Electronics Technology Team

# คู่มือการใช้งาน
# ET-BASE GSM SIM900

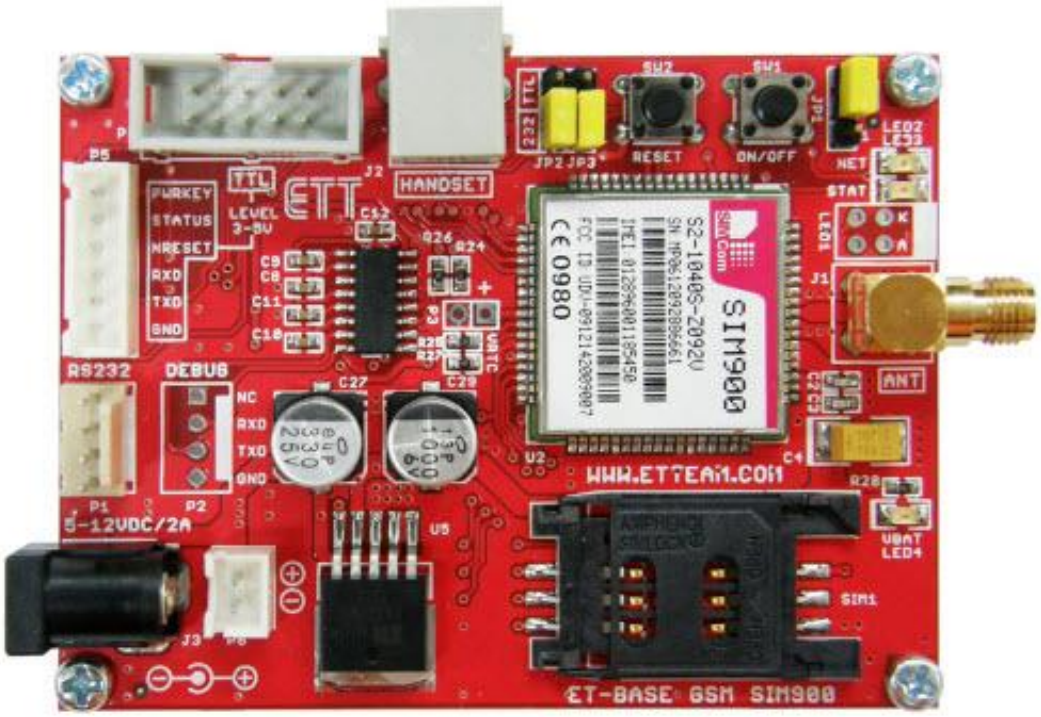
บริษัท อีทีที จำกัด
ETT CO., LTD.

http://www.etteam.com

# สารบัญ


1. คุณสมบัติของบอร์ด ET-BASE GSM SIM900 3
2. คุณสมบัติเบื้องต้นของโมดูล SIM900 3
3. ส่วนประกอบของบอร์ด ET-BASE GSM SIM900 4
4. การสั่ง เปิด และ ปิด การทำงานของโมดูล SIM900 6
5. การติดต่อสื่อสารกับโมดูล SIM900 7
6. คุณสมบัติการทำงานของสัญญาณที่ควรรู้ 9
7. ตัวอย่างการใช้งาน AT Command เพื่อสั่งงานโมดูล SIM900 10
8. การทดสอบการทำงานของบอร์ด ET-BASE GSM SIM900 11
9. ตัวอย่างการใช้งานคำสั่งเกี่ยวกับ RS232 14
10. การ Setup และตรวจสอบค่า Configuration 16
11. การตรวจสอบข้อมูลต่างๆ ของโมดูล SIM900 17
12. การโทรออก การรับสาย และ การยกเลิกการโทร 19
13. การตรวจสอบยอดเงินคงเหลือโดยใช้ USSD 20
14. การรับข้อความ SMS 20
15. การส่งข้อความ SMS ภาษาอังกฤษ 22
16. รหัสข้อความ SMS ภาษาไทย 23
17. หลักการถอดรหัสตัวอักษร Unicode 25
18. การส่ง SMS ภาษาไทย 28
19. การใช้ SIM Command (SIM Application Toolkit : STK) 30
20. การอ่านข้อมูลจากเว็บไซต์โดยใช้การเชื่อมต่อ GPRS (HTTP GET) 35
21. การเชื่อมต่อบอร์ด ET-BASE GSM SIM900 กับ บอร์ดไมโครคอนโทรลเลอร์ 37

# ET-BASE GSM SIM900




ET-BASE GSM SIM900 เป็นชุดเรียนรู้และพัฒนาระบบการสื่อสารไร้สาย โดยใช้โมดูล GSM/GPRS รุ่น SIM900 ของบริษัท SIMCom เป็นอุปกรณ์หลัก ซึ่ง SIM900 เป็นโมดูลสื่อสารระบบ GSM/GPRS ขนาดเล็ก รองรับระบบสื่อสาร GSM ความถี่ 850/900/1800/1900MHz โดยสั่งงานผ่านทางพอร์ตสื่อสารอนุกรม RS232 ด้วยชุดคำสั่ง AT Command สามารถประยุกต์ใช้งานได้มากมายหลายรูปแบบ ไม่ว่าจะเป็นการรับส่งสัญญาณแบบ Voice, SMS, Data, FAX และยังรวมถึงการสื่อสารด้วย Protocol TCP/IP ด้วย ซึ่งตามปรกติแล้ว ถึงแม้ว่าโมดูล SIM900 จะมีวงจร และ Firmware บรรจุไว้ภายในตัวเป็นที่เรียบร้อยแล้วก็ยังไม่สามารถนำไปใช้งานได้โดยตรงทันที เนื่องจากในการใช้งานจริงๆนั้น ผู้ใช้งานเองจำเป็นต้องออกแบบวงจรรอบนอกที่จำเป็นมาเชื่อมต่อกับขาสัญญาณของตัวโมดูลอีกในบางส่วน ไม่ว่าจะเป็นวงจรภาค Power Supply, วงจรเชื่อมต่อกับ SIM Card รวมไปถึงวงจร Line Driver ของ RS232 เป็นต้น ดังนั้นทางทีมงาน อีทีที จึงได้จัดสร้างบอร์ดสำหรับเป็นตัวกลางในการเชื่อมต่อระหว่างโมดูล SIM900 กับอุปกรณ์ภายนอกเพื่อให้ผู้ใช้งานสามารถนำโมดูล GSM ของ SIM900 ไปทำการทดลองและศึกษาเรียนรู้การสั่งงานต่างๆได้โดยสะดวก ก่อนที่จะนำเอาโมดูลตัวนี้ไปออกแบบดัดแปลงและประยุกต์ใช้งานในด้านต่างๆได้ต่อไปในอนาคต ซึ่งถึงแม้ว่าวงจรการเชื่อมต่อทั้งหมดที่ทาง อีทีที ได้จัดทำขึ้นมานี้จะยังไม่สามารถรองรับการใช้งานทรัพยากรต่างๆที่มีอยู่ภายในโมดูลได้ครบถ้วนทั้งหมดก็ตามที แต่ในส่วนของการใช้งานโมดูลในส่วนที่เป็นความสามารถหลักๆ ที่จำเป็นนั้นมีไว้รองรับอย่างครบถ้วนเพียงพอแล้ว

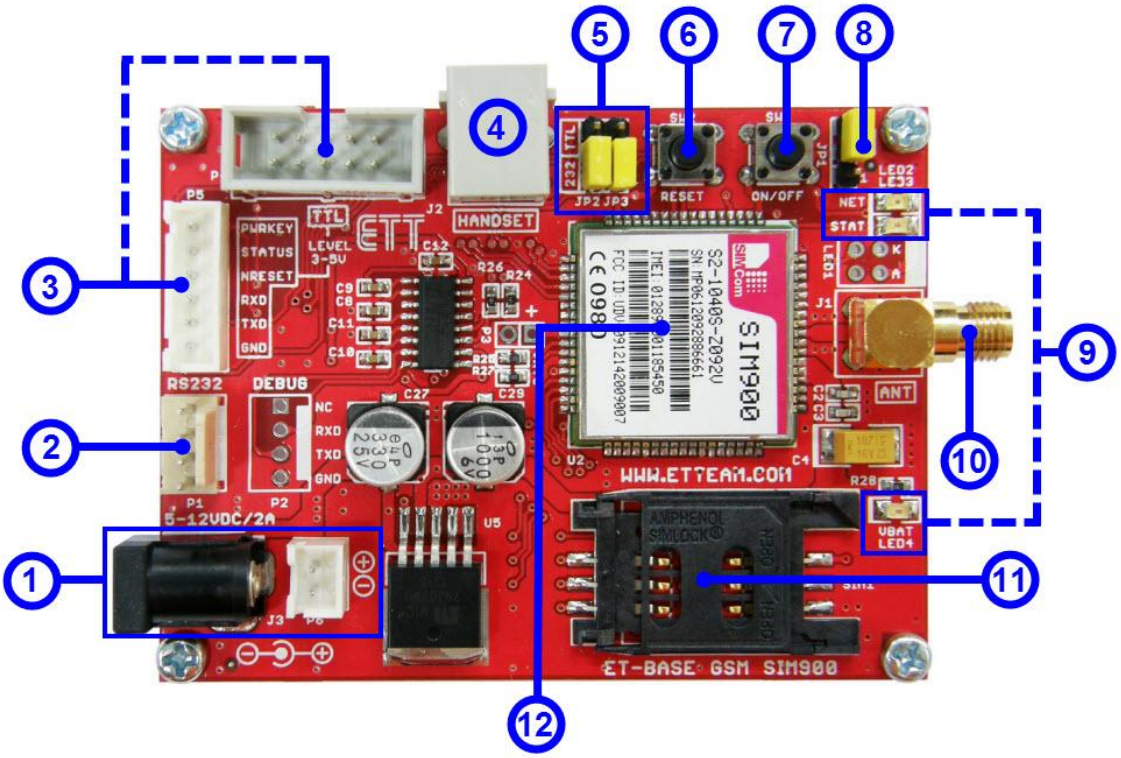
## 1. คุณสมบัติของบอร์ด ET-BASE GSM SIM900

- มีสวิตช์แบบ Push-Button สำหรับใช้สั่ง เปิด-ปิด การทำงานของโมดูลภายในบอร์ด
- มีสวิตช์แบบ Push-Button สำหรับใช้สั่ง RESET การทำงานของโมดูลภายในบอร์ด
- มี Socket SIM รองรับ SIM Card พร้อมวงจร ESD ป้องกัน SIM เสียหาย
- มีวงจร Regulate แยกอิสระ จำนวน 2 ชุด สามารถใช้กับแหล่งจ่ายภาย Adapter ตั้งแต่ 5-12 VDC สามารถจ่ายกระแสให้กับโมดูล SIM900 และอุปกรณ์เชื่อมต่อต่างๆได้อย่างเพียงพอ
    - มีวงจร Regulate ขนาด 4.2V / 3A สำหรับจ่ายให้กับโมดูล SIM900 ได้อย่างเพียงพอ สามารถใช้กับ SIM ของระบบ GSM900MHz แบบ 2-Watt ได้อย่างไม่เกิดปัญหา
    - มีวงจร Regulate ขนาด 2.8V / 150mA สำหรับจ่ายให้กับวงจรแปลงระดับสัญญาณโลจิก
- มีวงจร Line Driver สำหรับแปลงระดับสัญญาณโลจิกจากโมดูล SIM900 ให้เป็น RS232 (1200 bps-115200 bps) สำหรับพอร์ตที่ใช้ในการสื่อสารสำหรับสั่งงานโมดูล
- มีวงจรแปลงระดับสัญญาณโลจิก TTL ระดับแรงดัน 3V-5V ทำให้สามารถเชื่อมต่อกับไมโครคอนโทรลเลอร์ได้โดยตรงโดยไม่ต้องผ่านวงจร Line Driver RS232
- มี LED แสดงสถานะพร้อมในบอร์ด สำหรับแสดงสถานะของแหล่งจ่ายไฟ สถานะพร้อมทำงานของโมดูล สถานะในการเชื่อมต่อกับ Network และ สถานะ Power-ON/Power-OFF ของโมดูล
- มีขั้วสำหรับเชื่อมต่อกับ Handset (ชุดปากพูด และหูฟัง ของโทรศัพท์บ้าน) โดยใช้ขั้วต่อแบบ RJ11 มาตรฐาน พร้อมวงจร Voice Filter สามารถนำชุด Handset ของโทรศัพท์บ้าน ต่อเข้ากับบอร์ดทางขั้วต่อแบบ RJ11 สำหรับใช้พูดคุย โทรออก และ รับสายได้โดยสะดวก

