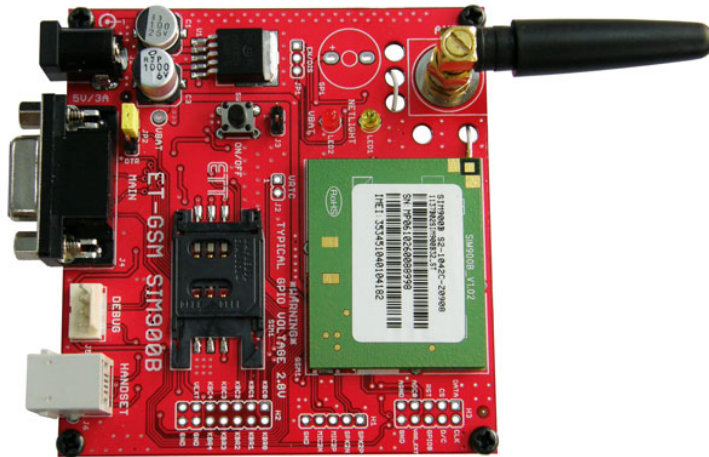


ET-GSM SIM900B



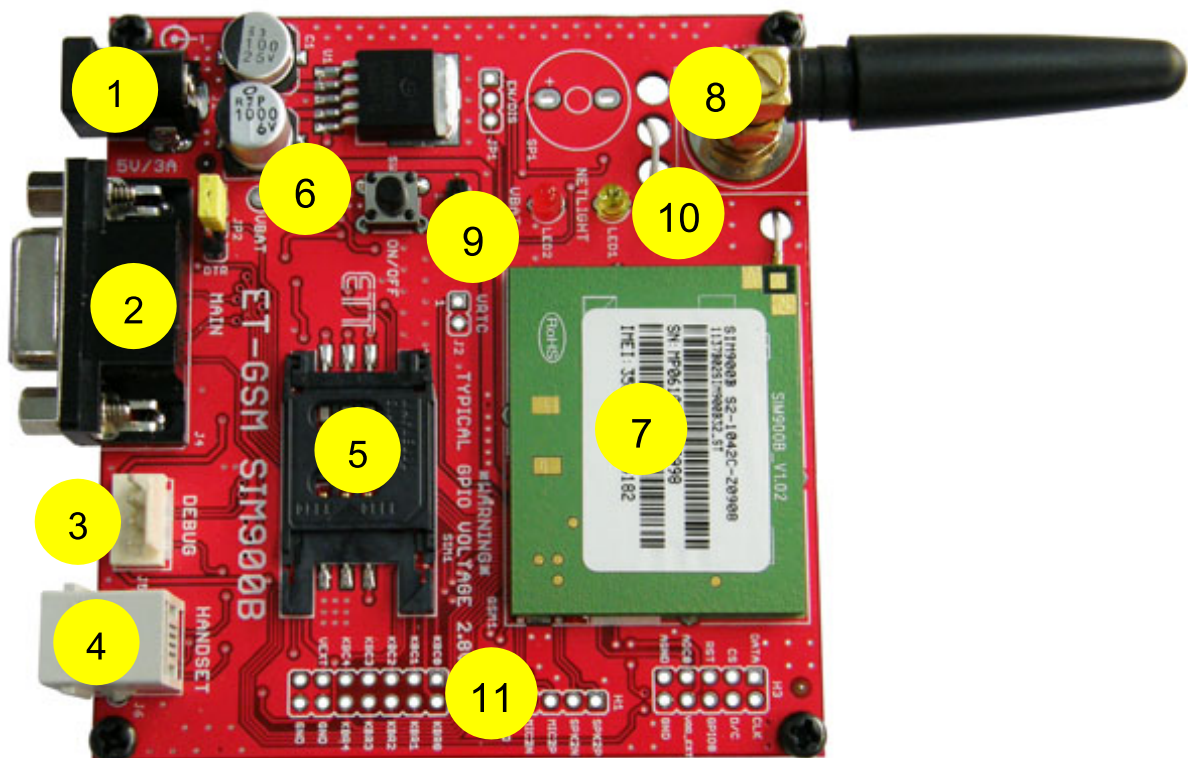
ET-GSM SIM900B เป็นชุดเรียนรู้และพัฒนากระบวนการสื่อสารไร้สาย โดยใช้โมดูล GSM/GPRS รุ่น SIM900B ของ “SIMCom Ltd.” เป็นอุปกรณ์หลัก ซึ่ง SIM900B เป็นโมดูลสื่อสารระบบ GSM/GPRS ขนาดเล็ก รองรับระบบสื่อสาร GSM ความถี่ 850/900/1800/1900MHz โดยส่งงานผ่านทางพอร์ตสื่อสารอนุกรม RS232 ด้วยชุดคำสั่ง AT Command สามารถประยุกต์ใช้งานได้มากมายหลายรูปแบบ ไม่ว่าจะเป็นการรับส่งสัญญาณแบบ Voice, SMS, Data, FAX และยังรวมถึงการสื่อสารด้วย Protocol TCP/IP ด้วย ซึ่งตามปกติแล้ว ถึงแม้ว่าโมดูล SIM900B จะมีวงจร และ Firmware บรรจุไว้ภายในตัวเป็นที่เรียบร้อยแล้ว ก็ยังไม่สามารถนำไปใช้งานได้โดยตรงทันที เนื่องจากในการใช้งานจริงนั้น