Setup-->Serial port... เพื่อตั้งค่าของพอร์ต RS232 ในขั้นตอนนี้ให้เลือกค่า Baud rate ให้ตรงและสอดคล้องกับที่กำหนดให้กับโมดูลไว้ ในที่นี้จะเลือกเป็น 115200 Data 8 Bit,Parity =None, Stop bits=1, Flow Control = None ซึ่งเป็นค่าเริ่มต้นของโมดูล แล้วคลิก OK ดังตัวอย่าง



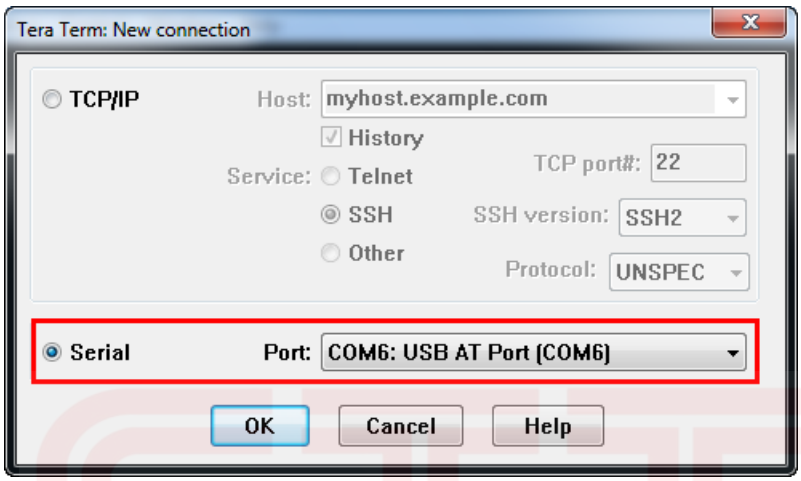
8.3 ซึ่งหลังจากกำหนดการเชื่อมต่อต่างๆเรียบร้อยแล้ว ถ้าทุกอย่างถูกต้องให้ทดลองทำการต่อสายสัญญาณ RS232 ระหว่างบอร์ดกับพอร์ต RS232 ของคอมพิวเตอร์ (ต้องเลือกจั้มเปอร์ JP1-JP4 ไปที่ตำแหน่ง RS232 ด้วย) แล้วจ่ายไฟเลี้ยงวงจรให้กับบอร์ด ซึ่งถ้าทุกอย่างถูกต้องจะเห็น LED VBAT บนบอร์ดติดสว่างให้เห็น จากนั้นให้สั่ง Power ON ตัวโมดูล โดยการกดสวิตช์ PWRKEY ค้างไว้ประมาณ 0.1 วินาที จะสังเกตเห็น LED STATUS ติดสว่างขึ้น จากนั้น LED STATUS ก็จะเริ่มกระพริบเป็นจังหวะตลอดเวลา แสดงว่าโมดูลเริ่มต้นทำงานแล้ว ส่วนที่หน้าจอของ Tera Term จะปรากฏข้อความดังรูป จากนั้นก็จะสามารถสั่งงานโมดูลด้วยคำสั่งต่างๆ ได้ตามต้องการดังตัวอย่าง



ให้ผู้ใช้ทดลองพิมพ์ตัวอักษร AT และตามด้วยกดปุ่ม Enter จากนั้นโมดูลจะตอบ OK กลับมาดังรูป แสดงว่าโมดูลพร้อมทำงานแล้ว



หมายเหตุ ในกรณีที่ผู้ใช้เชื่อมต่อบอร์ด ET-3G UC15 ผ่านทางสาย USB แนะนำให้เชื่อมต่อบอร์ดให้เรียบร้อยก่อน แล้วจึงเปิดการทำงานของโมดูล แล้วจึงเลือกพอร์ตการเชื่อมต่อ โดยจะต้องเลือกพอร์ตเชื่อมต่อเป็น USB AT Port ดังรูป



9. ตัวอย่างการใช้งานคำสั่งเกี่ยวกับ RS232

ในที่นี้จะขอยกตัวอย่างคำสั่งที่ใช้งานบ่อยๆ ถ้าต้องการดูข้อมูลเพิ่มเติมให้ศึกษาจากเอกสาร Quectel_UC15_AT_Commands_Manual_V1.1.pdf

9.1 ตัวอย่างการใช้งานคำสั่งตั้งค่า Baud rate

สมมุติว่าเราต้องการใช้คำสั่งสำหรับกำหนดค่าอัตราความเร็วของการสื่อสารของโมดูล ซึ่งจะต้องใช้คำสั่ง AT+IPR โดยเราสามารถสั่งงานคำสั่งนี้ได้หลายรูปแบบดังตัวอย่างต่อไปนี้ คือ ถ้าเราจำไม่ได้ว่าค่าพารามิเตอร์ของคำสั่งหรืออัตรา Baud rate ที่สามารถกำหนดได้ มีค่าอะไรบ้างและกำหนดอย่างไร เราก็สามารถใช้รูปแบบการทดสอบคำสั่ง โดยใช้คำสั่ง AT+IPR=? และจบด้วย Enter เพื่อสอบถามได้ โดยโมดูลจะตอบรับด้วย +IPR: พร้อมกับพิมพ์ค่าพารามิเตอร์ของคำสั่งที่มีอยู่ทั้งหมดให้ทราบ คือ 9600,19200,38400,…,4000000 ดังตัวอย่าง (**ข้อความสีดำคือคำสั่งที่ป้อนเข้าไป** **ส่วนสีแดงคือข้อความที่โมดูลตอบกลับมา**)

```
AT+IPR=?
+IPR: (9600,19200,38400,57600,115200,230400,460800,921600,2900000,
3200000,3686400,4000000)

OK
```

ถ้าต้องการทราบว่าในขณะนี้ ค่า Baud rate ที่กำหนดไว้แล้ว มีค่าเป็นเช่นไร ก็สามารถใช้รูปแบบคำสั่งสำหรับสั่งอ่านค่าพารามิเตอร์ของคำสั่งนี้ คือ AT+IPR? ซึ่งโมดูลจะตอบรับด้วย +IPR: ตามด้วยค่าพารามิเตอร์ที่กำหนดไว้แล้วให้ทราบดังตัวอย่าง

```
AT+IPR?
+IPR: 115200

OK
```

จากตัวอย่างข้างต้น จะเห็นได้ว่าค่าพารามิเตอร์ของคำสั่ง IPR ที่กำหนดไว้แล้วคือ 115200 แต่ถ้าผู้ใช้ถ้าต้องการกำหนด Baud rate เป็นค่าอื่นเช่น 9600 ก็จะใช้รูปแบบคำสั่งเป็น AT+IPR=9600 และตามด้วยกดปุ่ม Enter ดังตัวอย่าง จากนั้นให้ใช้คำสั่ง AT&W และตามด้วยกดปุ่ม Enter เพื่อบันทึกค่า

```
AT+IPR=9600
OK
```

ซึ่งหลังจากสั่งเปลี่ยนค่า Baud rate เป็น 9600 แล้ว และบันทึกค่า ต่อจากนี้ไปก็สามารถสื่อสารกับโมดูลด้วยค่าความเร็วนี้ได้ตลอด แต่ถ้าผู้ใช้ไม่ได้ใช้คำสั่ง AT&W เมื่อปิดโมดูลและกลับมาเปิดใหม่ ค่าจะกลับเป็นค่าก่อนที่จะตั้งค่าใหม่

9.2 การกำหนด Flow Control

โมดูล UC15 สามารถกำหนด Flow Control หรือ รูปแบบการตรวจสอบความพร้อมในการสื่อสารและรับส่งข้อมูลได้ด้วย ซึ่ง Flow Control จะมีความสำคัญเป็นอย่างมาก เนื่องจากการประมวลผลของอุปกรณ์ต่างๆจะมีความช้าเร็วที่แตกต่างกัน เมื่อมีการรับส่งข้อมูลที่มีจำนวนข้อมูลมากๆ แบบต่อเนื่องนั้น ถ้าฝ่ายรับไม่พร้อมรับข้อมูลแต่ฝ่ายส่งยังคงส่งข้อมูลออกไป ก็จะทำให้ข้อมูลสูญหายและเกิดความผิดพลาดขึ้นได้ โดยโมดูล UC15 เองรองรับการตรวจสอบความพร้อมแบบ Hardware Flow Control

- Hardware Flow Control (RTS/CTS Flow Control) เป็นการตรวจสอบความพร้อมด้วยสัญญาณทางฮาร์ดแวร์ โดยใช้การ Active(“LOW”) สัญญาณ CTS เพื่อบอกให้ฝ่ายส่งหยุดการส่งข้อมูลเมื่อโมดูลไม่พร้อมรับข้อมูล และในทางกลับกันก่อนการส่งข้อมูลกลับออกไปมัน

จะตรวจสอบสถานะของ RTS ว่า Active อยู่หรือไม่ ถ้า Active(“LOW”) แสดงว่าฝ่ายรับยังไม่พร้อมรับมันจะหยุดรอจนกว่า RTS จะเป็น “HIGH”

การปิด Hardware Flow Control จะใช้คำสั่ง AT+IFC=0,0 **(ค่าเบื้องต้นจากโรงงาน)**
การเปิด Hardware Flow Controlจะใช้คำสั่ง AT+IFC=2,2