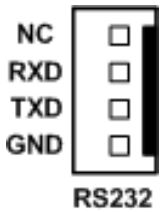
## 2. คุณสมบัติเบื้องต้นของโมดูล SIM900

- รองรับความถี่ GSM/GPRS 850/900/1800/1900MHz
- รองรับ GPRS Multi-Slot Class10 และ GPRS Mobile Station Class B
- รองรับมาตรฐานคำสั่ง AT Command (GSM 07.07 / 07.05 และคำสั่งเพิ่มเติมจาก SIMCOM)
- รองรับ SIM Applications Toolkit
- ทำงานที่ย่านแรงดัน 3.2V ถึง 4.8V
- รองรับการเชื่อมต่อภายนอก
    - ใช้ได้กับ SIM card 1.8V และ 3V
    - มีวงจร Analog Audio (MIC & Speaker)

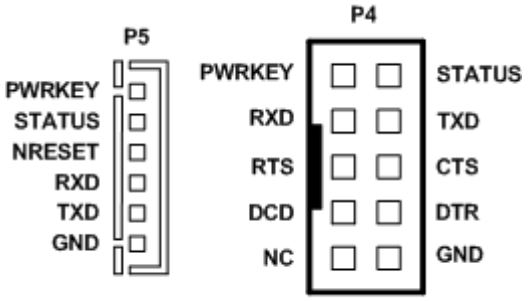
## 3. ส่วนประกอบของบอร์ด ET-BASE GSM SIM900




- **หมายเลข** 1 เป็นขั้วต่อไฟเลี้ยงเข้าบอร์ดโดยมีให้เลือกต่อ 2 แบบ คือ แบบ DC JACK ซึ่งขั้วด้านนอกเป็นไฟบวก ด้านในเป็นลบ และ ขั้วต่อแบบ JST โดยแรงดันไฟเลี้ยงที่จ่ายให้บอร์ดสามารถใช้ได้ตั้งแต่ 5-12 VDC กระแสอย่างน้อย 2 A
- **หมายเลข** 2 เป็นขั้วต่อสัญญาณ RS232 แบบ 4 PINS (มาตรฐานอีทีที) สำหรับเชื่อมต่อกับอุปกรณ์ที่ใช้การรับส่งข้อมูลด้วย RS232 เช่น คอมพิวเตอร์ หรือ บอร์ดไมโครคอนโทรลเลอร์ต่างๆ ที่ต่อผ่านวงจร Line Driver RS232 โดยการจัดตำแหน่งขาสัญญาณแสดงดังรูป




- **หมายเลข** 3 เป็นขั้วต่อสัญญาณระดับ TTL 3-5 V สำหรับเชื่อมต่อกับไมโครคอนโทรลเลอร์ได้โดยตรงโดยไม่ต้องผ่านวงจร Line Driver RS232 โดยการจัดตำแหน่งขาสัญญาณแสดงดังรูป




TTL LEVEL 3-5V

| ชื่อขาสัญญาณ | ทิศทาง | รายละเอียด |
|---|---|---|
| PWRKEY | INPUT | ใช้ควบคุมการเปิดปิดโมดูล SIM900 โดยจะทำงานที่ลอจิก 1 |
| STATUS | OUTPUT | ใช้บอกสถานะว่าโมดูล SIM900 เปิดการทำงานอยู่ ถ้าเป็นลอจิก 1 แสดงว่าโมดูลเปิดอยู่ ถ้าเป็นลอจิก 0 แสดงว่าแสดงว่าโมดูลปิดอยู่หรืออยู่สภาวะ power down |
| NRESET | INPUT | ใช้สำหรับรีเซ็ตการทำงานของโมดูล SIM900 โดยจะทำงานที่ลอจิก 1 |
| 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 |
| NC | - | ขาว่างไม่ได้ใช้งาน |
| GND | | กราวด์ |

- **หมายเลข** 4 เป็น ขั้วต่อ RJ11 สำหรับใช้เชื่อมต่อกับชุด Handset ในกรณีที่ต้องการใช้งานโมดูล SIM900 เพื่อโทรออกและรับสาย โดยสามารถเชื่อมต่อกับ Handset มาตรฐานได้ทั่วไป
- **หมายเลข** 5 เป็นจั้มเปอร์เลือกว่าจะต่อขาสัญญาณ RXD, TXD ของโมดูลผ่านวงจร Line Driver RS232 หรือไม่ ถ้าผู้ใช้ต้องการเชื่อมต่อผ่านขั้ว RS232 ก็ให้เลือกจั้มเปอร์ JP2 และ JP3 ไปที่ตำแหน่ง 232 แต่ถ้าต้องการเชื่อมต่อทางขั้ว TTL P4, P5 ก็ให้เลือกจั้มเปอร์ JP2 และ JP3 ไปที่ตำแหน่ง TTL
- **หมายเลข** 6 เป็น Switch Push-Button สำหรับใช้รีเซ็ตการทำงานของตัวโมดูล

- **หมายเลข** 7 เป็น Switch Push-Button สำหรับใช้ Power-ON และ Power-OFF ตัวโมดูล
- **หมายเลข** 8 เป็นจั้มเปอร์สำหรับเปิดการทำงานของโมดูล SIM900 แบบอัตโนมัติทันทีเมื่อจ่ายไฟเลี้ยงเข้าบอร์ดโดยให้เลือกไปที่ตำแหน่ง AT แต่ถ้าต้องการควบคุมการเปิดปิดโดย สวิตช์ ON/OFF หรือทางขา PWRKEY ก็ให้เลือกจั้มเปอร์ไปที่ตำแหน่งขา 1-2
- **หมายเลข** 9 เป็น LED แสดงสถานะการทำงานของบอร์ดโดยมีรายละเอียดดังนี้
  - ❖ LED VBAT ใช้ทำหน้าที่แสดงสถานะของแหล่งจ่ายไฟจากภายนอกที่ต่อมาให้กับบอร์ด โดย LED นี้จะติดสว่างก็ต่อเมื่อมีการจ่ายไฟให้กับบอร์ดเป็นที่เรียบร้อยแล้ว
  - ❖ LED NET (NETLIGHT) ใช้แสดงสถานะของโมดูล ในขณะทำการเชื่อมต่อกับเครือข่ายอยู่ โดย LED ตัวนี้จะถูกควบคุมด้วยสัญญาณ NETLIGHT(PIN 52) ของโมดูล SIM900 เมื่อทำงานจะมีสถานะทางโลจิกเป็นโลจิก “1” โดยเมื่อโมดูลอยู่ในสถานะพร้อมทำงาน LED นี้จะติดกระพริบด้วยค่าความเร็วต่างๆ ซึ่งมีความหมายดังนี้
    - ○ OFF แสดงว่าโมดูลอยู่ในสถานะของ Power OFF (ไม่ทำงาน)
    - ○ 64mS ON / 800mS OFF แสดงว่า โมดูล SIM900 ไม่สามารถการค้นหาเครือข่ายได้
    - ○ 64mS ON / 3000mS OFF แสดงว่าโมดูล SIM900 สามารถการค้นหาเครือข่ายได้
    - ○ 64mS ON / 300mS OFF แสดงว่าโมดูล SIM900 อยู่ระหว่างการเชื่อมต่อกับเครือข่ายหรืออุปกรณ์อื่นๆ ด้วย GPRS อยู่
  - ❖ LED STAT (STATUS) ใช้แสดงสถานะของโมดูล SIM900 ว่าทำงานอยู่หรือเปล่า ถ้า LED ติดแสดงว่าโมดูลทำงานอยู่ ถ้า LED ไม่ติดแสดงว่าโมดูลไม่ทำงาน หรืออยู่ในสภาวะ Power down mode
- **หมายเลข** 10 เป็นคอนเน็กเตอร์เสาอากาศ GSM ย่านความถี่ 850/900/1800/1900MHz
- **หมายเลข** 11 เป็น Socket สำหรับติดตั้ง SIM Card ให้กับโมดูล
- **หมายเลข** 12 โมดูล SIM900

## 4. การสั่ง เปิด และ ปิด การทำงานของโมดูล SIM900

ตามปรกติแล้ว โมดูล SIM900 จะมีโหมดการทำงานอยู่หลายโหมด เราสามารถทำงานสั่ง เปิด และปิดการทำงานของโมดูลได้ โดยใช้วิธีดังต่อไปนี้

4.1 **สวิตช์** ON/OFF (SW1) เป็นการสั่ง เปิด และ ปิด การทำงานของโมดูล SIM900 ด้วยการกดสวิตช์ โดยสวิตช์ตัวนี้ จะเป็นแบบ Push-Button Switch (สวิตช์ กดติด-ปล่อยดับ) โดยเป็นการกำหนดสถานะทางโลจิกให้กับขาสัญญาณ PWRKEY(PIN 1) ของโมดูล โดยเมื่อกดสวิตช์จะเป็นโลจิก “0” เมื่อปล่อยสวิตช์จะเป็นโลจิก “1” โดยการทำงานของสวิตช์จะต้องทำการกดสวิตช์ต่อเนื่องกันเป็นเวลามากกว่า 1000mS (1 วินาที) จึงจะมีผลต่อการทำงานของโมดูล โดยลักษณะการทำงานของ

สวิตซ์ จะเป็นแบบ Toggle กล่าวคือ ถ้าโมดูลอยู่ในสถานะของ Power OFF อยู่ แล้วทำการกดสวิตซ์ เป็นเวลามากกว่า 1000mS (1 วินาที) จะเป็นการสั่งให้โมดูลกลับเข้าสู่ Power ON หรือพร้อมทำงาน แต่ถ้าหากว่าโมดูลอยู่ในสถานะของ Power ON อยู่ แล้วทำการกดสวิตซ์ เป็นเวลามากกว่า 1000mS (1 วินาที) แล้วปล่อยจะเป็นการสั่งให้โมดูลหยุดทำงานและกลับเข้าสู่สถานะของ Power OFF (หยุดทำงาน) ส่วนสถานะ LED ต่างๆ แสดงดังตาราง

| LED สถานะ | Power-ON | Power-OFF |
|---|---|---|
| VBAT (เขียว) | ติดสว่าง | ติดสว่าง |
| NET(ส้ม) | กระพริบ | ดับ |
| STAT(เขียว) | ติดสว่าง | ดับ |

**ตาราง แสดงสถานะของ LED ในโหมดต่างๆ**

4.2 **ควบคุมการเปิดปิดทางขา PWRKEY** การสั่งเปิดปิดการทำงานของโมดูลแบบนี้จะใช้สัญญาณควบคุมจากภายนอก เช่น จากไมโครคอนโทรลเลอร์ โดยผ่านทางขา PWRKEY (คอนเน็คเตอร์ P4 หรือ P5) โดยลักษณะการทำงานของสวิตซ์ จะเป็นแบบ Toggle กล่าวคือ ถ้าโมดูลอยู่ในสถานะของ Power OFF อยู่ แล้วทำการส่งลอจิก “1” เป็นเวลามากกว่า 1000mS (1 วินาที) แล้วปล่อยเป็นลอจิก “0” จะเป็นการสั่งให้โมดูลกลับเข้าสู่ Power ON หรือพร้อมทำงาน แต่ถ้าหากว่าโมดูลอยู่ในสถานะของ Power ON อยู่ แล้วทำการการส่งลอจิก “1” เป็นเวลามากกว่า 1000mS (1 วินาที) แล้วปล่อยเป็นลอจิก “0” จะเป็นการสั่งให้โมดูลหยุดทำงานและกลับเข้าสู่สถานะของ Power OFF (หยุดทำงาน)

4.3 **เปิดการทำงานแบบอัตโนมัติ** การทำงานแบบนี้จะเปิดการทำงานโมดูล SIM900 ทันทีเมื่อจ่ายไฟเลี้ยงเข้าบอร์ด ET-BASE GSM SIM900 ซึ่งสามารถทำได้โดยให้เลือกจั้มเปอร์ JP1 ไปที่ตำแหน่ง AT