ผู้ใช้งานเองจำเป็นต้องออกแบบวงจรรอบนอกที่จำเป็นมาเชื่อมต่อกับขาสัญญาณของตัวโมดูลอีกในบางส่วน ไม่ว่าจะเป็นวงจรภาค Power Supply, วงจรเชื่อมต่อกับ SIM Card รวมไปถึงวงจร Line Driver ของ RS232 เป็นต้น ดังนั้นทางทีมงาน อีทีที จึงได้จัดสร้างบอร์ดสำหรับเป็นตัวกลางในการเชื่อมต่อระหว่างโมดูล SIM900B กับอุปกรณ์ภายนอก เพื่อให้ผู้ใช้งานสามารถนำโมดูล GSM ของ SIM900B ไปทำการทดลองและศึกษาเรียนรู้การสั่งงานต่างๆได้โดยสะดวก ก่อนที่จะนำเอาโมดูลตัวนี้ไปออกแบบดัดแปลงและประยุกต์ใช้งานในด้านต่างๆได้ต่อไปในอนาคต ซึ่งถึงแม้ว่าวงจรการเชื่อมต่อทั้งหมดที่ทาง อีทีที ได้จัดทำขึ้นมานี้จะยังไม่สามารถรองรับการใช้งานทรัพยากรต่างๆที่มีอยู่ภายในโมดูลได้ครบถ้วนทั้งหมดก็ตามที แต่ในส่วนของการใช้งานโมดูลในส่วนที่เป็นความสามารถหลักๆที่จำเป็นนั้นก็มีไว้รองรับอย่างครบถ้วนเพียงพอแล้ว