9.3 การกำหนด Format ข้อมูลของ RS232

เราสามารถกำหนด Format ของข้อมูล ได้ว่าจะใช้รูปแบบการส่งข้อมูลเป็นอย่างไร ใช้ขนาดข้อมูลเป็นกี่บิต ใช้บิตตรวจสอบความผิดพลาด Parity หรือไม่ และต้องการใช้ Stop Bit เป็นกี่บิต ซึ่งตามปรกติทั่วไปแล้วจะใช้ Data 8 Bit ,None Parity,1 Stop Bit ซึ่งสามารถเปลี่ยนแปลงได้โดยใช้คำสั่ง AT+ICF

การกำหนด Format ข้อมูลเป็น 8 Bit Data ,None Parity ,1 Stop Bit จะใช้คำสั่ง AT+ICF=3,3

9.4 การเปิดปิดการ Echo

การ Echo คือ เมื่อผู้ใช้พิมพ์คำสั่งต่างๆ บนโปรแกรม Terminal จะมีการส่งคำสั่งที่พิมพ์กลับมาเพื่อให้ผู้ใช้รู้ว่าได้พิมพ์อะไรไป (ปกติจะเป็นค่าเริ่มต้นจากโรงงาน) แต่ในกรณีที่ผู้ใช้ไม่ต้องการก็สามารถปิดความสามารถนี้ได้ด้วยการใช้คำสั่ง ATE0 แล้วตามด้วย Enter ซึ่งเมื่อพิมพ์คำสั่งลงไปจะไม่เห็นคำสั่งที่พิมพ์บนโปรแกรม Terminal จะแสดงเพียงค่าที่โมดูล UC15 ตอบกลับมาเท่านั้น ถ้าต้องการกลับมาใช้ใหม่ก็ใช้คำสั่ง ATE1 แล้วตามด้วย Enter

10. การ Setup และตรวจสอบค่า Configuration

ตามปรกติแล้วการทำงานของโมดูล UC20G นั้นจะสามารถกำหนดรูปแบบการทำงานได้มากมายหลายลักษณะ เช่น เงื่อนไขในการติดต่อสื่อสารกับโมดูล ผู้ใช้สามารถเปลี่ยนแปลงค่าต่างๆได้มากมาย ไม่ว่าจะเป็นค่า Baud rate หรือรูปแบบของการ Handshake ต่างๆ ที่จะใช้ในการสื่อสาร เป็นต้นดังนั้นจึงจำเป็นต้องมีการกำหนดรูปแบบการทำงานของโมดูลให้ตรงกับความต้องการ ซึ่งตามปรกติแล้วเงื่อนไขต่างๆเหล่านี้จะมีค่าที่แน่นอนอยู่ค่าหนึ่งเสมอหลังการรีเซ็ต หรือ Power ON โดยโมดูลจะกำหนดค่าเงื่อนไขต่างๆให้กับตัวมันเองในตอนเริ่มต้นการทำงานด้วยค่าที่กำหนดไว้ใน Configuration ที่ถูกบันทึกไว้แล้ว แต่อย่างไรก็ตามผู้ใช้สามารถสั่งเปลี่ยนแปลงแก้ไขค่า Configuration ต่างๆได้เองตามต้องการ ซึ่งวิธีการกำหนดเงื่อนไขการทำงานให้กับโมดูลนั้นสามารถทำได้ 2 แบบ

- **การกำหนดค่าแบบถาวร** จะเป็นการสั่งบันทึกค่าเงื่อนไขการทำงานต่างๆของโมดูลตามรูปแบบที่เรากำหนดไว้ในหน่วยความจำถาวรภายในตัวโมดูล โดยใช้คำสั่ง AT&W ซึ่งหลังจากโมดูลเริ่มต้นทำงานใหม่ หรือ หลังการรีเซ็ตโมดูลแต่ละครั้ง ค่าการทำงานต่างๆของโมดูลจะถูกกำหนดเงื่อนไขตามที่เรากำหนดไว้แล้วเสมอ
- **การกำหนดค่าแบบชั่วคราว** เป็นการใช้คำสั่ง AT Command ต่างๆ เพื่อกำหนดเงื่อนไขการทำงาน ให้กับโมดูล แต่ไม่มีการสั่งบันทึกค่า Configuration ด้วยคำสั่ง AT&W ซึ่งการทำงานของโมดูลก็จะปรับเปลี่ยนไปตามการสั่งงานในขณะนั้นๆ แต่เมื่อสั่งรีเซ็ตการทำงานของโมดูล หรือ มีการ Power ON ใหม่คุณสมบัติการทำงานของโมดูลจะถูกเปลี่ยนกลับเป็นค่าเดิมอีก
  โดยเราสามารถใช้คำสั่ง AT Command ในการสั่ง ตรวจสอบ และ บันทึกค่า Configuration ต่างๆ ให้กับโมดูล UC15 ได้ดังนี้
  - ใช้คำสั่ง AT&V เพื่อสั่งให้โมดูลแสดงค่า Configuration ปัจจุบันให้ทราบ
  - ใช้คำสั่ง AT&F เพื่อสั่งกำหนดค่า Configuration ทั้งหมดให้กลับเป็นค่ามาตรฐาน
  - ใช้คำสั่ง AT&W เพื่อสั่งบันทึกค่า Configuration ด้วยค่าที่เรากำหนดไว้ในขณะนั้นๆ

ค่า Configuration ที่แนะนำ

- AT+CMGF=1 (SMS Message = Text Mode)
- ATE1 (Echo Mode ON)
- AT+QSCLK=0 (Disable Sleep Mode)

11. การตรวจสอบข้อมูลต่าง ๆ ของโมดูล UC15

11.1 การตรวจสอบคุณภาพสัญญาณ

การตรวจสอบคุณภาพสัญญาณ จะใช้คำสั่ง AT+CSQ โดยคำสั่งนี้ใช้ตรวจสอบระดับความแรงของสัญญาณ โดยโมดูลจะตอบรับเป็นค่าตัวเลข 0..31 โดยถ้าค่าระหว่าง 2..30 อยู่ในเกณฑ์ดี ถ้าค่า 31 ถือว่าระดับสัญญาณดีมาก ส่วน 99 หมายถึงตรวจสอบไม่ได้ รูปแบบดังตัวอย่าง

```
AT+CSQ
+CSQ: 12,99

OK
```

11.2 การตรวจสอบ รหัสผลิตภัณฑ์

```
ATI
Quectel
UC15
Revision: UC15TQAR03A02E1G

OK
```

11.3 การตรวจสอบ รหัสผู้ผลิต

```
AT+GMI
Quectel

OK
```

11.4 การตรวจสอบ รหัสรุ่น

```
AT+GMM
UC15

OK
```

11.5 การตรวจสอบ รหัส Version Firmware

```
AT+GMR
UC15TQAR03A02E1G

OK
```

11.6 การตรวจสอบ Serial Number (IMEI) ของโมดูล

```
AT+GSN
863835020558354

OK
```

11.7 การตรวจสอบ รหัสเครือข่าย SIM ผู้ให้บริการ

```
AT+COPS?
+COPS: 0,0,"TRUE-H",2

OK
```

12. การโทรออก การรับสาย และ การยกเลิกการโทร

- ใช้คำสั่ง ATD เพื่อสั่งโทรออก โดยรูปแบบการใช้คำสั่งให้ตามด้วยเบอร์ปลายทาง
- ใช้คำสั่ง ATA เพื่อรับสายเรียกเข้า โดยเมื่อมีสายเรียกเข้าจะมีข้อความ RING แสดงให้เราทราบ ถ้าต้องการรับสายให้ใช้คำสั่ง "ATA" เพื่อรับสายได้ทันที ซึ่งหลังจากสั่งรับสายแล้วผู้ใช้จะสามารถพูดคุยกับปลายสายได้ทันที โดยใช้ไมโครโฟนและลำโพง
- ใช้คำสั่ง ATH เพื่อสั่งวางสาย หรือยกเลิกการโทรออก

ตัวอย่างการโทรออก ซึ่งเป็นการสื่อสารด้วย Voice จะต้องปิดท้ายคำสั่งด้วยเครื่องหมายเซมิโคลอน (;) และจบคำสั่งด้วย Enter (0x0D) เช่นถ้าต้องการโทรออกไปยังเบอร์ 0894469xxx จะเป็นดังนี้

```
ATD0894469xxx;
OK
```

ในกรณีที่สั่งโทรออกแล้วสายไม่ว่าง โมดูลจะรายงานผลด้วยข้อความ "BUSY" ดังตัวอย่าง

```
ATD0894469xxx;
OK

BUSY
```

ส่วนในกรณีที่สั่งโทรออกแล้วไม่มีคนรับสาย โมดูลจะรายงานผลด้วยข้อความ "NO ANSWER" ดังตัวอย่าง

```
ATD0894469xxx;
OK
```

```
NO ANSWER
```

ตัวอย่างการรับสายเรียกเข้า เมื่อมีสายเรียกเข้าโมดูล UC15 จะมีข้อความ “RING” ออกมาตลอดเวลา ถ้าผู้ใช้ต้องการรับสาย ให้ใช้คำสั่ง ATA เพื่อสั่งรับสาย หรือใช้คำสั่ง ATH เพื่อวางหูหรือยกเลิกไม่รับสาย ดังตัวอย่าง

```
RING
ATA
OK
```