## 5. การติดต่อสื่อสารกับโมดูล SIM900

การติดต่อสื่อสารกับโมดูล SIM900 ของบอร์ด ET-BASE GSM SIM900 นั้นสามารถทำได้ 2 แบบ คือ เชื่อมต่อผ่านพอร์ตสื่อสารอนุกรม RS232 โดยใช้ขั้วต่อแบบ 4 PIN จัดเรียงสัญญาณตามมาตรฐานของบริษัท ETT ซึ่งสามารถนำไปเชื่อมต่อกับสัญญาณ RS232 มาตรฐาน เช่น คอมพิวเตอร์ RS232(Com Port) หรือ บอร์ดไมโครคอนโทรลเลอร์ของบริษัท ETT ที่มีขั้ว RS232 แบบ 4 PIN ได้ทันที นอกจากนี้บอร์ด ET-BASE GSM SIM900 ยังได้เตรียมขั้วต่อสัญญาณอนุกรมระดับสัญญาณ TTL 3-5V (P4 หรือ

P5) สำหรับเชื่อมต่อกับไมโครคอนโทรลเลอร์โดยตรง โดยไม่ต้องมีวงจรแปลงระดับสัญญาณเป็น RS232 โดยสัญญาณการเชื่อมต่ออนุกรมของโมดูล SIM900 จะมีดังนี้

- DCD (Data Carrier Detect) ของโมดูล SIM900 ซึ่งเป็น Output จาก SIM900 ซึ่งตามปรกติจะต่อเข้ากับ DCD Input ของอุปกรณ์ด้าน Host
- TXD (Transmit Data) ของโมดูล SIM900 ซึ่งเป็น Output จาก SIM900 ซึ่งตามปรกติจะต่อเข้ากับ RXD (Receive Data) ของอุปกรณ์ด้าน Host
- RXD (Receive Data) ของโมดูล SIM900 ซึ่งเป็น Input ของ SIM900 ซึ่งตามปรกติจะต่อเข้ากับ TXD (Transmit Data) จากอุปกรณ์ด้าน Host
- DTR (Data Terminal Ready) ของโมดูล SIM900 ซึ่งเป็น Input ของ SIM900 ซึ่งตามปรกติจะต่อเข้ากับ DTR จากอุปกรณ์ด้าน Host
- RTS (Request To Send) ของโมดูล SIM900 ซึ่งเป็น Input ของ SIM900 ซึ่งตามปรกติจะต่อเข้ากับ RTS ของอุปกรณ์ด้าน Host
- CTS (Clear To Send) ของโมดูล SIM900 ซึ่งเป็น Output จาก SIM900 ซึ่งตามปรกติจะต่อเข้ากับ CTS ของอุปกรณ์ด้าน Host
- RI (Ring Indicator) ของโมดูล SIM900 ซึ่งเป็น Output จาก SIM900 ซึ่งตามปรกติจะต่อเข้ากับ RI ของอุปกรณ์ด้าน Host
- GND ของโมดูล SIM900 ต้องต่อเข้ากับ GND ของอุปกรณ์ด้าน Host

**แสดงการต่อสายสัญญาณระหว่าง ET-BASE GSM SIM900 กับ ไมโครคอนโทรลเลอร์**

| SIM900 | Signal Direction | MCU |
|---|---|---|
| DCD | → | DCD |
| TXD | → | RXD |
| RXD | ← | TXD |
| DTR | ← | DTR |
| RTS | ← | RTS |
| CTS | → | CTS |
| RI | → | RI |
| GND | ___ | GND |

**ตารางการเชื่อมต่อสัญญาณแบบเต็ม**

| SIM900 | Signal Direction | MCU |
|---|---|---|
| TXD | → | RXD |
| RXD | ← | TXD |
| GND | ___ | GND |

**ตารางการเชื่อมต่อสัญญาณแบบ 3 เส้น**

**<u>คำแนะนำ</u>** ในกรณีที่ใช้การเชื่อมต่อสัญญาณแบบ 3 เส้น (RXD,TXD,GND) ต้องกำหนดเงื่อนไขของ Flow Control ให้กับโมดูล SIM900 เป็น No flow control โดยใช้คำสั่ง "AT+IFC=0,0" หรือ XON/XOFF โดยใช้คำสั่ง "AT+IFC=1,1"

## 6. คุณสมบัติการทำงานของสัญญาณที่ควรรู้

- RI(Ring Indicator) เป็น Output จากโมดูล SIM900 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 SIM900 เมื่อต้องการให้โมดูลทำงานต้องให้ขาสัญญาณนี้ได้รับโลจิก LOW ถ้าขา DTR ได้รับโลจิก HIGH โมดูลจะหยุดทำงานและเข้าสู่ Sleep Mode โดยอัตโนมัติ (ถ้ามีการสั่ง Enable Sleep Mode ด้วยคำสั่ง "AT+CSCLK=1" ไว้) ดังนั้นถ้าต้องการให้โมดูลทำงานตลอดเวลาต้องควบคุมให้ขาสัญญาณ DTR ด้านโมดูลได้รับโลจิก LOW หรือสั่งปิดการทำงานของ Sleep Mode โดยใช้คำสั่ง "AT+CSCLK=0" แล้วบันทึกค่า Configuration นี้ไว้ก็ได้เช่นเดียวกัน

## 7. ตัวอย่างการใช้งาน AT Command เพื่อสั่งงานโมดูล SIM900

โมดูล GSM/GPRS รุ่น SIM900 ถูกออกแบบให้ทำหน้าที่เหมือน Modem โดยจะใช้การติดต่อสั่งงานและสื่อสารกับโมดูล ผ่านทางพอร์ตสื่อสาร RS232 รองรับ Baudrate ตั้งแต่ 1200-115200 bps โดยใช้ชุดคำสั่งแบบ AT Command ซึ่งจะมีรูปแบบการใช้งานเหมือนกับ Modem มาตรฐานทั่วๆไป เพียงแต่จะมีการเพิ่มเติม Option และคำสั่งพิเศษอื่นๆเพิ่มเติมขึ้นมาอีก เพื่อให้เหมาะสมและสอดคล้องกับความสามารถในการทำงานของโมดูลได้อย่างครบถ้วน

สำหรับรายละเอียดการใช้คำสั่ง AT Command ที่จะใช้ สำหรับติดต่อสั่งงานโมดูล SIM900 ไม่ว่าจะเป็น รูปแบบคำสั่ง และ หน้าที่การทำงานของแต่ละคำสั่ง ผู้ใช้สามารถศึกษารายละเอียดต่างๆได้จากคู่มือคำสั่ง AT Command (**ไฟล์เอกสารชื่อ** SIM900_AT Command Manual_V1.06.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 จะใช้สำหรับสั่งอ่านค่าพารามิเตอร์ที่กำหนดไว้แล้วของคำสั่งนั้นๆ โดยโมดูลจะตอบรับด้วยการพิมพ์ค่าพารามิเตอร์ปัจจุบันที่กำหนดไว้แล้วให้ทราบ |
| กำหนดค่าการทำงาน | **AT+<x>=<...>** | รูปแบบการใช้คำสั่งแบบนี้ จะใช้สำหรับสั่งเขียนหรือกำหนดค่าพารามิเตอร์ให้กับคำสั่ง เช่น การกำหนดค่า Baudrate |
| สั่งให้ทำงาน | **AT+<x>** | รูปแบบการใช้คำสั่งแบบนี้ จะใช้สำหรับสั่งงานให้โมดูลปฏิบัติตามคำสั่งที่ต้องการ เช่น การสั่งรีเซ็ต (ATZ) |

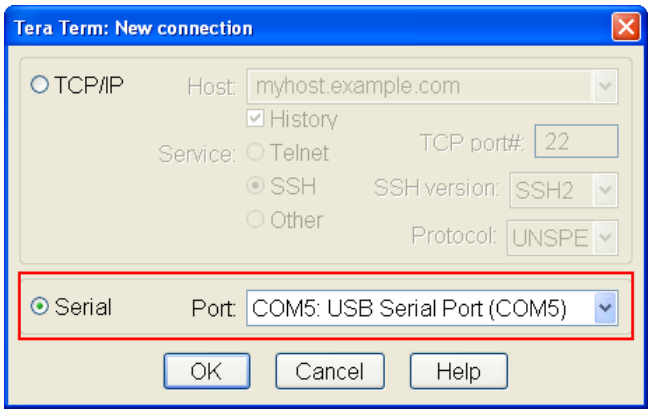
**ตารางแสดง รูปแบบการใช้งาน** AT Command (**เมื่อ** <x> **คือ รหัสคำสั่ง**)

## 8. การทดสอบการทำงานของบอร์ด ET-BASE GSM SIM900

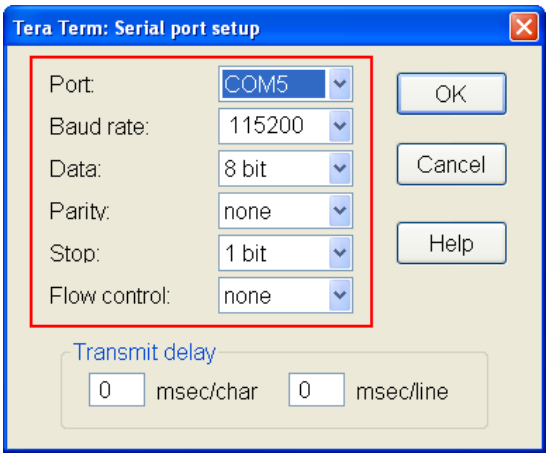
ดังได้ทราบแล้วว่าในการสั่งงานโมดูล SIM900 นั้น จะใช้วิธีการส่งค่ารหัสคำสั่งในรูปแบบของ AT Command ผ่านทางพอร์ตสื่อสารอนุกรมไปให้กับโมดูล ซึ่งตามปรกติจะต้องเขียนโปรแกรมเพื่อส่งรหัสคำสั่งต่างๆไปให้กับโมดูลเอง ขึ้นอยู่กับว่าจะใช้อุปกรณ์ใดเป็นตัวควบคุมการทำงานของโมดูล ซึ่งไม่ได้จำกัดว่าเป็นอุปกรณ์แบบใด อาจจะเป็นคอมพิวเตอร์ PC หรือ ไมโครคอนโทรลเลอร์ตระกูลใดๆก็ได้ ขอให้มีพอร์ตสื่อสารอนุกรม RS232 อยู่ก็สามารถนำมาเชื่อมต่อเพื่อสั่งงานโมดูล SIM900 ได้แล้ว ส่วนที่ว่าจะเขียนโปรแกรมอย่างไร และจะใช้ภาษาใดในการเขียนนั้น ขึ้นอยู่กับผู้พัฒนาโปรแกรมว่า มีความถนัดอย่างไรและมีพื้นฐานอะไรอยู่บ้าง ซึ่งหลักสำคัญก็คือ ผู้พัฒนาต้องหาคำตอบให้ได้ว่า การจะเขียนโปรแกรมสั่งงานอุปกรณ์ทำการ ส่ง และ รับ ข้อมูล ด้วยพอร์ตสื่อสารอนุกรม RS232 นั้นจะต้องทำอย่างไร ซึ่งจะไม่ขอกล่าวถึงในที่นี้ด้วย

สำหรับในการศึกษาเบื้องต้นนั้น ยังไม่จำเป็นต้องใช้วิธีการเขียนโปรแกรมก็ได้ แต่สามารถใช้โปรแกรมสำเร็จรูปจำพวก Serial Terminal ต่างๆ ของคอมพิวเตอร์เป็นตัวทดสอบการทำงานเพื่อทำความเข้าใจกับรูปแบบคำสั่งและผลของการทำงานต่างๆให้เข้าใจเสียก่อน ตัวอย่างเช่น ถ้าต้องการจะสั่งให้โมดูล SIM900 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 ก็จะทำการเลื่อน 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-BASE GSM SIM900 จากนั้นคลิก OK ดังรูป