อย่างไรก็ตามถ้าผู้ใช้งาน ต้องการพัฒนา Application ที่สูงขึ้นไป ก็สามารถประยุกต์ดัดแปลงหรือทำการเชื่อมต่ออุปกรณ์เพิ่มเติมให้กับบอร์ดได้โดยง่าย ทั้งนี้ก็เพราะว่าขาสัญญาณต่างๆจากโมดูล ในส่วนที่ยังไม่ได้ทำการออกแบบวงจรเตรียมไว้ให้ภายในบอร์ด เช่น ขาสัญญาณสำหรับเชื่อมต่อกับ Keyboard , LCD Display และ GPIO ต่างๆนั้น ทางอีทีที เองก็ได้จัดทำเป็นจุดต่อ Connector เตรียมไว้ให้เป็นที่เรียบร้อยแล้วผู้ใช้เพียงแต่ทำการเชื่อมต่อสัญญาณต่างๆ จากจุดเชื่อมต่อที่เตรียมไว้ไปยังวงจรส่วนที่ได้ทำการออกแบบไว้ได้โดยสะดวกอยู่แล้ว

คุณสมบัติของบอร์ด ET-GSM SIM900B

- มีสวิตช์แบบ Push-Button สำหรับใช้สั่ง เปิด-ปิด การทำงานของโมดูลภายในบอร์ด
- มี Socket SIM รองรับ SIM Card พร้อมวงจร ESD ป้องกัน SIM เสียหาย
- มีวงจร Regulate แยกอิสระ จำนวน 2 ชุด สามารถใช้กับแหล่งจ่ายไฟ Adapter ขนาดตั้งแต่ +5V ขึ้นไป สามารถจ่ายกระแสให้กับโมดูล SIM900B และอุปกรณ์เชื่อมต่อต่างๆได้อย่างเพียงพอ
 - มีวงจร Regulate ขนาด 4.2V / 3A สำหรับจ่ายให้กับโมดูล SIM900B ได้อย่างเพียงพอสามารถใช้กับ SIM ของระบบ GSM900MHz แบบ 2-Watt ได้อย่างไม่เกิดปัญหา
 - มีวงจร Regulate ขนาด 2.8V / 150mA สำหรับจ่ายให้กับวงจรแปลงระดับสัญญาณลอจิก
- มีวงจร Line Driver สำหรับแปลงระดับสัญญาณลอจิกจากโมดูล SIM900B ให้เป็น RS232 ระดับมาตรฐานครบทุกเส้นสัญญาณ ทั้งพอร์ตที่ใช้ในการสื่อสารสำหรับสั่งงานโมดูล และ พอร์ตสำหรับการใช้ในการพัฒนาโปรแกรม (Debug) สามารถเชื่อมต่อกับพอร์ต RS232 มาตรฐานได้ทันที
- 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
- มีจุดยึดเสาอากาศ สำหรับใช้เป็นจุดพักสำหรับเชื่อมต่อกับเสาอากาศแบบต่างๆได้โดยสะดวก
- มีขั้วต่อสำหรับติดตั้งโมดูล SIM900B พร้อมเสารองและสกรูยึดโมดูลกับตัวบอร์ด
- มีจุดต่อสัญญาณอื่นๆที่เหลือจากโมดูล เช่น Keyboard, Display ,GPIO ฯลฯ สำหรับให้ผู้ใช้งานขยายไปยังวงจรที่ออกแบบเพิ่มเติมได้โดยง่ายและสะดวก

โครงสร้างของบอร์ด ET-GSM SIM900B



- หมายเลข 1 เป็น JACK DC-IN แบบมีขั้ว โดยมีด้านนอกเป็นขั้วบวก และด้านในเป็น GND ใช้สำหรับรับแหล่งจ่ายไฟจากภายนอกโดยออกแบบให้ใช้กับ แหล่งจ่ายไฟขนาด 5V ขึ้นไปที่จ่ายกระแสได้ 1A ถึง 3A
- หมายเลข 2 เป็น ขั้วต่อ RS232(DCE) แบบ DB9 ตัวเมีย สำหรับใช้เชื่อมต่อกับสัญญาณ RS232(DTE) แบบ DB9 ตัวผู้ จากคอมพิวเตอร์ PC หรืออุปกรณ์ภายนอกอื่นๆ โดยใช้สาย 9 Pin แบบต่อตรง
- หมายเลข 3 เป็น ขั้วต่อ DEBUG ใช้สำหรับพัฒนา และ DEBUG โปรแกรม สำหรับต่อกับ RS232 ในกรณีที่ต้องการพัฒนาโปรแกรมเพิ่มเติมให้กับโมดูล SIM900B เอง
- หมายเลข 4 เป็น ขั้วต่อ RJ11 สำหรับใช้เชื่อมต่อกับชุด Handset ในกรณีที่ต้องการใช้งานโมดูล SIM900B เพื่อโทรออกและรับสาย โดยสามารถเชื่อมต่อกับ Handset มาตรฐานได้ทั่วไป
- หมายเลข 5 เป็น Socket สำหรับติดตั้ง SIM Card ให้กับโมดูล
- หมายเลข 6 เป็น Switch Push-Button สำหรับใช้ Power-ON และ Power-OFF ตัวโมดูล
- หมายเลข 7 โมดูล SIM900B
- หมายเลข 8 เป็น จุดยึด Connector เส้าอากาศ GSM/GPRS ย่านความถี่ 900/1800/1900MHZ

- หมายเลข 9 เป็น LED แสดงแหล่งจ่าย VBAT โดยจะติดสว่างเมื่อมีการจ่ายไฟให้กับบอร์ดแล้ว
- หมายเลข 10 เป็น LED NETLIGHT สีเหลืองจะกะพริบเมื่อโมดูลอยู่ในสถานะ POWER ON
- หมายเลข 11 เป็น จุดต่อสัญญาณเพิ่มเติมในกรณีที่ต้องการประยุกต์ใช้งานโมดูลเพิ่มเติม

คุณสมบัติของโมดูล SIM900B

- 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 3V และ 1.8V
 - มีวงจร Analog Audio (MIC & Speaker) จำนวน 2 ชุด
 - รองรับ 5x4 Keypad Interface & SPI LCD Interface
 - มีระบบ RTC พร้อมวงจร Backup
 - มีหัวต่อเสาสื่ออากาศภายนอกแบบ Connector และจุดเชื่อมต่อแบบ PAD