ในกรณีที่ต้องการให้แสดงหมายเลขที่โทรเข้ามา ให้ใช้คำสั่ง AT+CLIP=1 ตามด้วย Enter โมดูลจะแสดงหมายเลขที่โทรเข้าดังตัวอย่าง

```
RING

+CLIP: "0894469XXX",161,,,,0
```

หมายเหตุ ในกรณีที่โปรแกรม Terminal ไม่แสดงข้อความเตือนต่างๆ เช่น RING ให้ผู้ใช้ตรวจสอบการตั้งค่า urcport ว่าตรงกับที่เชื่อมต่อบอร์ดอยู่หรือไม่ โดยใช้คำสั่ง AT+QURCCFG? เพื่อตรวจสอบ ซึ่งถ้าผู้ใช้เชื่อมต่อผ่านพอร์ต RS232 หรือ UART ค่า urcport ต้องมีค่าเป็น “uart1” และถ้าเชื่อมต่อผ่านทางพอร์ต USB (USB AT Port) ต้องมีค่าเป็น “usbat” ถ้าไม่ตรงกันก็ให้ทำการตั้งค่าใหม่โดยใช้คำสั่ง

```
AT+QURCCFG=”urcport”,”uart1”        //เมื่อเชื่อมต่อผ่านพอร์ต RS232 หรือ uart
AT+QURCCFG=”urcport”,”usbat”        //เมื่อเชื่อมต่อผ่านพอร์ต USB (USB AT Port)
```

13. การตรวจสอบยอดเงินคงเหลือโดยใช้ USSD

สามารถใช้คำสั่ง AT+CUSD ตามด้วยค่า USSD (Unstructure Supplementary Service Data) สำหรับตรวจสอบยอดเงินได้เช่นกันดังตัวอย่าง ซึ่งจากตัวอย่างจะเป็นของระบบ truemove H ซึ่งจะใช้รหัส #123# ในการตรวจสอบยอดเงินคงเหลือ

```
AT+CUSD=1,"#123#"
OK
```

```
+CUSD: 0,"TrueMove-H credit is 64.64 Bt, you can use service until 30/09/2014",15
```

14. การรับข้อความ SMS

ตามปรกติแล้วโมดูล UC15 จะสามารถกำหนดโหมดการทำงานของข้อความหรือ SMS ได้ 2 โหมด คือ PDU Mode และ Text Mode โดย PDU Mode การรับและแสดงผลการทำงานของคำสั่งจะเป็นรูปแบบของรหัสตัวเลขแบบ Binary Code ส่วน Text Mode การรับและแสดงผลการทำงานของคำสั่งจะเป็นข้อความ ซึ่งจะง่ายต่อการแปลความหมายและทำความเข้าใจมากกว่า PDU Mode ซึ่งในการทดสอบจะขอแสดงให้เห็นด้วย Text Mode

- ใช้คำสั่ง AT+CMGF=1 เพื่อกำหนดรูปแบบของข้อความเป็น Text Mode ซึ่งเมื่อมีการส่งข้อความ SMS มายังโมดูล จะมีข้อความแจ้งให้ทราบ เช่น +CMTI: "SM",3 ซึ่งหมายความว่า มีข้อความส่งเข้าและเก็บไว้ในหน่วยความจำลำดับที่ 3
- ใช้คำสั่ง AT+CMGR เพื่อสั่งอ่านข้อความ เช่นถ้าต้องการอ่านข้อความลำดับที่ 3 ก็ให้ใช้คำสั่งเป็น AT+CMGR=3
- ใช้คำสั่ง AT+CMGL="ALL" เพื่อสั่งแสดงข้อความทั้งหมดที่เก็บไว้ในหน่วยความจำ โดยสามารถเลือกประเภทของข้อความได้ เช่น ข้อความใหม่ ข้อความทั้งหมด
- ใช้คำสั่ง AT+CMGD เพื่อสั่งลบข้อความออกจากหน่วยความจำ เช่น ถ้าต้องการสั่งลบข้อความลำดับที่ 3 ก็ให้ใช้คำสั่งเป็น AT+CMGD=3
- ใช้คำสั่ง AT+CMGD=1,4 เพื่อสั่งลบข้อความทั้งหมดออกจากหน่วยความจำ

ตัวอย่างการรับข้อความ SMS ในตัวอย่างจะทดสอบด้วยการส่งข้อความ “Hello 12345” ไปให้กับโมดูล UC20G ซึ่งเมื่อรับข้อความได้จะมีข้อความ +CMTI: “SM”,n โดย n หมายถึงลำดับที่ของข้อความ

```
+CMTI: "SM",3                                          // มีข้อความเข้าอันดับที่ 3
AT+CMGR=3                                              // อ่านข้อความอันดับที่ 3
+CMGR: "REC UNREAD","+66894469XXX",,"14/08/06,11:22:23+28"
Hello 12345                                            // ข้อความที่ได้รับมา

OK
```

ถ้ามีการสั่งอ่านข้อความเดิมซ้ำใหม่สถานะของข้อความจะเปลี่ยนเป็น “REC READ” แทน เพื่อแสดงให้ทราบว่าข้อความนี้ถูกอ่านไปแล้วดังตัวอย่าง

```
AT+CMGR=3
+CMGR: "REC READ","+66894469XXX",,"14/08/06,11:22:23+28"
Hello 12345

OK
```

15. การส่งข้อความ SMS ภาษาอังกฤษ

ก่อนการส่ง SMS นั้นต้องทำการตั้งค่ากำหนดรูปแบบของข้อความเป็น Text Mode โดยใช้คำสั่ง AT+CMGF=1 เลือกพารามิเตอร์ของ SMS โดยใช้คำสั่ง AT+CSMP=17,167,0,0 จากนั้นเลือกชุดของตัวอักษรที่จะส่งโดยใช้คำสั่ง AT+CSCS="GSM" ดังตัวอย่าง (เราสามารถตรวจสอบค่าทั้ง 3 นี้ว่าถูกต้องหรือไม่โดยใช้คำสั่ง AT+CMGF? , AT+CSMP? และ AT+CSCS? ถ้าค่าถูกต้องอยู่แล้วก็ไม่ต้องกำหนดใหม่)

```
AT+CMGF=1
OK
AT+CSMP=17,167,0,0
OK
AT+CSCS="GSM"
OK
```

ในการส่งข้อความ SMS นั้นจะใช้คำสั่ง AT+CMGS ในการสั่งงาน โดยในกรณีที่ใช้ Text Mode นั้นให้ใช้รูปแบบคำสั่งเป็น AT+CMGS="**+เบอร์ผู้รับ**" โดยเบอร์ของผู้รับต้องใส่รหัสประเทศนำหน้าแทนศูนย์ด้วยเสมอ ซึ่งในกรณีที่เป็นประเทศไทยจะใช้รหัสประเทศเป็น "66" ดังนั้นถ้าต้องการส่งข้อความ SMS ให้กับเบอร์ที่ใช้งานอยู่ในประเทศไทย เช่น 089-4469XXX ก็จะต้องกำหนดหมายเลขของเบอร์ผู้รับปลายทางเป็น 6689-4469XXX แทน ซึ่งในกรณีนี้จะได้รหัสเบอร์ผู้รับข้อความเป็น "+66894469XXX" ซึ่งเมื่อโมดูล UC20G ได้รับคำสั่ง AT+CMGS เรียบร้อยแล้วมันจะตอบรับด้วยการส่งเครื่องหมาย ">" กลับมาบอก ซึ่งหลังจากนี้เป็นต้นไปผู้ใช้ก็สามารถจะทำการพิมพ์ข้อความต่างๆที่ต้องการจะส่งให้กับโมดูลได้ทันที โดยให้ปิดท้ายข้อความด้วยการกดปุ่ม Ctrl+Z (0x1A) เช่นถ้าต้องการส่งข้อความ SMS ให้กับหมายเลข 0894469XXX ด้วยข้อความ "Hello Test SMS" จะเป็นดังนี้

```
AT+CMGS="+66894469XXX"          // เบอร์ที่ต้องการส่งข้อความ
> Hello Test SMS                // พิมพ์ข้อความที่ต้องการส่งตามด้วยกด <Ctrl+Z>
+CMGS: 5
```

OK

ในกรณีที่พิมพ์คำสั่ง AT+CMGS="+66894469XXX" แล้วข้อความตอบกลับมาว่า ERROR แสดงว่าพิมพ์คำสั่งผิดหรือไม่ได้ตั้งค่ากำหนดรูปแบบของข้อความเป็น Text Mode ให้ตรวจสอบโดยใช้คำสั่ง AT+CMGF? ถ้าข้อความตอบกลับมาเป็น +CMGF: 0 แสดงว่ายังไม่ได้ตั้งค่า ให้ใช้คำสั่ง AT+CMGF=1 ตามด้วย Enter เพื่อกำหนดรูปแบบของข้อความเป็น Text Mode