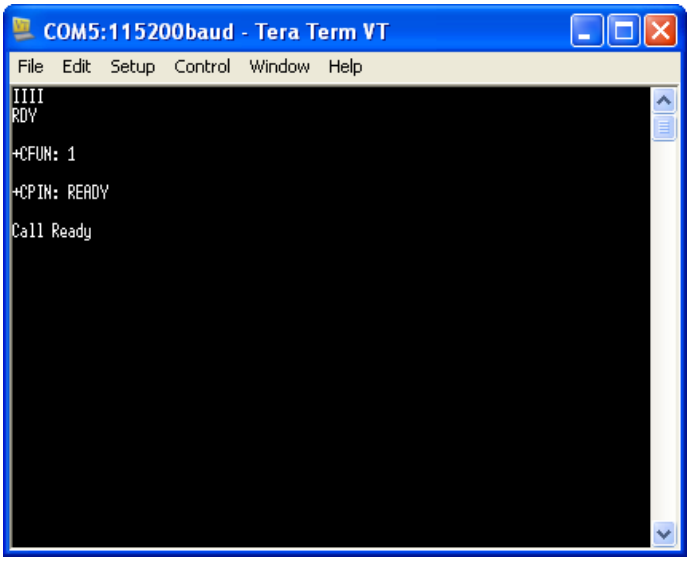



8.2 เลือกที่เมนู Setup-->Serial port… เพื่อตั้งค่าของพอร์ต RS232 ในขั้นตอนนี้ให้เลือกค่า Baud rate ให้ตรงและสอดคล้องกับที่กำหนดให้กับโมดูลไว้ หรือในกรณีที่กำหนดค่า Baud rate ของโมดูลเป็น Auto-Baud rate ไว้ก็สามารถกำหนดค่าใดๆ ที่โมดูลสามารถรองรับได้ระหว่าง 1200, 2400,4800 ,9600,19200, 38400, 57600,115200 ในที่นี้จะเลือก 115200 ส่วน Data ให้เลือกเป็น 8 Bit ,Parity =None, Stop bits=1, Flow Control = None แล้วเลือก “OK” ดังตัวอย่าง

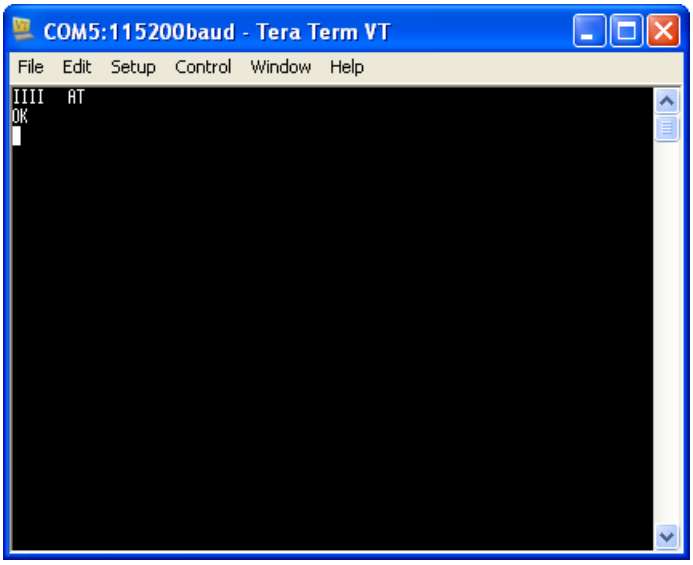



8.3 ซึ่งหลังจากกำหนดการเชื่อมต่อต่างๆเรียบร้อยแล้ว ถ้าทุกอย่างถูกต้องให้ทดลองทำการต่อสายสัญญาณ RS232 ระหว่างบอร์ดกับ Comport ของคอมพิวเตอร์ PC (ต้องเลือกจั้มเปอร์ JP2,JP3 ไปที่ตำแหน่ง RS232 ด้วย) แล้วจ่ายไฟเลี้ยงวงจรให้กับบอร์ด ซึ่งถ้าทุกอย่างถูกต้องจะเห็น LED VBAT สีเขียวบนบอร์ดติดสว่างให้เห็น จากนั้นให้สั่ง Power-ON ตัวโมดูล โดยการกดสวิตช์ ON/OFF ค้างไว้ประมาณ 1 วินาที จะสังเกตุเห็น LED STAT (STATUS) ติดสว่างขึ้น จากนั้น LED NET (NETLIGHT) ก็จะเริ่มกระพริบเป็นจังหวะตลอดเวลา แสดงว่าโมดูลเริ่มต้นทำงานแล้ว ส่วนที่หน้าจอของ Tera Term จะปรากฏ

ข้อความการทำงานให้เห็น ให้รอจนพบคำว่า “Call Ready” ซึ่งหมายถึงโมดูลทำการค้นหาและเครือข่ายได้เรียบร้อยแล้ว จากนั้นก็จะสามารถสั่งงานโมดูลด้วยคำสั่งต่างๆ ได้ตามต้องการดังตัวอย่าง




ในกรณีที่หน้าจอของโปรแกรม HyperTerminal ไม่ปรากฏข้อความใดๆ โมดูล SIM900 อาจถูกตั้งค่า Baud rate เป็นแบบ Auto Baud rate ไว้ ซึ่งโมดูลจะทำการปรับค่าความเร็ว Baud rate ให้โดยอัตโนมัติ ซึ่งค่านี้เป็นค่าเริ่มต้นจากโรงงาน ให้ผู้ใช้พิมพ์ตัวอักษร AT (ตัวพิมพ์ใหญ่เท่านั้น) และตามด้วยกดปุ่ม Enter จากนั้นโมดูลจะตอบ OK กลับมาดังรูป แสดงว่าโมดูลพร้อมทำงานแล้ว




ถ้าผู้ใช้ต้องการให้ปรากฏข้อความ Call Ready ก็ต้องตั้งค่า Baud rate ของบอร์ดให้เป็นค่าคงที่ โดยใช้คำสั่ง AT+IPR=115200 จากนั้นกดปุ่ม Enter ซึ่งคำสั่งนี้จะตั้งค่า Baud rate เป็น 115200 bps จากนั้นให้ ทดลองปิดโมดูลและทำการเปิดใหม่ ก็จะเห็นข้อความ Call Ready เมื่อโมดูลพร้อมทำงาน

## 9. ตัวอย่างการใช้งานคำสั่งเกี่ยวกับ RS232

ในที่นี้จะขอยกตัวอย่างคำสั่งที่ใช้งานบ่อยๆ ถ้าต้องการดูข้อมูลเพิ่มเติมให้ศึกษาจากเอกสาร SIM900_AT Command Manual_V1.06.pdf

### 9.1 ตัวอย่างการใช้งานคำสั่งตั้งค่า Baud rate

สมมุติว่าเราต้องการใช้คำสั่งสำหรับกำหนดค่าอัตราความเร็วของการสื่อสารของโมดูล ซึ่งจะต้องใช้คำสั่ง AT+IPR โดยเราสามารถสั่งงานคำสั่งนี้ได้หลายรูปแบบดังตัวอย่างต่อไปนี้ คือ ถ้าเราจำไม่ได้ว่าค่าพารามิเตอร์ของคำสั่งหรืออัตรา Baud rate ที่สามารถกำหนดได้ มีค่าอะไรบ้าง และกำหนดอย่างไร เราก็สามารถใช้รูปแบบการทดสอบคำสั่ง โดยใช้คำสั่ง AT+IPR=? และจบด้วย Enter เพื่อสอบถามได้ โดยโมดูลจะตอบรับด้วย +IPR: พร้อมกับพิมพ์ค่าพารามิเตอร์ของคำสั่งที่มีอยู่ทั้งหมดให้ทราบ คือ 0,300,1200,…,115200 ดังตัวอย่าง ( **ข้อความสีดำคือคำสั่งที่ป้อนเข้าไป ส่วนสีแดงคือข้อความที่โมดูลตอบกลับมา** )

```
AT+IPR=?<Ent>
+IPR: (),(0,1200,2400,4800,9600,19200,38400,57600,115200)

OK
```

ถ้าต้องการทราบว่าในขณะนี้ ค่า Baud rate ที่กำหนดไว้แล้ว มีค่าเป็นเช่นไร ก็สามารถใช้รูปแบบคำสั่งสำหรับสั่งอ่านค่าพารามิเตอร์ของคำสั่งนี้ คือ AT+IPR? ซึ่งโมดูลจะตอบรับด้วย +IPR: ตามด้วยค่าพารามิเตอร์ที่กำหนดไว้แล้วให้ทราบดังตัวอย่าง (0=Auto Baudrate)

```
AT+IPR?<Ent>
+IPR: 0

OK
```

จากตัวอย่างข้างต้น จะเห็นได้ว่าค่าพารามิเตอร์ของคำสั่ง IPR ที่กำหนดไว้แล้วคือ 0 ซึ่งหมายถึง Auto-Baud rate โดยโมดูลจะทำการปรับค่าความเร็ว Baud rate โดยอัตโนมัติ ในตอนเริ่มต้นการทำงานครั้งแรก แต่ถ้าเราต้องการกำหนดค่า Baud rate เป็นค่าคงที่ไปเลย เพื่อให้โมดูลใช้อัตราความเร็วนี้ตลอด ก็สามารถใช้รูปแบบคำสั่ง กำหนดค่าการทำงาน ได้ เช่น ถ้าต้องการกำหนด

Baud rate เป็น 115200 ก็จะใช้รูปแบบคำสั่งเป็น AT+IPR=115200 และตามด้วยกดปุ่ม Enter ดังตัวอย่าง

```
AT+IPR=115200<Ent>
OK
```

ซึ่งหลังจากสั่งเปลี่ยนค่า Baud rate เป็น 115200 แล้ว ต่อจากนี้ไปก็สามารถสื่อสารกับโมดูลด้วยค่าความเร็วนี้ได้ตลอด

## 9.2 การกำหนด Flow Control

โมดูล SIM900 สามารถกำหนด Flow Control หรือ รูปแบบการตรวจสอบความพร้อมในการสื่อสารและรับส่งข้อมูลได้ด้วย ซึ่ง Flow Control จะมีความสำคัญเป็นอย่างมาก เนื่องจากการประมวลผลของอุปกรณ์ต่างๆจะมีความช้าเร็วที่แตกต่างกัน เมื่อมีการรับส่งข้อมูลที่มีจำนวนข้อมูลมากๆแบบต่อเนื่องนั้น ถ้าฝ่ายรับไม่พร้อมรับข้อมูลแต่ฝ่ายส่งยังคงส่งข้อมูลออกไป ก็จะทำให้ข้อมูลสูญหายและเกิดความผิดพลาดขึ้นได้ โดย SIM900 เองรองรับการตรวจสอบความพร้อมหรือ Flow Control ได้ 2 แบบ คือ

- Software Flow Control (XON/XOFF Flow Control) เป็นการตรวจสอบความพร้อมด้วย Software โดยจะใช้รหัส XOF(13H) เป็นตัวสั่งหยุดการส่งข้อมูลจากฝ่ายส่ง และใช้รหัส XON(11H) เพื่อบอกหรืออนุญาตให้ฝ่ายส่งเริ่มต้นส่งข้อมูลลำดับต่อไปมายังโมดูลได้ โดยการใช้ Flow Control แบบนี้เหมาะกับการเชื่อมต่อกับอุปกรณ์ที่ไม่มีสัญญาณตรวจสอบความพร้อม เช่น ไมโครคอนโทรลเลอร์ หรืออุปกรณ์ที่ใช้การต่อสายสัญญาณเพียง 3 เส้น (RXD,TXD และ GND)
- Hardware Flow Control (RTS/CTS Flow Control) เป็นการตรวจสอบความพร้อมด้วยสัญญาณทางฮาร์ดแวร์ โดยใช้การ Active(“LOW”) สัญญาณ CTS เพื่อบอกให้ฝ่ายส่งหยุดการส่งข้อมูลเมื่อโมดูลไม่พร้อมรับข้อมูล และในทางกลับกันก่อนการส่งข้อมูลกลับออกไปมันจะตรวจสอบสถานะของ RTS ว่า Active อยู่หรือไม่ ถ้า Active(“LOW”) แสดงว่าฝ่ายรับยังไม่พร้อมรับมันจะหยุดรอจนกว่า RTS จะเป็น “HIGH”

การกำหนด Flow Control เป็น No flow control จะใช้คำสั่ง AT+IFC=0,0 **(ค่าเบื้องต้นจากโรงงาน)**
การกำหนด Flow Control เป็น XON/XOFF จะใช้คำสั่ง AT+IFC=1,1
การกำหนด Flow Control เป็น RTS/CTS จะใช้คำสั่ง 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) แต่ในกรณีที่ไม่ต้องการก็สามารถปิดความสามารถนี้ได้ด้วยการใช้คำสั่ง ATE0&W แล้วตามด้วย Enter ซึ่งเมื่อพิมพ์คำสั่งลงไปจะไม่เห็นบนโปรแกรม Terminal จะแสดงเพียงค่าที่โมดูล SIM900 ตอบกลับมาเท่านั้น ถ้าต้องการกลับมาใช้ใหม่ก็ใช้คำสั่ง ATE1&W แล้วตามด้วย Enter

## 10. การ Setup และตรวจสอบค่า Configuration

ตามปรกติแล้วการทำงานของโมดูล SIM900 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 ซึ่งการทำงานของ

โมดูลก็จะปรับเปลี่ยนไปตามการสั่งงานในขณะนั้นๆ แต่เมื่อสั่งรีเซ็ตการทำงานของโมดูล หรือ มีการ Power ON ใหม่คุณสมบัติการทำงานของโมดูลจะถูกเปลี่ยนกลับเป็นค่าเดิมอีก

โดยเราสามารถใช้คำสั่ง AT Command ในการสั่ง ตรวจสอบ และ บันทึกค่า Configuration ต่างๆให้กับโมดูล SIM900 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)
- ATE=1 (Echo Mode ON)
- AT+CSCLK=0 (Disable Sleep Mode)

## 11. การตรวจสอบข้อมูลต่างๆ ของโมดูล SIM900

### 11.1 การตรวจสอบคุณภาพสัญญาณ

การตรวจสอบคุณภาพสัญญาณ จะใช้คำสั่ง AT+CSQ โดยคำสั่งนี้ใช้ตรวจสอบระดับความแรงของสัญญาณ โดยโมดูลจะตอบรับเป็นค่าตัวเลข 0..31 หรือ 00 โดยถ้าค่าระหว่าง 2..30 อยู่ในเกณฑ์ดี ถ้าค่า 31 ถือว่าระดับสัญญาณดีมาก ส่วน 99 หมายถึงตรวจสอบไม่ได้ รูปแบบดังตัวอย่าง

```
AT+CSQ<Ent>
+CSQ: 16,0

OK
```

### 11.2 การตรวจสอบ รหัสผลิตภัณฑ์

```
ATI<Ent>
SIM900 R11.0

OK
```

## 11.3 การตรวจสอบ รหัสผู้ผลิต

```
AT+GMI<Ent>
SIMCOM_Ltd

OK
```

## 11.4 การตรวจสอบ รหัสรุ่น

```
AT+GMM<Ent>
SIMCOM_SIM900
OK
```

## 11.5 การตรวจสอบ รหัส Version Firmware

```
AT+GMR<Ent>
Revision:1137B10SIM900M64_ST_PZ

OK
```

## 11.6 การตรวจสอบ Serial Number (IMEI) ของโมดูล

```
AT+GSN<Ent>
012896001185450

OK
```

## 11.7 การตรวจสอบ รหัสเครือข่าย SIM ผู้ให้บริการ

```
AT+COPS?<Ent>
+COPS: 0,0,"TH GSM"

OK
```

## 12. การโทรออก การรับสาย และ การยกเลิกการโทร

- ใช้คำสั่ง ATD เพื่อสั่งโทรออก โดยรูปแบบการใช้คำสั่งให้ตามด้วยเบอร์ปลายทาง
- ใช้คำสั่ง ATDL เพื่อสั่งโทรออกด้วยหมายเลขโทรออกครั้งสุดท้าย
- ใช้คำสั่ง ATA เพื่อรับสายเรียกเข้า โดยเมื่อมีสายเรียกเข้าจะมีเสียงเรียกเข้าที่ หูฟังของ Handset ให้เราทราบ ถ้าต้องการรับสายให้ใช้คำสั่ง “ATA” เพื่อรับสายได้ทันที ซึ่งหลังจากสั่งรับสายแล้วผู้ใช้จะสามารถพูดคุยกับปลายสายได้ทันที โดยใช้ Handset หรือชุด ปากพูดหูฟังของโทรศัพท์บ้าน
- ใช้คำสั่ง ATH เพื่อสั่งวางสาย หรือยกเลิกการโทรออก

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

```
ATD0894469xxx;<Ent>
OK
```

ในกรณีที่สั่งโทรออกแล้วไม่มีการรับสาย หรือ สายไม่ว่างโมดูลจะรายงานผลด้วยข้อความ “BUSY” ดังตัวอย่าง

```
ATD0894469xxx;<Ent>
OK
BUSY
```

ตัวอย่างการสั่งตรวจสอบยอดเงินของระบบ 1-2-CALL ซึ่งถ้าเป็นโทรศัพท์มือถือปรกติจะใช้วิธีการพิมพ์เครื่องหมาย “*121#” แล้วสั่งโทรออก แต่สำหรับโมดูล SIM900 จะต้องใช้คำสั่ง ATD สำหรับสั่งโทรออกแล้วตามด้วยเครื่องหมายแทน ดังตัวอย่าง

```
ATD*121#<Ent>
OK

+CUSD: 0,"The balance of 0870681xxx is 111.62 B. & valid until 03/05/13
Pay59B.Get3G/EDGE 70MB within7days exceed up to main pro.Press*500*70#",64
```

ตัวอย่างการรับสายเรียกเข้า เมื่อมีสายเรียกเข้าโมดูล SIM900 จะมีข้อความ “RING” และสร้างเสียงเรียกเข้าเป็นจังหวะที่ หูฟังของ Handset ให้ทราบ ถ้าผู้ใช้ต้องการรับสาย ให้ใช้คำสั่ง ATA เพื่อสั่งรับสาย หรือใช้คำสั่ง ATH เพื่อวางหูหรือยกเลิกไม่รับสาย ดังตัวอย่าง

```
RING
ATA<Ent>
OK
```

ในกรณีที่ต้องการให้แสดงหมายเลขที่โทรเข้ามา ให้ใช้คำสั่ง AT+CLIP=1 ตามด้วย Enter โมดูลจะแสดงหมายเลขที่โทรเข้าดังตัวอย่าง

```
RING
+CLIP: “0894469xxx”,129,””,,””,0
```

## 13. การตรวจสอบยอดเงินคงเหลือโดยใช้ USSD

สามารถใช้คำสั่ง AT+CUSD ตามด้วยค่า USSD (Unstructure Supplementary Service Data) สำหรับตรวจสอบยอดเงินได้เช่นกันดังตัวอย่าง ซึ่งจากตัวอย่างเป็นของระบบ 1-2-CALL

```
AT+CUSD=1,"*121#"<Ent>
OK

+CUSD: 0,"The balance of 0870681xxx is 111.62 B. & valid until 03/05/13
Pay59B.Get3G/EDGE 70MB within7days exceed up to main pro.Press*500*70#",64
```

## 14. การรับข้อความ SMS

ตามปรกติแล้วโมดูล SIM900 จะสามารถกำหนดโหมดการทำงานของข้อความหรือ SMS ได้ 2 โหมด คือ PDU Mode และ Text Mode โดย PDU Mode การรับและแสดงผลการทำงานของคำสั่งจะเป็นรูปแบบของรหัสตัวเลขแบบ Binary Code ส่วน Text Mode การรับและแสดงผลการทำงานของคำสั่งจะเป็น ข้อความ 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 เช่นถ้าต้องการอ่านข้อความลำดับที่3 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+CMGDA=”DEL ALL” เพื่อสั่งลบข้อความทั้งหมดออกจากหน่วยความจำ

ตัวอย่างการรับข้อความ SMS ในตัวอย่างจะทดสอบด้วยการส่งข้อความ “Hello 12345” ไปให้กับโมดูล SIM900B ซึ่งเมื่อรับข้อความได้จะมีข้อความ +CMTI: “SM”,n โดย n หมายถึงลำดับที่ของข้อความ

```
+CMTI: "SM",3
AT+CMGR=3<Ent>
+CMGR: "REC UNREAD","+66894469xxx",,"07/11/19,13:29:25+28"
Hello 12345

OK
```

ถ้ามีการสั่งอ่านข้อความเดิมซ้ำใหม่สถานะของข้อความจะเปลี่ยนเป็น “REC READ” แทน เพื่อแสดงให้ทราบว่าข้อความนี้ถูกอ่านไปแล้วดังตัวอย่าง

```
AT+CMGR=3<Ent>
+CMGR: "REC READ","+66894469xxx",,"07/11/19,13:29:25+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<Ent>
OK
AT+CSMP=17,167,0,0<Ent>
OK
AT+CSCS="GSM"<Ent>
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 SIM900 ได้รับคำสั่ง AT+CMGS เรียบร้อยแล้วมันจะตอบรับด้วยการส่งเครื่องหมาย ">" กลับมาบอก ซึ่งหลังจากนี้เป็นต้นไปผู้ใช้ก็สามารถจะทำการพิมพ์ข้อความต่างๆที่ต้องการจะส่งให้กับโมดูลได้ทันที โดยให้ปิดท้ายข้อความด้วยการกดปุ่ม Ctrl+Z (0x1A) เช่นถ้าต้องการส่งข้อความ SMS ให้กับหมายเลข 0894469xxx ด้วยข้อความ "Hello Test SMS" จะเป็นดังนี้

```
AT+CMGS="+66894469xxx"<Ent>
> Hello Test SMS<Ctrl+Z>
+CMGS: 6

OK
```

ในกรณีที่พิมพ์คำสั่ง AT+CMGS="+66894469xxx" แล้วข้อความตอบกลับมาว่า ERROR แสดงว่าพิมพ์คำสั่งผิดหรือไม่ได้ตั้งค่ากำหนดรูปแบบของข้อความเป็น Text Mode ให้ตรวจสอบโดยใช้

คำสั่ง 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 | l | 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 SIM900 และสมมุติว่า SIM900 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 แต่จะถือเป็นคำสั่ง

ให้เลื่อน 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 SIM900B 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

แต่ถ้าเราทดลองส่งข้อความ SMS ที่เป็นภาษาอังกฤษเพียงอย่างเดียว เช่น “Hello Jack” ไปให้กับโมดูล SIM900 และสมมุติว่า SIM900 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. การส่ง 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

+ = 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<Ent>
```

```
OK
AT+CSCS="UCS2"<Ent>
OK
AT+CSCA="002B003600360038003100380033003100300038003000380",145<Ent>
OK
AT+CSMP=17,167,0,25<Ent>
OK
AT+CMGS="002B003600360038003100310032003300340035003600 37"<Ent>
>0E010E020E04<Ctrl+Z>                                    ; ข้อความภาษาไทยที่จะส่ง
+GMGS: 57

OK
```

## 19. การใช้ SIM Command (SIM Application Toolkit : STK)

SIM Application Tool Kit เป็นชุดคำสั่งพิเศษของโมดูล SIM900 ใช้สำหรับสั่งงาน SIM ผ่านทางชุดคำสั่งพิเศษใน SIM บางรุ่นที่รองรับสร้าง Application จากผู้ใช้ได้ ซึ่งตามปรกติแล้วผู้ที่จะทำการพัฒนา Application ต่างๆให้กับ SIM ได้นั้น จะต้องมีความรู้ความเข้าใจเกี่ยวกับ SIM เป็นอย่างดี โดยข้อกำหนดต่างๆสามารถศึกษาเพิ่มเติมได้จาก Technical Reference ของ “GSM11.14” ซึ่งสำหรับผู้ให้บริการ SIM ในประเทศไทยเอง ในปัจจุบันก็ได้มีการสร้าง Application บรรจุไว้ภายใน SIM บางรุ่นด้วย เช่น SIM รุ่นที่ใช้สำหรับบริการเติมเงินให้กับโทรศัพท์มือถือ (Mobile Top Up Service) ของค่ายต่างๆ ไม่ว่าจะเป็น AIS หรือ DTAC หรือ TRUE ต่างก็มีการพัฒนา Application บน SIM Card ไว้สนับสนุนและบริการแก่ผู้ใช้งานทั่วไปแล้วทั้งสิ้น

โดยในที่นี้จะไม่ขอกล่าวถึงวิธีการพัฒนา Application บน SIM แต่จะขออธิบายถึงการติดต่อและเข้าถึงคำสั่งใน Application ที่สร้างไว้แล้วใน SIM ตัวอย่างเช่น SIM ของระบบ DTAC ในส่วนที่มีการพัฒนา Application บน SIM ไว้เป็นที่เรียบร้อยแล้ว อันได้แก่ SIM สำหรับผู้ให้บริการเติมเงินมือถือผ่านมือถือ หรือ SIM Happy Online ซึ่งจะเห็นได้ว่าในส่วนของผู้ใช้งานนั้นไม่จำเป็นต้องทราบรายละเอียดของการทำงานใน Application บน SIM แต่อย่างใด ผู้ใช้งานเพียงแต่ศึกษาข้อกำหนดและวิธีการในการเลือกเมนูและป้อนข้อมูลต่างๆให้ถูกต้องเท่านั้น ตัวอย่างเช่น วิธีการสั่งเติมเงินให้กับมือถือจะมีขั้นตอนดังนี้