16. รหัสข้อความ SMS ภาษาไทย

สำหรับข้อความ SMS ที่เป็นภาษาไทยนั้น จะไม่สามารถแสดงผลด้วยโปรแกรม Terminal ปรกติได้ ทั้งนี้ก็เนื่องมาจากว่าระบบตัวอักษรที่ใช้ในโปรแกรม Terminal นั้นจะใช้รหัส ASCII ปรกติที่มีขนาดเพียง 1 ไบท์ แต่สำหรับรหัสภาษาไทยที่ใช้ในระบบสื่อสารของโทรศัพท์มือถือต่างๆนั้น จะใช้รหัสพิเศษเฉพาะที่เรียกว่า “Unicode” ซึ่งตัวอักษร 1 ตัวจะประกอบไปด้วยข้อมูลจำนวน 2 ไบท์ โดยรหัส Unicode ของภาษาไทยนั้นจะมีค่าอยู่ระหว่าง 0E00H...0E7FH สำหรับภาษาอังกฤษนั้นถ้าเป็น Unicode จะใช้รหัสตัวอักษรขนาด 2 Byte เช่นเดียวกันกับภาษาไทย โดยจะมีค่าอยู่ระหว่าง 0000H..007FH โดยตามปรกติแล้วถ้าข้อความเป็นภาษาอังกฤษอย่างเดียวรหัสของตัวอักษรที่ใช้ใน SMS จะเป็นแบบ ASCII คือ ใช้รหัส ขนาด 1 ไบท์ โดยตัดรหัส 00H ไบท์แรกใน Unicode ทิ้งไป เช่น A แทนที่จะเป็นรหัส 0041H ก็จะเหลือเพียง 41H เป็นต้น

	┌	┐	└	┘	│	─	●	◘						♫	☼
0000	0001	0002	0003	0004	0005	0006	0007	0008	0009	000A	000B	000C	000D	000E	000F
┼	◄	↕	‼	¶	┴	┬	┤	↑	├	→	←				
0010	0011	0012	0013	0014	0015	0016	0017	0018	0019	001A	001B	001C	001D	001E	001F
	!	“	#	$	%	&	‘	(	)	*	+	,	-	.	/
0020	0021	0022	0023	0024	0025	0026	0027	0028	0029	002A	002B	002C	002D	002E	002F
0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0030	0031	0032	0033	0034	0035	0036	0037	0038	0039	003A	003B	003C	003D	003E	003F
@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0040	0041	0042	0043	0044	0045	0046	0047	0048	0049	004A	004B	004C	004D	004E	004F
P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
0050	0051	0052	0053	0054	0055	0056	0057	0058	0059	005A	005B	005C	005D	005E	005F
`	a	B	c	d	e	f	g	h	i	j	k	l	m	n	o

0060	0061	0062	0063	0064	0065	0066	0067	0068	0069	006A	006B	006C	006D	006E	006F
p	q	R	s	t	u	v	w	x	y	z	{	|	}	~	
0070	0071	0072	0073	0074	0075	0076	0077	0078	0079	007A	007B	007C	007D	007E	007F

ตาราง แสดงรหัส Unicode ภาษาอังกฤษ

	!	“	#	$	%	&	‘	(	)	*	+	,	-	.	/
20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F
0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F
@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F
P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F
`	a	B	c	d	e	f	g	h	i	j	k	I	m	n	o
60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F
p	q	R	s	t	u	v	w	x	y	z	{	|	}	~	
70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F

ตาราง แสดงรหัส ASCII ภาษาอังกฤษ

	ก	ข	ฃ	ค	ฅ	ฆ	ง	จ	ฉ	ช	ซ	ฌ	ญ	ฎ	ฏ
0E00	0E01	0E02	0E03	0E04	0E05	0E06	0E07	0E08	0E09	0E0A	0E0B	0E0C	0E0D	0E0E	0E0F
ฐ	ฑ	ฒ	ณ	ด	ต	ถ	ท	ธ	น	บ	ป	ผ	ฝ	พ	ฟ
0E10	0E11	0E12	0E13	0E14	0E15	0E16	0E17	0E18	0E19	0E1A	0E1B	0E1C	0E1D	0E1E	0E1F
ภ	ม	ย	ร	ฤ	ล	ฦ	ว	ศ	ษ	ส	ห	ฬ	อ	ฮ	ฯ
0E20	0E21	0E22	0E23	0E24	0E25	0E26	0E27	0E28	0E29	0E2A	0E2B	0E2C	0E2D	0E2E	0E2F
ะ	ั	า	ำ	ิ	ี	ึ	ื	ุ	ู	ฺ					฿
0E30	0E31	0E32	0E33	0E34	0E35	0E36	0E37	0E38	0E39	0E3A	0E3B	0E3C	0E3D	0E3E	0E3F
เ	แ	โ	ใ	ไ	ๅ	ๆ	็	่	้	๊	๋	์	๎	๏	๏
0E40	0E41	0E42	0E43	0E44	0E45	0E46	0E47	0E48	0E49	0E4A	0E4B	0E4C	0E4D	0E4E	0E4F
๐	๑	๒	๓	๔	๕	๖	๗	๘	๙	๚	๛				
0E50	0E51	0E52	0E53	0E54	0E55	0E56	0E57	0E58	0E59	0E5A	0E5B	0E5C	0E5D	0E5E	0E5F
0E60	0E61	0E62	0E63	0E64	0E65	0E66	0E67	0E68	0E69	0E6A	0E6B	0E6C	0E6D	0E6E	0E6F

0E70	0E71	0E72	0E73	0E74	0E75	0E76	0E77	0E78	0E79	0E7A	0E7B	0E7C	0E7D	0E7E	0E7F

ตาราง แสดงรหัส Unicode ภาษาไทย

17. หลักการถอดรหัสตัวอักษร Unicode

สำหรับรหัสตัวอักษรที่เป็น Unicode นั้น จะเห็นได้ว่าแต่ละตัวอักษรจะประกอบไปด้วยรหัส Code จำนวน 2 ไบท์เสมอ โดยตัวแรกเป็นตัวบอกรหัส Table ว่าเป็น Unicode ของภาษาใด โดยถ้าเป็นรหัส Unicode ของภาษาอังกฤษ ไบท์แรกจะมีค่าเป็น 00H ส่วนไบท์ที่2 จะเป็นรหัสตัวอักษร ซึ่งมีค่าตรงกันกับรหัส ASCII ส่วนภาษาไทยนั้น ไบท์แรกจะมีค่ารหัสเป็น 0EH ส่วนไบท์ที่ 2 จะเป็นรหัสตัวอักษร ซึ่งจากการทดสอบรับข้อความรหัสตัวอักษรจาก SMS พบว่า ถ้าใช้ภาษาอังกฤษอย่างเดียว รหัสของตัวอักษรจะเป็นแบบรหัส ASCII คือ 1 ตัวอักษร จะมีรหัส 1 ไบท์ แต่เมื่อมีการใช้ข้อความที่มีทั้งภาษาไทยและภาษาอังกฤษรวมกันพบว่ามีการเข้ารหัสตัวอักษรภาษาอังกฤษเป็นแบบ Unicode ด้วย

ดังนั้นจึงน่าจะพอสรุปได้ว่า ถ้าใช้ข้อความที่เป็นภาษาไทย ในระบบ SMS จะใช้รหัสตัวอักษรที่เป็นแบบ Unicode เสมอ แต่สำหรับภาษาอังกฤษนั้น ในระบบโทรศัพท์จะสามารถเลือกใช้ได้ทั้งระบบ Unicode และ ASCII Code โดยถ้าเป็น Unicode จะใช้รหัสตัวอักษรขนาด 2 Byte เช่นเดียวกันกับภาษาไทย โดยจะมีค่าอยู่ระหว่าง 0000H..007FH โดยมีรหัส 00H เป็นข้อมูลไบท์แรก ซึ่งถ้าข้อความเป็นภาษาอังกฤษอย่างเดียวรหัสของตัวอักษรที่ใช้ใน SMS จะเป็นแบบ ASCII คือ ใช้รหัส ขนาด 1 ไบท์ โดยตัดรหัส 00H ไบท์แรกใน Unicode ทิ้งไป เช่น A แทนที่จะเป็นรหัส 0041H ก็จะเหลือเพียง 41H เป็นต้น แต่สำหรับข้อความที่มีทั้งภาษาไทยและภาษาอังกฤษรวมกันพบว่ามีการเข้ารหัส Code ตัวอักษรเป็นแบบ Unicode ด้วยเช่นเดียวกันกับภาษาไทย

ดังนั้นในการถอดรหัสตัวอักษรต้องพิจารณาถึงจุดนี้ด้วย โดยมีข้อสังเกตุว่า ถ้าพบรหัสตัวอักษรที่มีค่าระหว่าง 20H-7FH แสดงว่าเป็นรหัสแบบ ASCII สามารถนำไปแสดงผลได้เลย แต่ถ้าพบว่ารหัสเป็น 00H แสดงว่าเป็นรหัสแบบ Unicode ภาษาอังกฤษ ซึ่งรหัส Code ที่เป็นรหัสตัวอักษรจะอยู่ในรหัสข้อมูลไบท์ถัดไป และถ้าพบรหัสเป็น 0EH แสดงว่าเป็นรหัส Unicode ภาษาไทย ซึ่งรหัส Code ที่เป็นรหัสตัวอักษรจะอยู่ในข้อมูลไบท์ถัดไป เช่นเดียวกัน