1. เข้าสู่เมนู Happy Refill
2. เลือกภาษาเพื่อทำการเติมเงิน
3. เลือกเมนูเติมเงิน

4. ระบุเบอร์มือถือที่ต้องการเติมเงิน
5. เลือกราคาหรือจำนวนเงินในการเติมเงิน
6. ใส่รหัสผ่าน
7. ยืนยันการเติมเงิน

ซึ่งหลังจากที่ผู้ใช้ทำรายการต่างๆครบทั้ง 7 ขั้นตอนแล้ว โปรแกรม Application ที่บรรจุไว้ภายใน SIM ก็จะสั่งงานให้โทรศัพท์ส่ง SMS ไปยัง Server ผู้ให้บริการเพื่อดำเนินการเติมเงินให้กับเบอร์ที่ระบุไว้ ซึ่งจะเห็นได้ว่าในส่วนของผู้ใช้งานเอง จะไม่มีโอกาสทราบได้เลยว่า ข้อความ SMS ที่ส่งออกไปเพื่อร้องขอการเติมเงินนั้นเป็นอย่างไร และส่งออกไปยังหมายเลขใด ข้อมูลต่างๆที่ผู้ใช้ทำการป้อนเข้าไปนั้น มีการนำไปแก้ไขดัดแปลง หรือ ผนวกรวมกับข้อมูลอื่นๆ หรือ มีการเข้ารหัสข้อมูลอย่างไรบ้าง ผู้ใช้เองจะไม่มีโอกาสได้ทราบเลย ทำให้ระบบการทำงาน ถูกปกปิดเป็นความลับ และ มีความปลอดภัย มากขึ้น เพราะไม่ต้องกังวลว่าจะมีใครพยายามส่งข้อความ SMS ปลอมเข้ามายัง Server ของผู้ให้บริการเติมเงิน เพื่อขอเติมเงินบ้าง

## ตัวอย่างการตรวจสอบยอดเงินจากกระเป๋าเงินสด True Money ของ SIM True

เราจะมาทดลองทำความเข้าใจกระบวนการทำงานของ Application ใน SIM กันสัก 1 ตัวอย่าง โดยในที่นี้จะขอยกตัวอย่างการใช้ Application ใน SIM ระบบเติมเงิน ของ True ซึ่งได้ทำการ Download Application ของ True Money ให้กับ SIM เรียบร้อยแล้ว โดยในอันดับแรกเราจะทดลองใช้โทรศัพท์มือถือในการทำรายการกันดูก่อน จากนั้นจึงจะทดลองให้โมดูล SIM900 สั่งทำรายการแบบเดียวกันเพื่อให้ผู้ใช้ได้เห็นการทำงานจะได้ทำความเข้าใจได้ดีขึ้น โดยในกรณีใช้โทรศัพท์มือถือทำรายการจะมีขั้นตอนดังนี้

1. เมื่อเลือกเมนูไปยัง True Money จะมีตัวเลือกรายการหัวข้อย่อยใน Menu ให้เลือก 6 ตัวเลือก คือ
   - Check balance
   - Transfer
   - Refill
   - Change Password
   - Register
   - About
2. สมมุติว่าเราต้องการตรวจสอบยอดเงิน ก็ต้องเลือกทำรายการใน Submenu ลำดับที่ 1 ซึ่งก็คือ “Check Balance”
3. เมื่อโปรแกรม Application ในโทรศัพท์มือถือรับรู้การกดคีย์ มันจะสั่งให้ Application ใน SIM ปฏิบัติงานตามหัวข้อที่ได้เลือกไว้ โดยในกรณีนี้จะปรากฏข้อความ password ที่หน้าจอแสดงผล

ของโทรศัพท์และรอรับการกดคีย์เพื่อป้อนรหัสผ่าน ในขั้นตอนนี้ผู้ใช้ต้องทำการป้อนรหัสผ่านเป็นตัวเลข 4 หลักตามที่ลงทะเบียนกับผู้ให้บริการไว้

4. เมื่อทำการป้อนรหัสผ่านเรียบร้อยแล้วโปรแกรม Application บนมือถือ ก็จะแปลรหัส Password ที่ได้รับจากผู้ใช้ส่งให้กับโปรแกรม Application ใน SIM ซึ่ง Application ใน SIM ก็จะทำการนำค่ารหัส Password ที่รับได้เข้ารหัสแปลงเป็นข้อความ SMS แล้วส่งไปยัง Server ของผู้ให้บริการ
5. ผู้ใช้จะได้รับข้อความเป็น SMS แจ้งยอดเงินคงเหลือให้ทราบดังตัวอย่าง

จากกระบวนการข้างต้นจะเห็นได้ว่าในกรณีที่ใช้ โทรศัพท์มือถือ เป็นตัวทำรายการนั้นจะทำได้ง่ายมาก เนื่องจากโทรศัพท์มือถือเองมี Application รองรับการทำงานเหล่านี้ไว้อยู่แล้ว และมีคีย์บอร์ดพร้อมหน้าจอสำหรับแสดงรายการต่างๆให้เห็นได้โดยสะดวก แต่สำหรับในกรณีที่จะใช้โมดูล SIM900 เป็นตัวทำรายการนั้น จะไม่สามารถทำได้โดยตรง เพราะเรายังไม่มี Application สำเร็จรูปรองรับการทำงานกับ SIM เหมือนกับโทรศัพท์มือถือ แต่สิ่งที่เราสามารถทำได้ก็คือ การใช้ชุดคำสั่งต่างๆของโมดูล SIM900 สำหรับติดต่อกับ SIM อันได้แก่ การเข้าถึง Application ใน SIM การขอดูรายการตัวเลือกในเมนูของ Application ใน SIM การสั่งให้โมดูลปฏิบัติงานตามหัวข้อ(เลือก Submenu) ต่างๆเหล่านี้เป็นต้น

## ตัวอย่าง ขั้นตอนการตรวจสอบยอดเงินของ "True Money" ด้วย SIM900 มีขั้นตอนดังนี้

ก่อนการใช้งานให้ทำการตั้งค่าโมดูล SIM900 ก่อนดังตัวอย่าง ( ศึกษาเพิ่มเติมได้จากเอกสาร AN_SIM900_STK_UGD_V1.00.pdf )

```
AT*PSSTKI=1<Ent>                    ; เปิดการใช้งาน STK function
OK
AT+CMGF=1<Ent>                      ; แสดงในรูปแบบ TEXT mode
OK
AT+CMEE=2<Ent>                      ; แสดงการรายงานของคำสั่ง
OK
AT+CSCS="UCS2"<Ent>                 ; รูปแบบตัวอักษรแบบ UCS2
OK
```

- จากนั้นให้ทำการปิดโมดูล SIM900 แล้วเปิดใหม่จะเห็นข้อความดังตัวอย่าง ซึ่งแสดงให้เห็นว่ามีเมนูของ STK อยู่ทั้งหมด 7 เมนู

```
*PSSTK: "SETUP MENU",1,4,"Menu",0,0,1,0,0,7
```

- เริ่มต้นการใช้งานเมนูโดยใช้คำสั่ง

```
AT*PSSTK="SETUP MENU",1,1<Ent>
OK

*PSSTK: "END SESSION"
```

- แสดงรายการว่ามีเมนูอะไรบ้างโดยใช้คำสั่ง

```
AT*PSSTK="GET ITEM LIST",7<Ent>
```

- หลังจากนั้นโมดูลจะส่งรายการเมนูทั้ง 7 เมนูมาแสดงดังตัวอย่าง

```
*PSSTK: "GET ITEM LIST",1,1,4,"True Money",0,0,0

*PSSTK: "GET ITEM LIST",2,2,4,"True Product",0,0,0

*PSSTK: "GET ITEM LIST",3,3,4,"True Payment",0,0,0

*PSSTK: "GET ITEM LIST",4,4,4,"Other Service",0,0,0

*PSSTK: "GET ITEM LIST",5,5,4,"True Transfer",0,0,0

*PSSTK: "GET ITEM LIST",6,6,4,"Fun & Smart",0,0,0

*PSSTK: "GET ITEM LIST",7,7,4,"Setting2Play",0,0,0

OK
```

- ทำการเลือกเมนู True Money โดยใช้คำสั่งดังตัวอย่าง โดยเลือกเมนูที่ 1 ซึ่งจะเห็นว่าจากข้อความตอบกลับ ในเมนู True Money จะมีเนนูย่อยอีก 6 เมนู

```
AT*PSSTK="MENU SELECTION",1<Ent>
OK

*PSSTK: "SELECT ITEM",0,0,"",0,0,1,0,0,6
```

- แสดงรายการว่ามีเมนูย่อยอะไรบ้างในเมนู True Money โดยใช้คำสั่ง

```
AT*PSSTK="GET ITEM LIST",6<Ent>
*PSSTK: "GET ITEM LIST",1,1,4,"Check Balance",0,0,0

*PSSTK: "GET ITEM LIST",2,2,4,"Transfer",0,0,0

*PSSTK: "GET ITEM LIST",3,3,4,"Refill",0,0,0

*PSSTK: "GET ITEM LIST",4,4,4,"Change Password",0,0,0

*PSSTK: "GET ITEM LIST",5,5,4,"Register",0,0,0

*PSSTK: "GET ITEM LIST",6,6,4,"About",0,0,0

OK
```

- เลือกที่เมนู Check Balance โดยใช้คำสั่งดังตัวอย่าง หลังจากนั้นจะเห็นข้อความเตือนให้ใส่ password ที่ได้สมัครไว้กับบริการทรูมันนี่

```
AT*PSSTK="SELECT ITEM",1,1,0,0<Ent>
OK

*PSSTK: "GET INPUT",1,0,4,1,4,"password:",0,0,255,"",1,4,0
```

- ทำการป้อน password ( 1234 ) โดยใช้คำสั่งดังตัวอย่าง

```
AT*PSSTK="GET INPUT",1,4,"1234",0,0
OK

*PSSTK: "NOTIFICATION",1,19,0,255,"",0,0
```

- ทำการป้อนคำสั่งดังตัวอย่าง รอสักครู่จะเห็นว่ามีข้อความใหม่ เข้ามาในกล่องข้อความ ซึ่งสามารถใช้คำสั่ง AT+CMGR เพื่ออ่านข้อความได้

```
AT*PSSTK="NOTIFICATION",1,0
OK

*PSSTK: "END SESSION"

+CMTI: "SM",7                                   ; มีข้อความใหม่เข้ามา
```

## 20. การอ่านข้อมูลจากเว็บไซต์โดยใช้การเชื่อมต่อ GPRS (HTTP GET)

ในการเชื่อมต่อ GPRS สิ่งที่ผู้ใช้ต้องรู้คือค่า APN ในการเชื่อมต่อเครือข่ายอินเตอร์เน็ต ของแต่ละเครือข่าย ซึ่งแต่ละเครือข่ายจะมีค่าดังนี้

AIS = internet

TRUE = internet

DTAC = www.dtac.co.th

ซึ่งในตัวอย่างนี้จะทดลองดึงข้อมูลจากหน้าเว็บไซต์ www.etteam.com โดยส่งคำสั่งดังต่อไปนี้ ( สามารถศึกษาเพิ่มเติมได้จากเอกสาร SIM900_IP_Application Note_V1.03.pdf )