ตัวอย่างเช่น ถ้าเราทดลองส่งข้อความ SMS ด้วยคำว่า “สวัสดี Jack” ไปให้กับโมดูล UC20G และสมมุติว่า UC20G รับข้อความนั้นไว้ และจัดเก็บไว้เป็นข้อความที่1 ถ้าใช้โปรแกรม Hyper Terminal หรือ Terminal อื่นๆที่แสดงผลเป็น ASCII จะได้รับการรายงานผลดังรูป

```
+CMTI: "SM",1
```

แต่เมื่อแสดงผลของข้อมูลที่รับได้ในรูปแบบของ HEX String จะพบว่ามีข้อมูลที่รับได้มากกว่าที่มองเห็นจากหน้าจอของโปรแกรม Hyper-Terminal ทั้งนี้ก็เนื่องมาจากว่าโปรแกรม Terminal จะแสดงผลข้อมูลที่รับได้เฉพาะในส่วนของรหัส ASCII (20H..FFH) เท่านั้น ส่วนรหัสที่ต่ำกว่า 20H (00H-1FH) โปรแกรม Hyper-Terminal จะถือว่าเป็นคำสั่ง เช่น 0DH,0AH จะไม่ถูกนำมาแสดงผล แต่จะถือเป็นคำสั่งให้เลื่อน Cursor มาไว้ในตำแหน่งเริ่มต้นของบรรทัดและขึ้นบรรทัดใหม่เป็นต้น ซึ่งในที่นี้ผู้เขียนจะขอแสดงข้อมูลที่รับได้ในรูปแบบของ Hex String แทน เช่นเมื่อรับรหัส ASCII ของตัว “A” ได้จะแสดงค่าเป็น “41” แทน โดยจะแสดง HEX String ไว้ทางด้านซ้าย และ จะแสดงรหัส ASCII ไว้ทางด้านขวาเพื่อเปรียบเทียบให้เห็น เพื่อให้ผู้อ่านสามารถเข้าใจรูปแบบได้ดีขึ้น โดยจากข้อความ +CMTI: “SM”,1 ที่เรามองเห็นจากหน้าจอของโปรแกรม Hyper-Terminal เมื่อนำมาแสดงให้เห็นในรูปแบบของ HEX String จะได้ผลดังนี้

```
0D 0A                                                 ..
2B 43 4D 54 49 3A 20 22 53 4D 22 2C 31 0D 0A          +CMTI: "SM",1..
```

จากผลข้อความ +CMTI: “SM”,1 หมายถึง มีข้อความส่งเข้ามาใหม่และถูกเก็บไว้ในหน่วยความจำโดยเป็นข้อความลำดับที่1 ซึ่งเราสามารถสั่งอ่านข้อความได้โดยใช้คำสั่ง AT+CMGR=1 ดังตัวอย่าง

```
41 54 2B 43 4D 47 52 3D 31 0D                         AT+CMGR=1.
```

โดยเมื่อได้รับคำสั่ง AT+CMGR=1 โมดูล UC20G จะแสดงข้อความลำดับที่1 ให้ทราบโดยมีรูปแบบ

```
+CMGR: "REC UNREAD","+66811234567",,"07/11/22,10:21:37+28"
<...ข้อความที่รับได้...>
```

+CMGR: คือผลตอบรับการสั่งอ่านข้อความ

“REC UNREAD” คือสถานะของข้อความ โดย REC UNREAD หมายถึงข้อความที่ยังไม่เคยถูกสั่งอ่านมาก่อนแต่ถ้าเป็นข้อความที่เคยถูกสั่งอ่านมาแล้วจะมีสถานะเป็น REC READ

“+66811234567” คือหมายเลขโทรศัพท์ของผู้ส่ง ซึ่งเป็นหมายเลขในประเทศไทย คือ 0811234567

“07/11/22,10:21:37+28" คือ วันเวลาที่รับข้อความ

ซึ่งจากตัวอย่างการทดลองถ้าแสดงผลข้อมูลที่รับได้จากโมดูลในรูปแบบ HEX String ผลการสั่งอ่านข้อความจะได้ผลดังตัวอย่าง

```
0D 0A                                                 ..
2B 43 4D 47 52 3A 20 22 52 45 43 20 55 4E 52 45       +CMGR: "REC UNRE
41 44 22 2C 22 2B 36 36 38 31 31 32 33 34 35 36       AD","+6681123456
37 22 2C 2C 22 30 37 2F 31 31 2F 32 32 2C 31 30       7",,"07/11/22,10
3A 32 31 3A 33 37 2B 32 38 22 0D 0A                   :21:37+28"..
0E 2A 0E 27 0E 31 0E 2A 0E 14 0E 35 00 20 00 4A       .*.'.1.*...5. .J
00 61 00 63 00 6B 0D 0A                               .a.c.k..
```

```
0D 0A                                                    ..
4F 4B 0D 0A                                              OK..
```

เมื่อลองพิจารณาถึงรหัสส่วนที่เป็นข้อความจะเห็นได้ว่า รหัสของข้อความทั้งหมดจะเป็นรหัสแบบ Unicode โดยอักษรตัวแรกจะเป็น 0E 2A และตัวสุดท้ายจะเป็น 00 6B ซึ่งเมื่อถอดรหัสข้อความดูจะได้ว่า

0EH 2AH = รหัส Unicode ของตัวอักษร ส

0EH 27H = รหัส Unicode ของตัวอักษร ว

0EH 31H = รหัส Unicode ของตัวอักษร

0EH 2AH = รหัส Unicode ของตัวอักษร ส

0EH 14H = รหัส Unicode ของตัวอักษร ด

0EH 35H = รหัส Unicode ของตัวอักษร

00H 20H = รหัส Unicode ของตัวอักษรภาษาอังกฤษ Space

00H 4AH = รหัส Unicode ของตัวอักษรภาษาอังกฤษ J

00H 61H = รหัส Unicode ของตัวอักษรภาษาอังกฤษ a

00H 63H = รหัส Unicode ของตัวอักษรภาษาอังกฤษ c

00H 6BH = รหัส Unicode ของตัวอักษรภาษาอังกฤษ k

www.etteam.com

แต่ถ้าเราทดลองส่งข้อความ SMS ที่เป็นภาษาอังกฤษเพียงอย่างเดียว เช่น “Hello Jack” ไปให้กับโมดูล UC20G และสมมุติว่า UC20G รับข้อความนั้นไว้ และจัดเก็บไว้เป็นข้อความที่2 ถ้าใช้โปรแกรม Hyper Terminal หรือ Terminal อื่นๆที่แสดงผลเป็น ASCII จะได้รับการรายงานผลดังรูป

```
+CMTI: "SM",2
```

โดยเมื่อแสดงผลด้วยรูปแบบการแสดงผลแบบ HEX String จะได้ผลดังรูป

```
   0D 0A                                                 ..
2B 43 4D 54 49 3A 20 22 53 4D 22 2C 32 0D 0A          +CMTI: "SM",2..
```

จากผลข้อความ +CMTI: “SM”,2 หมายถึง มีข้อความส่งเข้ามาใหม่และถูกเก็บไว้ในหน่วยความจำโดยเป็นข้อความลำดับที่2 ซึ่งเราสามารถสั่งอ่านข้อความได้โดยใช้คำสั่ง AT+CMGR=2 ดังตัวอย่าง

```
41 54 2B 43 4D 47 52 3D 32 0D                          AT+CMGR=2.
```

ซึ่งจากตัวอย่างการทดลองถ้าแสดงผลข้อมูลที่รับได้จากโมดูลในรูปแบบ HEX String ผลการสั่งอ่านข้อความจะได้ผลดังตัวอย่าง

```
    0D 0A                                                    ..
 2B 43 4D 47 52 3A 20 22 52 45 43 20 55 4E 52 45      +CMGR: "REC UNRE
 41 44 22 2C 22 2B 36 36 38 31 31 32 33 34 35 36      AD","+6681123456
 37 22 2C 2C 22 30 37 2F 31 31 2F 32 32 2C 31 31      7",,"07/11/22,11
 3A 33 34 3A 30 36 2B 32 38 22 0D 0A                  :34:06+28"..
 48 65 6C 6C 6F 20 4A 61 63 6B 0D 0A                  Hello Jack..
 0D 0A                                                ..
 4F 4B 0D 0A                                          OK..
```

ซึ่งจะเห็นได้ว่ารหัสของข้อความใน SMS จะเป็นแบบ ASCII ปรกติ โดยแต่ละตัวอักษรจะใช้รหัสขนาด 1 Byte ดังนี้

48H = รหัส ASCII ของ H

65H = รหัส ASCII ของ e

6CH = รหัส ASCII ของ l

6FH = รหัส ASCII ของ o

20H = รหัส ASCII ของ Space

4AH = รหัส ASCII ของ J

61H = รหัส ASCII ของ a

63H = รหัส ASCII ของ c

6BH = รหัส ASCII ของ k

18. การส่งข้อความ SMS ภาษาไทย

ในการส่ง SMS ในโหมดนี้จำเป็นต้องกำหนดเบอร์ SMS Service Center ซึ่งแต่ละเครือข่ายจะมีเบอร์ดังนี้

TRUE = +66891009120

DTAC = +66816110400

AIS = +66818110888 1-2-CALL = +66818310808

เราสามารถตรวจสอบเบอร์ SMS Service Center ได้โดยใช้คำสั่ง AT+CSCA? ตามด้วย Enter ถ้าเบอร์ถูกต้องตามเครือข่ายที่ใช้แล้วก็ไม่จำเป็นต้องตั้งค่าใหม่ แต่ถ้าไม่ตรงก็ให้ตั้งค่าใหม่โดยใช้คำสั่ง AT+CSCA ดังตัวอย่าง