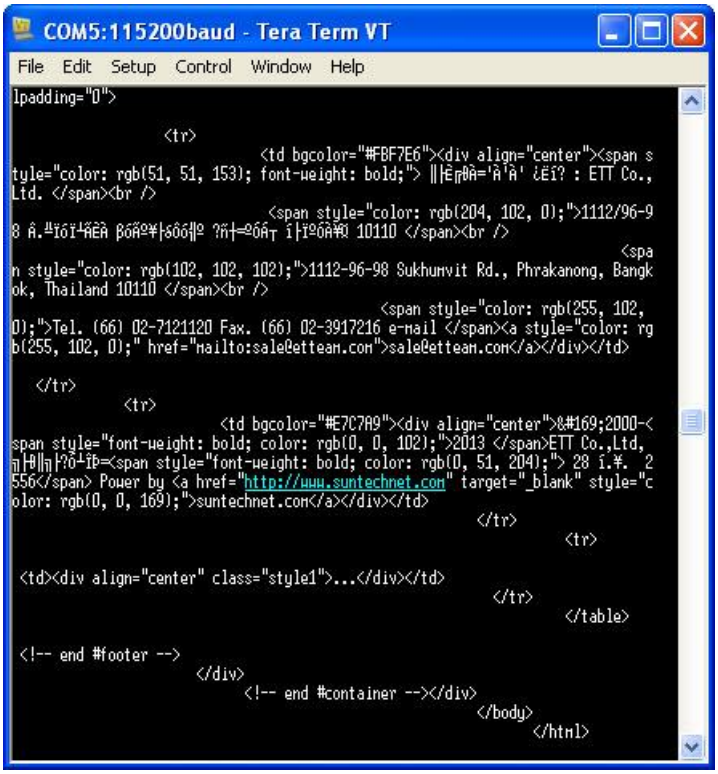
```
AT+SAPBR=3,1,"Contype","GPRS"<Ent>                ; เริ่มเปิดการใช้งาน GPRS
OK
AT+SAPBR=3,1,"APN","internet"<Ent>
OK
AT+SAPBR=1,1<Ent>
OK
AT+SAPBR=2,1<Ent>
```

```
+SAPBR: 1,1,"10.179.72.166"

OK
AT+HTTPINIT<Ent>                                        ; เริ่มการใช้งาน HTTP
OK
AT+HTTPPARA="CID",1<Ent>
OK
AT+HTTPPARA="URL","www.etteam.com"<Ent>                 ; เว็บไซต์ที่ต้องการดึงข้อมูล
OK
AT+HTTPACTION=0<Ent>
OK

+HTTPACTION:0,200,58509
AT+HTTPREAD<Ent>                                        ; เริ่มการอ่านข้อมูล HTTP
+HTTPREAD:58509
```

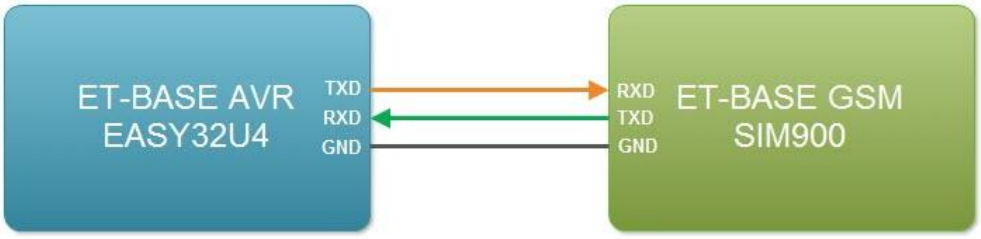
หลังจากนั้นจะปรากฏข้อมูลของหน้าเว็บไซต์ www.etteam.com ดังตัวอย่าง ถ้าผู้ใช้ต้องการสิ้นสุดการเชื่อมต่อ HTTP ให้ใช้คำสั่ง AT+HTTPTERM จากนั้นให้ใช้คำสั่ง AT+SAPBR=0,1 เพื่อปิดการเชื่อมต่อ GPRS

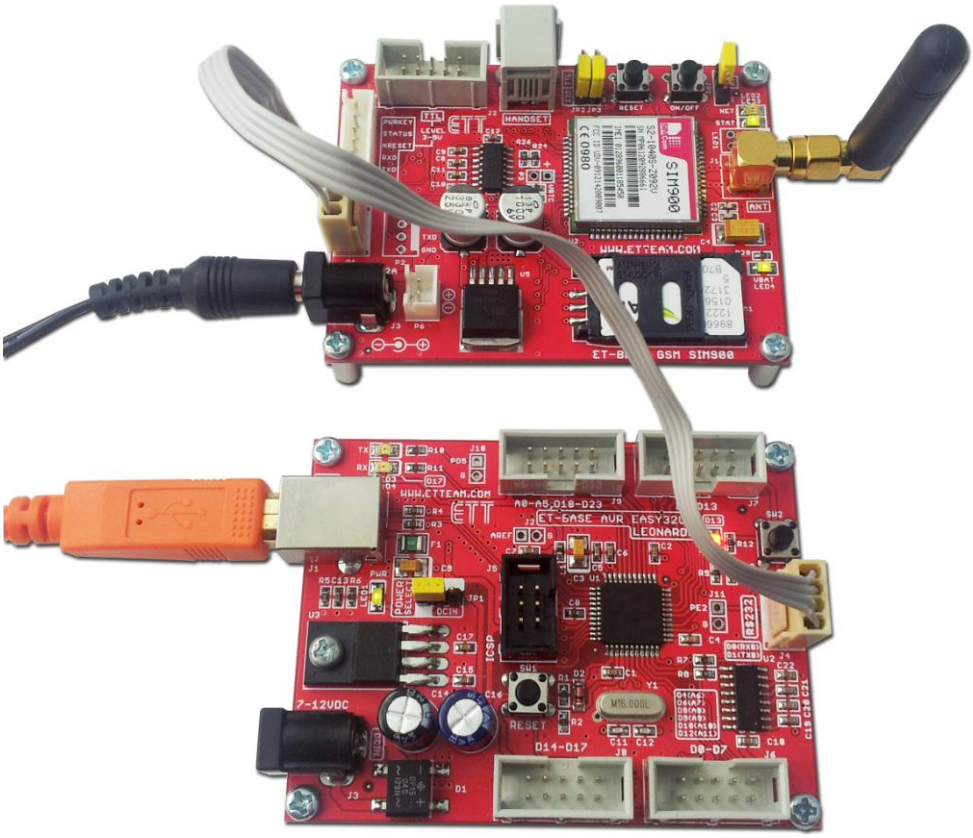


**ข้อควรระวัง** การเชื่อมต่อ GPRS มีค่าใช้จ่าย ดังนั้นผู้ใช้ควรศึกษาถึงอัตราค่าบริการต่างๆ ของแต่ละเครือข่าย และปิดการเชื่อมต่อเมื่อไม่ได้ใช้งานข้อมูล

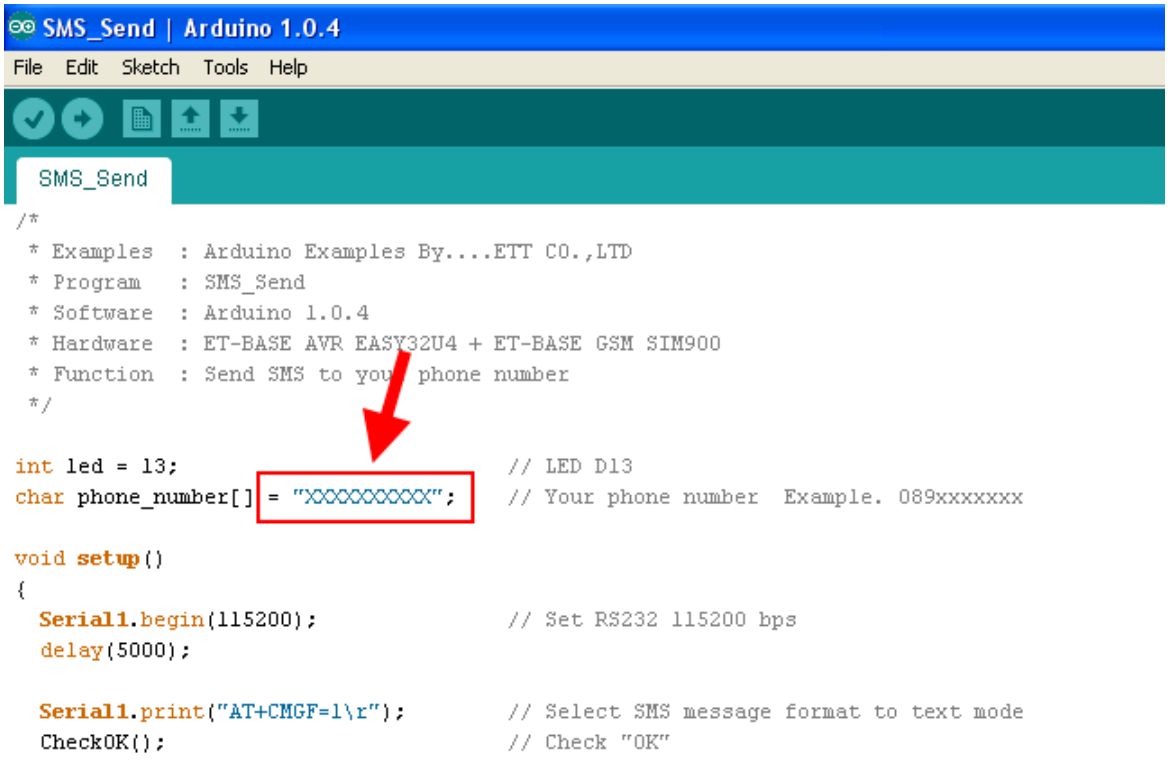
## 21. การเชื่อมต่อบอร์ด ET-BASE GSM SIM900 กับ บอร์ดไมโครคอนโทรลเลอร์

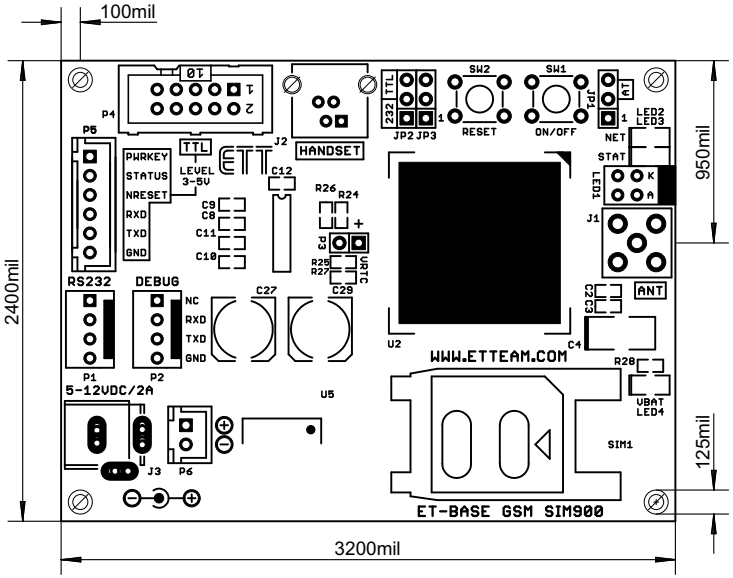
ตัวอย่างนี้จะขอยกตัวอย่างการเชื่อมต่อ ET-BASE GSM SIM900 ร่วมกับบอร์ด ET-BASE AVR EASY32U4 ซึ่งจะตัวอย่างนี้จะเชื่อมต่อผ่านพอร์ต RS232 โดยไดอะแกรมการเชื่อมต่อ และรูปการเชื่อมต่อดังรูป