จากตัวอย่างนี้ตั้งค่าเป็นของระบบ 1-2-CALL = +66818310808 ส่งข้อความ “**กขค**” ให้เบอร์ +66811234567 โดยก่อนการส่งต้องแปลงค่าต่างๆ เป็น Unicode ก่อน โดยดูจากตาราง

1. SMS Service Center **ของ** 1-2-CALL เบอร์ +66818310808 เมื่อดูในตารางจะได้ค่าดังนี้ คำสั่งที่ใช้ส่งคือ AT+CSCA โดยปกติค่า SMS Service Center จะมีอยู่ใน SIM Card อยู่แล้ว ถ้าการส่ง SMS ไม่มีปัญหา ก็ไม่จำเป็นต้องแก้ไข ให้ข้ามคำสั่งนี้ไปได้เลย

+ = 002B
6 = 0036
6 = 0036
8 = 0038
1 = 0031
8 = 0038
3 = 0033
1 = 0031
0 = 0030
8 = 0038
0 = 0030
8 = 0038

2. **เบอร์โทรปลายทางของผู้รับข้อความ เบอร์** +66811234567 (ให้เปลี่ยนตามเบอร์ที่ใช้งานจริงของผู้ใช้) เมื่อดูในตารางจะได้ค่าดังนี้ คำสั่งที่ใช้ส่งคือ AT+CMGS

+ = 002B
6 = 0036
6 = 0036
8 = 0038
1 = 0031
1 = 0031
2 = 0032
3 = 0033
4 = 0034
5 = 0035
6 = 0036
7 = 0037

3. **ข้อความภาษาไทยที่ต้องการจะส่ง “กขค”** เมื่อดูในตารางจะได้ค่าดังนี้

ก = 0E01

ข = 0E02

ค = 0E04

เมื่อได้ค่าต่างแล้วก็ทดลองส่งคำสั่งดังตัวอย่าง

```
AT+CMGF=1
OK
AT+CSCS="UCS2"
OK
AT+CSCA="002B00360036003800310038003300310030003800300038",145  // ไม่จำเป็นต้องใส่
OK
AT+CSMP=17,167,0,8
OK
AT+CMGS="002B00360036003800310031003200330034003500360037"
>0E010E020E04                                    // ข้อความภาษาไทยที่จะส่งตามด้วยกด <Ctrl+Z>
+GMGS: 8

OK
```

19. การส่งข้อความ MMS (Multimedia Messaging Service)

โมดูล UC15 มีความสามารถในการส่ง MMS ไม่ว่าจะเป็น รูปภาพ,เสียง หรือข้อความต่างๆ ถ้าผู้ใช้ต้องการดูข้อมูลเพิ่มเติมให้ศึกษาจากเอกสาร Quectel_UC15_MMS_AT_Commands_Manual_V1.0.pdf ซึ่งตัวอย่างนี้จะแสดงตัวอย่างเบื้องต้นเท่านั้น

ในการส่ง MMS นั้นผู้ใช้จำเป็นต้องรู้ค่า APN ของเครือข่ายที่ใช้ด้วย โดยในตัวอย่างนี้จะใช้เครือข่ายของ truemove H ซึ่งมีค่าพารามิเตอร์ต่างๆ ดังนี้

การตั้งค่า APN สำหรับ MMS ของ truemove H

Name = TRUE-H MMS

APN = hmms

Username = true

Password = true

MMSC = http://mms.trueh.com:8002/

MMS Proxy = 010.004.007.039

MMS Port = 8080

MCC = 520

MNC = 00

APN type = mms

โดยขั้นตอนของการส่งข้อความ MMS มีดังนี้

1. ทำการตั้งค่า APN และเชื่อมต่อข้อมูล

```
AT+QICSGP=1,1,"hmms","true","true",1          // ใส่ค่า APN,Username และ Password
OK
AT+QIACT=1                                    // Activate PDP context 1
OK
AT+QMMSCFG="contextid",1
OK
```

2. ตั้งค่า URL และ proxy

```
AT+QMMSCFG="mmsc","http://mms.trueh.com:8002/"        // ใส่ค่า MMSC URL
OK
AT+QMMSCFG="proxy","010.004.007.039",8080             // ใส่ค่า MMS Proxy และ MMS Port
OK
```

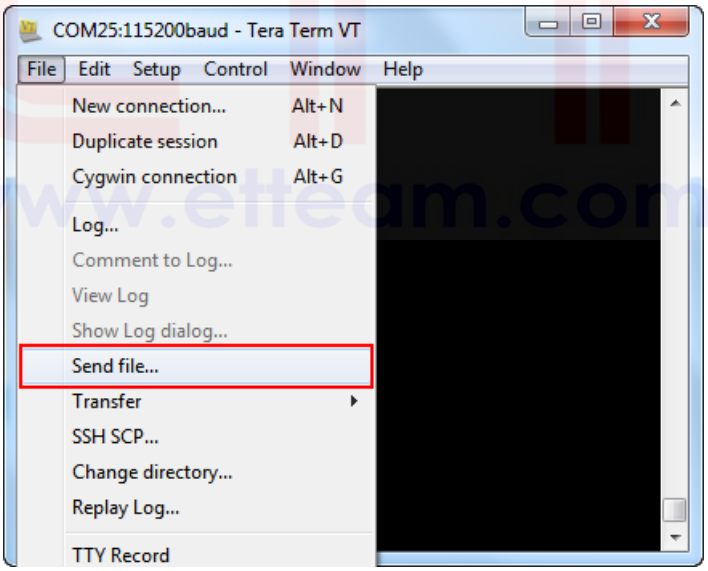
3. กำหนดพารามิเตอร์ที่ใช้ในการส่ง

```
AT+QMMSCFG="sendparam",6,2,0,0,2,4
OK
```

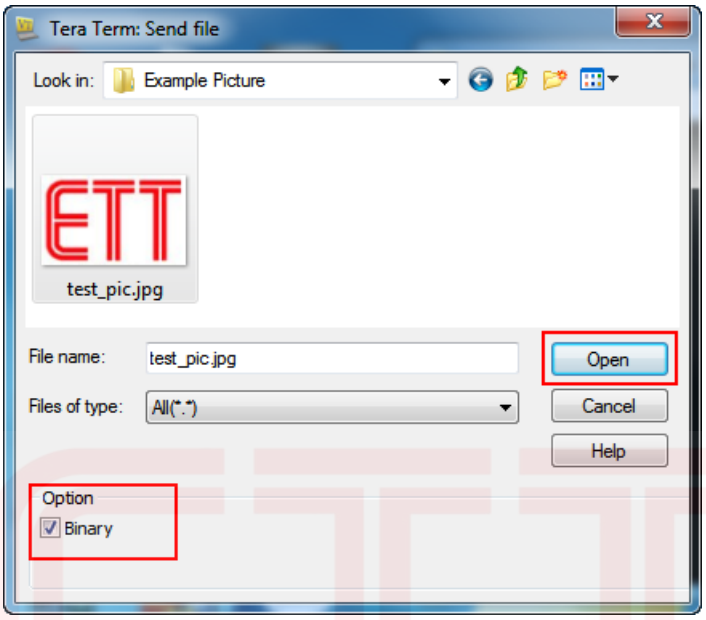
4. เริ่มต้นการเขียนข้อความที่จะส่ง ซึ่งในตัวอย่างนี้จะทำการส่งข้อความ MMS ชื่อ test uc20 mms โดยจะส่งรูปภาพชื่อ test_pic.jpg ซึ่งมีขนาดไฟล์ 6,188 bytes ไปที่เบอร์ปลายทาง 089XXXXXXX

```
AT+QMMSEDIT=1,1,"089XXXXXXX"                    // เบอร์ปลายทางที่จะส่ง
OK
AT+QMMSCFG="character","ASCII"                  // กำหนดรูปแบบข้อความชื่อเป็น ASCII
OK
AT+QMMSEDIT=4,1,"test uc15 mms"                 // ชื่อข้อความ
OK
AT+QFUPL="RAM:test_pic.jpg",6188,300,1          // อัพโหลดไฟล์ภาพเข้าสู่ RAM ของโมดูล
CONNECT                                         // รอรับไฟล์
```

หลังจากส่งคำสั่งอัพโหลดไฟล์เข้าสู่ RAM ของโมดูลจะมีข้อความ CONNECT ตอบกลับมาเพื่อให้เริ่มส่งไฟล์ ให้ไปที่เมนู File เลือก Send file… ดังรูป



เลือก Option เป็น Binary และเลือกไฟล์ภาพที่ต้องการส่งจากนั้นคลิก Open เพื่อเริ่มส่งไฟล์ โดยการส่งนั้นต้องส่งให้เสร็จภายในเวลา timeout ที่กำหนดไว้ตอนอัพโหลดไฟล์ ซึ่งก็คือ 300 วินาที



เมื่ออัพโหลดรูปเสร็จเรียบร้อยจะมีข้อความว่าได้ทำการอัพโหลดไฟล์ขนาด 6,188 bytes

```
+QFUPL: 6188,c77d

OK
```