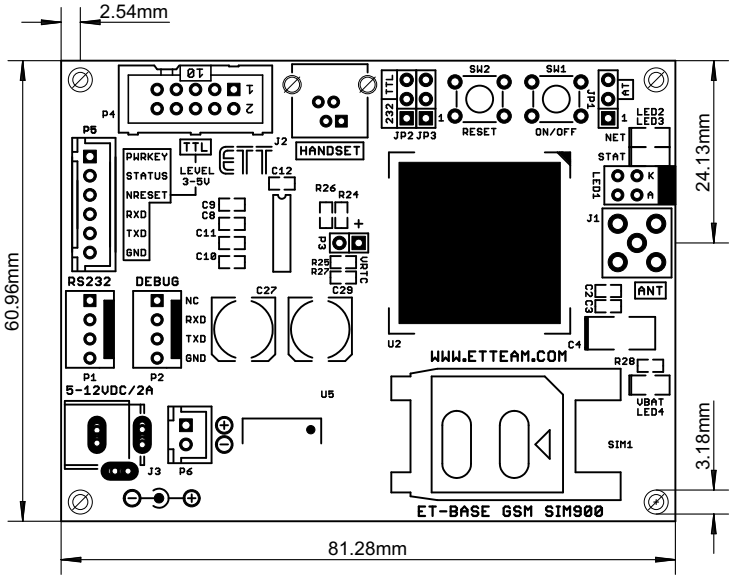


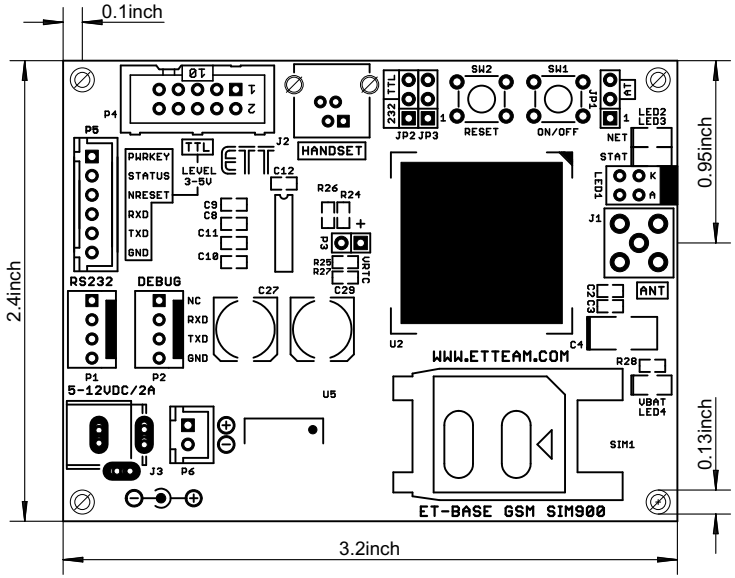



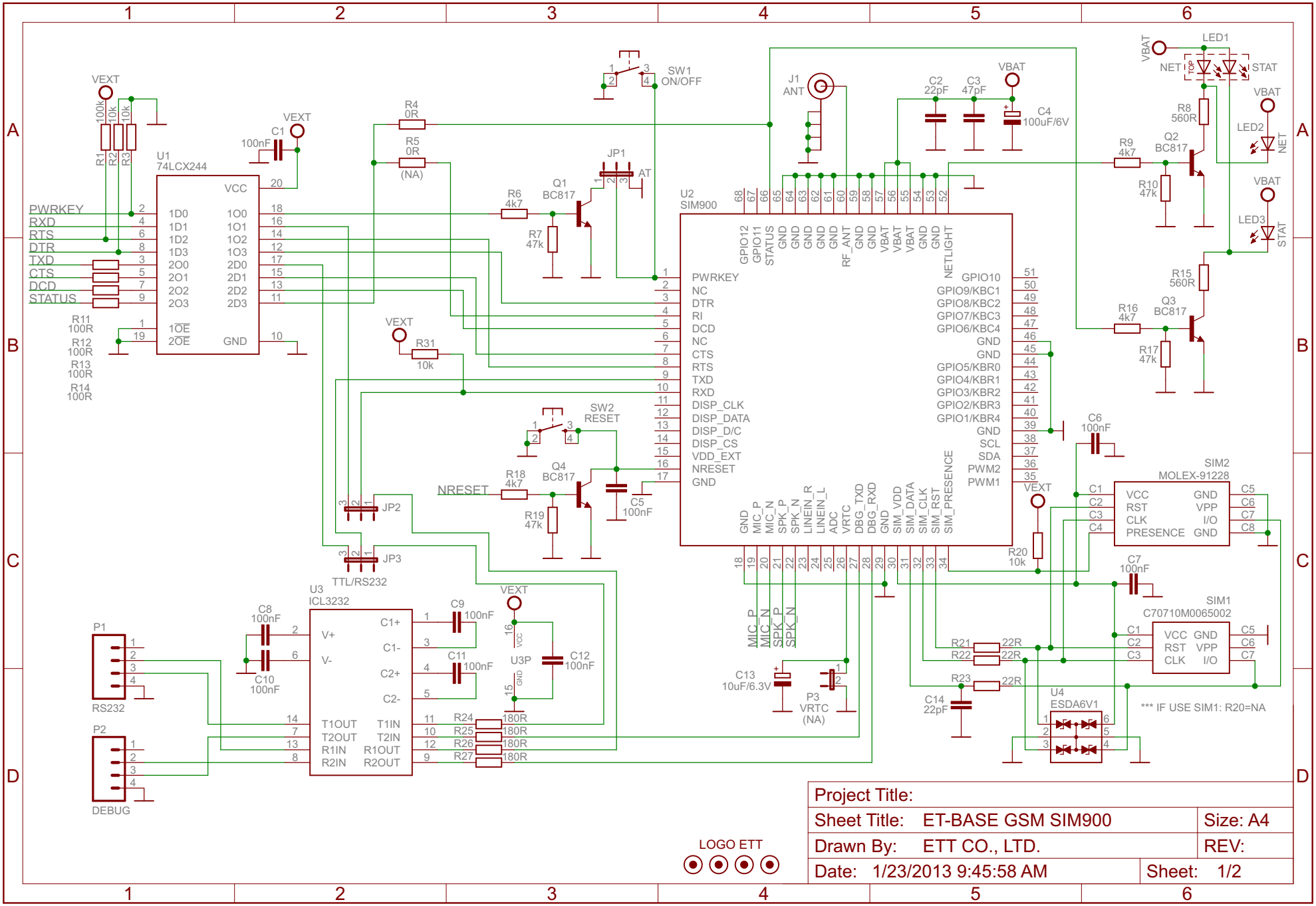
จากนั้นให้เปิดตัวอย่างโปรแกรม SMS_Send ดังรูป ซึ่งตัวอย่างนี้จะเป็นตัวอย่างส่งข้อความไปยังเบอร์ของผู้รับ ให้ทำการแก้ไขข้อความ XXXXXXXXXX เป็นเบอร์ที่ต้องการให้ ET-BASE GSM SIM900 ส่งข้อความไป จากนั้นให้ Verify และ Upload โปรแกรมที่แก้ไขแล้ว จากนั้นรอสักครู่จะมีข้อความ "Hello from ET-BASE GSM SIM900" ไปปรากฏที่เบอร์ของผู้รับ

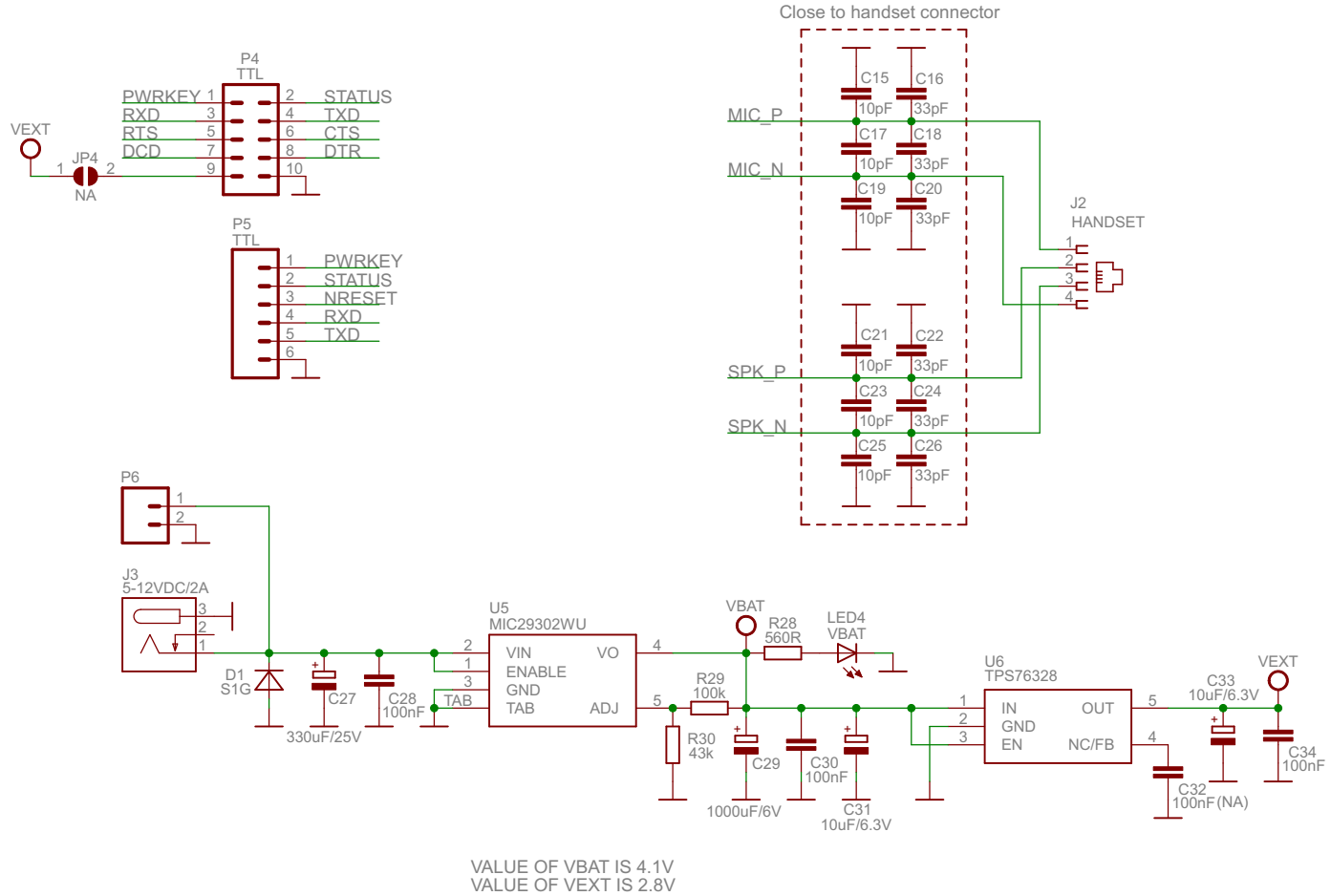



100mil
2400mil
950mil
125mil
3200mil
P4
P5
PWRKEY
STATUS
NRESET
RXD
TXD
GND
TTL
LEVEL
3-5V
J2
HANDSET
C12
R26
R24
C9
C8
C11
C10
P3
R25
R27
VBTC
RS232
DEBUG
NC
RXD
TXD
GND
C27
C29
P1
P2
5-12VDC/2A
J3
P6
U5
U2
WWW.ETTEAM.COM
SW2
SW1
RESET
ON/OFF
JP2
JP3
JP1
ANT
NET
STAT
LED2
LED3
LED1
K
A
J1
ANT
C2
C3
C4
R28
VBAT
LED4
SIM1
ET-BASE GSM SIM900


2.54mm
60.96mm
24.13mm
3.18mm
81.28mm
P4
P5
PWRKEY
STATUS
NRESET
RXD
TXD
GND
TTL
LEVEL
3-5V
J2
HANDSET
C12
R26
R24
C9
C8
C11
C10
P3
R25
R27
VBTC
RS232
DEBUG
NC
RXD
TXD
GND
C27
C29
P1
P2
5-12VDC/2A
J3
P6
U5
U2
WWW.ETTEAM.COM
ET-BASE GSM SIM900
SW2
SW1
RESET
ON/OFF
JP2
JP3
JP1
ANT
NET
STAT
LED2
LED3
LED1
K
A
J1
ANT
C2
C3
C4
R28
VBAT
LED4
SIM1


0.1inch
2.4inch
0.95inch
0.13inch
3.2inch
P4
P5
PWRKEY
STATUS
NRESET
RXD
TXD
GND
TTL
LEVEL
3-5V
J2
HANDSET
C12
C9
C8
C11
C10
R26
R24
P3
R25
R27
VRTC
C29
C27
RS232
DEBUG
NC
RXD
TXD
GND
P1
P2
5-12VDC/2A
J3
P6
U5
U2
WWW.ETTEAM.COM
SW2
SW1
RESET
ON/OFF
JP2
JP3
JP1
LED2
LED3
NET
STAT
LED1
J1
ANT
C2
C3
C4
R28
VBAT
LED4
SIM1
ET-BASE GSM SIM900


U1
74LCX244
U2
SIM900
U3
ICL3232
U4
ESDA6V1
SIM1
C70710M0065002
SIM2
MOLEX-91228
J1
ANT
SW1
ON/OFF
SW2
RESET
JP1
JP2
JP3
TTL/RS232
P1
RS232
P2
DEBUG
P3
VRTC
(NA)
LED1
LED2
LED3
Q1
BC817
Q2
BC817
Q3
BC817
Q4
BC817
VEXT
VBAT
PWRKEY
RXD
RTS
DTR
TXD
CTS
DCD
STATUS
NRESET
*** IF USE SIM1: R20=NA
LOGO ETT
Project Title:
Sheet Title: ET-BASE GSM SIM900
Size: A4
Drawn By: ETT CO., LTD.
REV:
Date: 1/23/2013 9:45:58 AM
Sheet: 1/2




VALUE OF VBAT IS 4.1V
VALUE OF VEXT IS 2.8V

| | | |
|---|---|---|
| Project Title: | | |
| Sheet Title: ET-BASE GSM SIM900 | | Size: A4 |
| Drawn By: ETT CO., LTD. | | REV: |
| Date: 1/23/2013 9:45:58 AM | Sheet: 2/2 | |