จากนั้นให้การแนบไฟล์ test_pic.jpg ที่อยู่ใน RAM เพื่อทำการส่งโดยใช้คำสั่ง

```
AT+QMMSEDIT=5,1,"RAM:test_pic.jpg"
OK
```

5. ทำการส่งข้อความ MMS

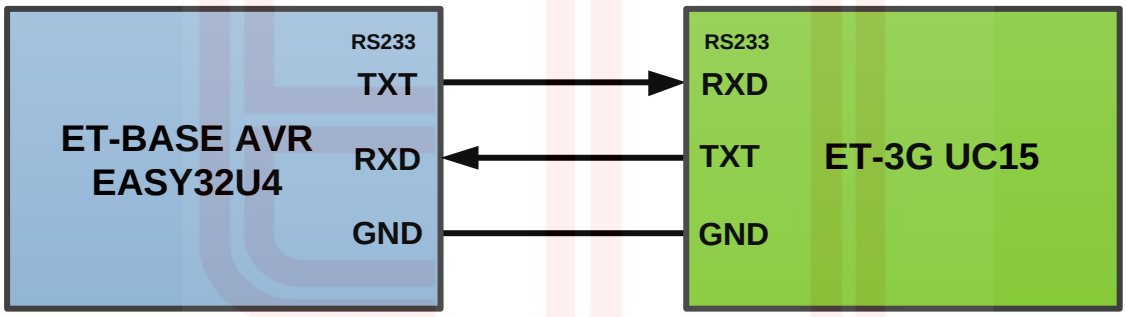
```
AT+QMMSEND=100                          // ทำการส่งข้อความ MMS
OK

+QMMSEND: 0,200                         // การส่งข้อความ MMS เสร็จเรียบร้อย
```

6. ทำการเคลียร์ข้อมูลของข้อความ MMS และ ลบข้อมูลรูปภาพที่อยู่ใน RAM ของโมดูล

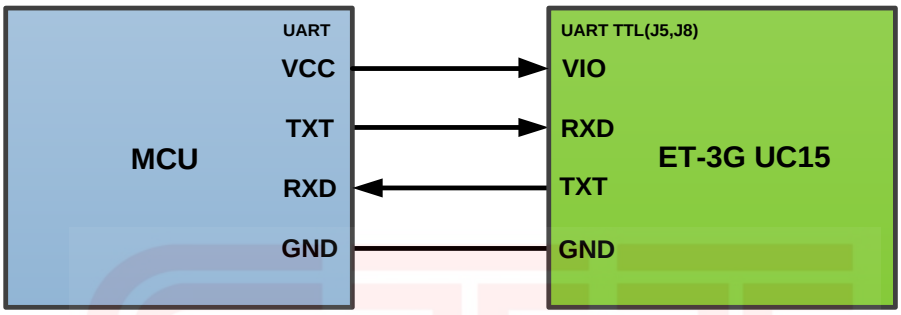
```
AT+QMMSEDIT=0                          // เคลียร์ข้อมูลของข้อความ MMS
OK
AT+QFDEL="RAM:test_pic.jpg"            // ลบข้อมูลรูปภาพที่อยู่ใน RAM
OK
```

20. การเชื่อมต่อบอร์ด ET-3G UC15 กับ บอร์ดไมโครคอนโทรลเลอร์

ตัวอย่างนี้จะขอยกตัวอย่างการเชื่อมต่อ ET-3G UC15 ร่วมกับบอร์ด ET-BASE AVR EASY32U4 ซึ่งจะตัวอย่างนี้จะเชื่อมต่อผ่านพอร์ต RS232 (เลือกจั้มเปอร์ JP1-JP4 ของบอร์ด ET-3G UC15 ไปที่ตำแหน่ง RS232) โดยไดอะแกรมการเชื่อมต่อ และรูปการเชื่อมต่อดังรูป



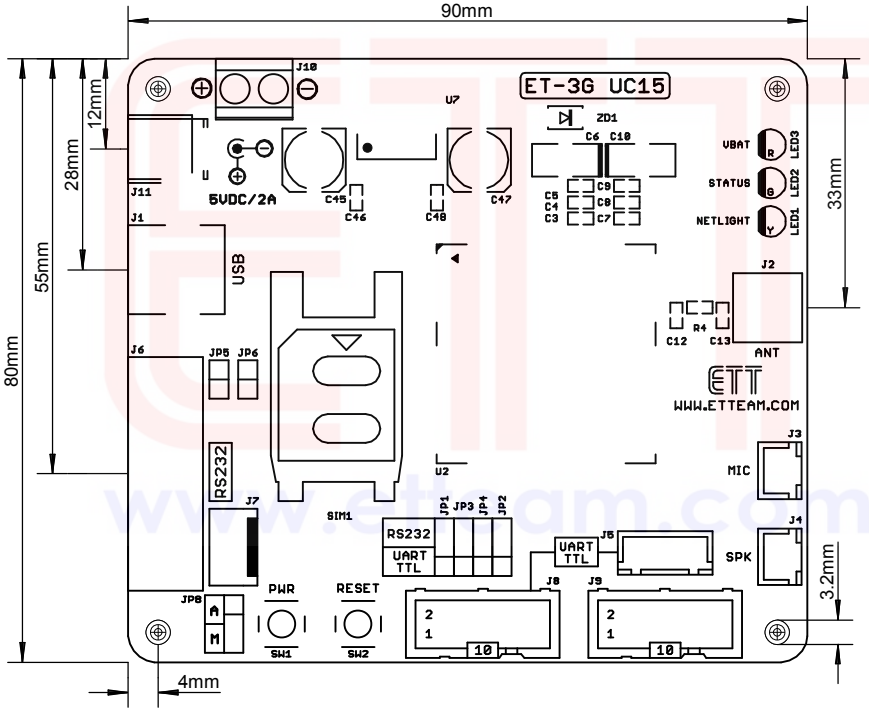
หมายเหตุ ในกรณีที่ผู้ใช้ต่อจากขาของไมโครคอนโทรลเลอร์โดยตรง โดยไม่ผ่านวงจร Line Driver RS232 ก็ให้เลือกจั้มเปอร์ JP1-JP4 ของบอร์ด ET-3G UC15 ไปที่ตำแหน่ง UART TTL และต้องเชื่อมต่อสัญญาณทางขั้ว TTL (J5,J8) เท่านั้น และจำเป็นต้องต่อแรงดันไฟเลี้ยงของไมโครคอนโทรลเลอร์ (VCC) ไปต่อที่ขา VIO ของบอร์ด ET-3G UC15 ด้วยเพื่อใช้เป็นแรงดันอ้างอิงให้กับวงจร Level Shifter ของบอร์ด ET-3G UC15 ด้วย

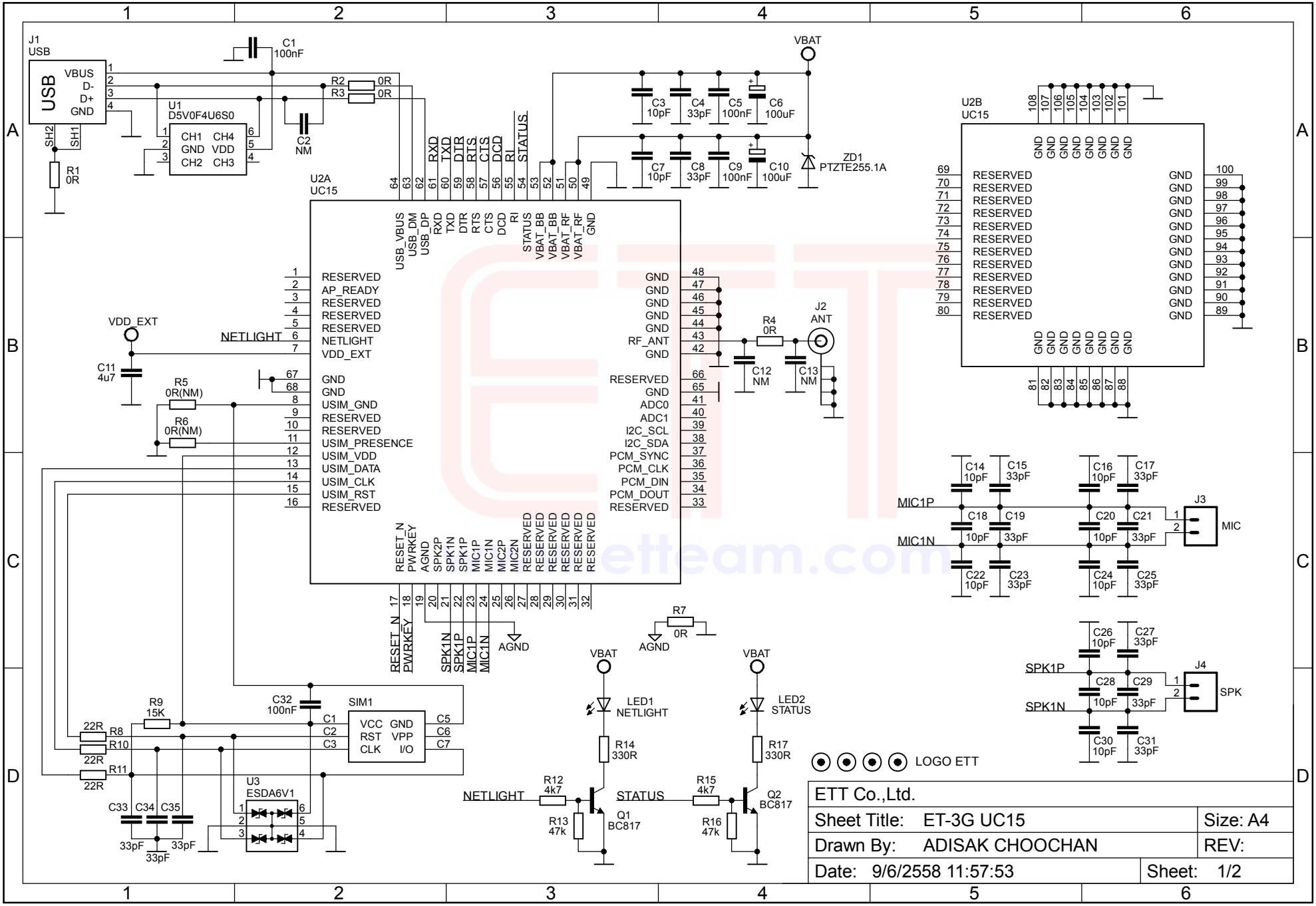


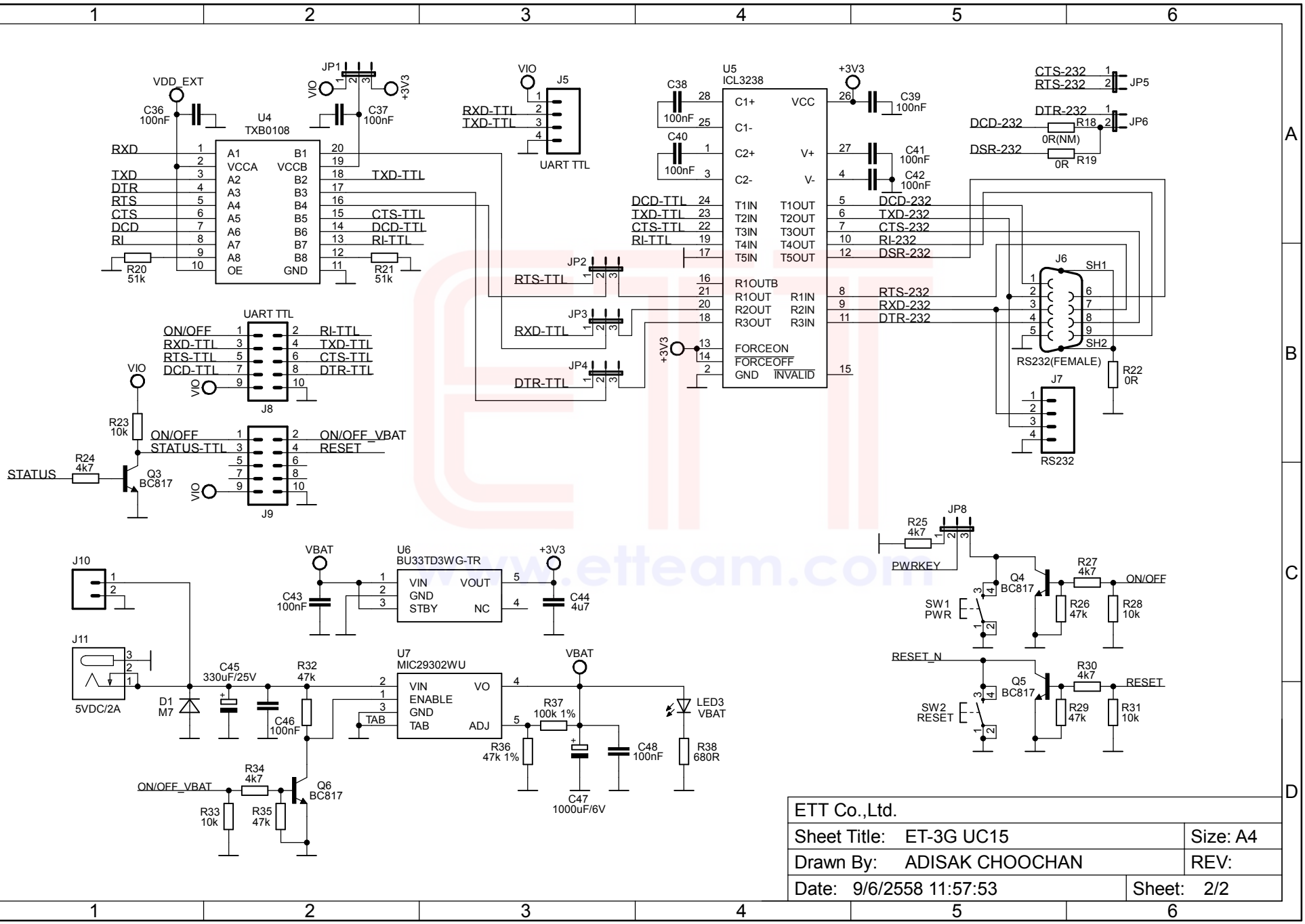
จากนั้นให้เปิดตัวอย่างโปรแกรม SMS_Send ดังรูป ซึ่งตัวอย่างนี้จะเป็นตัวอย่างส่งข้อความไปยังเบอร์ของผู้รับ ให้ทำการแก้ไขข้อความ XXXXXXXXXX เป็นเบอร์ที่ต้องการให้บอร์ด ET-3G UC20 ส่งข้อความไป จากนั้นให้ Verify และ Upload โปรแกรมที่แก้ไขแล้ว จากนั้นรอสักครู่จะมีข้อความ “Hello from ET-3G UC15” ไปปรากฏที่เบอร์ของผู้รับ

SMS_Send

```
/*
 * Examples  : Arduino Examples By....ETT CO.,LTD
 * Program   : SMS_Send
 * Software  : Arduino 1.6.0
 * Hardware  : ET-BASE AVR EASY32U4 + ET-3G U
 * Function  : Send SMS to your phone numb
 */

int led = 13;                          // LED D13
char phone_number[] = "XXXXXXXXXX";    // Your phone number  Example. 089xxxxxxx

void setup()
{
  Serial1.begin(115200);               // Set RS232 115200 bps
  delay(5000);

  Serial1.print("AT+CMGF=1\r");        // Select SMS message format to text mode
  CheckOK();                           // Check "OK"

  Serial1.print("AT+CSMP=17,167,0,0\r");        // Set SMS Text Mode Parameters
  CheckOK();                                    // Check "OK"

  Serial1.print("AT+CSCS=\"GSM\"\r");           // Select TE Character Set
  CheckOK();                                    // Check "OK"

  Serial1.print("AT+CMGS=\"");                  // Send to your phone number
  Serial1.print(phone_number);
  Serial1.print("\"\r");

  CheckReady();                                // Check ">"
  Serial1.print("Hello from ET-3G UC15");      // SMS body
  delay(100);
  Serial1.write(0x1A);                         // Ctrl+Z
}
```


90mm
80mm
55mm
28mm
12mm
33mm
3.2mm
4mm
ET-3G UC15
5VDC/2A
USB
RS232
SIM1
PWR
RESET
SW1
SW2
RS232
UART
TTL
SPK
MIC
ANT
VBAT
STATUS
NETLIGHT
WWW.ETTEAM.COM


J1
USB
C1
100nF
R2 0R
R3 0R
U1
D5V0F4U6S0
C2
NM
R1
0R
VBAT
C3 10pF
C4 33pF
C5 100nF
C6 100uF
C7 10pF
C8 33pF
C9 100nF
C10 100uF
ZD1
PTZTE255.1A
U2A
UC15
U2B
UC15
RESERVED
AP_READY
NETLIGHT
VDD_EXT
GND
USIM_GND
USIM_PRESENCE
USIM_VDD
USIM_DATA
USIM_CLK
USIM_RST
RF_ANT
ADC0
ADC1
I2C_SCL
I2C_SDA
PCM_SYNC
PCM_CLK
PCM_DIN
PCM_DOUT
R4 0R
J2
ANT
C12 NM
C13 NM
R5 0R(NM)
R6 0R(NM)
C11 4u7
C14 10pF
C15 33pF
C16 10pF
C17 33pF
C18 10pF
C19 33pF
C20 10pF
C21 33pF
C22 10pF
C23 33pF
C24 10pF
C25 33pF
J3
MIC
MIC1P
MIC1N
C26 10pF
C27 33pF
C28 10pF
C29 33pF
C30 10pF
C31 33pF
J4
SPK
SPK1P
SPK1N
R7 0R
AGND
C32 100nF
SIM1
R8 22R
R9 15K
R10 22R
R11 22R
C33 33pF
C34 33pF
C35 33pF
U3
ESDA6V1
LED1
NETLIGHT
LED2
STATUS
R14 330R
R17 330R
R12 4k7
R13 47k
R15 4k7
R16 47k
Q1
BC817
Q2
BC817
LOGO ETT
ETT Co.,Ltd.
Sheet Title: ET-3G UC15
Size: A4
Drawn By: ADISAK CHOOCHAN
REV:
Date: 9/6/2558 11:57:53
Sheet: 1/2



ETT Co.,Ltd.			
Sheet Title:	ET-3G UC15	Size: A4	
Drawn By:	ADISAK CHOOCHAN	REV:	
Date:	9/6/2558 11:57:53	Sheet:	2/2