อุปกรณ์แสดงการทำงานของโมดูล SIM900B

สำหรับบอร์ด ET-GSM SIM900B นั้น ได้ออกแบบอุปกรณ์แสดงผลการทำงานของบอร์ดไว้ในบอร์ดเพื่อใช้แสดงสถานะของการทำงานต่างๆให้ผู้ใช้งานทราบด้วย คือ

- LED VBAT ใช้ทำหน้าที่แสดงสถานะของแหล่งจ่ายไฟจากภายนอกที่ต่อมาให้กับบอร์ด โดย LED นี้จะติดสว่างก็ต่อเมื่อมีการจ่ายไฟให้กับบอร์ดเป็นที่เรียบร้อยแล้ว
- LED NETLIGHT ใช้แสดงสถานะของโมดูล ในขณะที่ทำการเชื่อมต่อกับเครือข่ายอยู่ โดย LED ตัวนี้ จะถูกควบคุมด้วยสัญญาณ NETLIGHT(Pin30) ของโมดูล SIM900B เมื่อทำงานจะมีสถานะทางลอจิกเป็นลอจิก “1” โดยเมื่อโมดูลอยู่ในสถานะพร้อมทำงาน LED นี้จะติดกะพริบด้วยความถี่ต่างๆ ซึ่งมีความหมายดังนี้
 - OFF แสดงว่าโมดูลอยู่ในสถานะของ Power OFF (ไม่ทำงาน)
 - 64mS ON / 800mS OFF แสดงว่า โมดูล SIM900B ไม่สามารถการค้นหาเครือข่ายได้
 - 64mS ON / 3000mS OFF แสดงว่าโมดูล SIM900B สามารถการค้นหาเครือข่ายได้

- 64mS ON / 300mS OFF แสดงว่าโมดูล SIM900B อยู่ระหว่างการเชื่อมต่อกับเครือข่าย หรืออุปกรณ์อื่นๆ ด้วย GPRS อยู่

การสั่ง เปิด และ ปิด การทำงานของโมดูล

ตามปกติแล้ว โมดูล SIM900B จะมีโหมดการทำงานอยู่หลายโหมด เราสามารถทำงานสั่ง เปิด และ ปิดการทำงานของโมดูลได้ หลายวิธี

- Switch ON/OFF เป็นการสั่ง เปิด และ ปิด การทำงานของโมดูล SIM900B ด้วยการกดสวิตช์ โดย สวิตช์ตัวนี้ จะเป็นแบบ Push-Button Switch (สวิตช์ กดติด-ปล่อยดับ) โดยเป็นการกำหนดสถานะทางโลจิกให้กับขาสัญญาณ PWRKEY(Pin34) ของโมดูล โดยเมื่อกดสวิตช์จะเป็นโลจิก “0” เมื่อปล่อยสวิตช์จะเป็นโลจิก “1” โดยการทำงานของสวิตช์จะต้องทำการกดสวิตช์ต่อเนื่องกันเป็นเวลาประมาณ 1000mS (1 วินาที) จึงจะมีผลต่อการทำงานของโมดูล โดยลักษณะการทำงานของสวิตช์ จะเป็นแบบ Toggle กล่าวคือ ถ้าโมดูลอยู่ในสถานะของ Power OFF อยู่ แล้วทำการกดสวิตช์ เป็นเวลาประมาณ 1000mS (1 วินาที) จะเป็นการสั่งให้โมดูลกลับเข้าสู่ Power On หรือพร้อมทำงาน แต่ถ้าหากว่าโมดูลอยู่ในสถานะของ Power ON อยู่ แล้วทำการกดสวิตช์ เป็นเวลาประมาณ 1000mS (1 วินาที) แล้วปล่อยจะเป็นการสั่งให้โมดูลหยุดทำงานและกลับเข้าสู่สถานะของ Power OFF (หยุดทำงาน)

LED สถานะ	Power-ON	Power-OFF
VBAT (แดง)	ติดสว่าง	ติดสว่าง
NETLIGHT (เหลือง)	กระพริบ	ดับ

ตาราง แสดงสถานะของ LED ในโหมดต่างๆ

หลังจากทำการสั่ง Power-ON ในครั้งแรกนั้น ก่อนที่จะเริ่มต้นส่งคำสั่งใดๆให้กับโมดูล ควรรอให้ตัวโมดูลพร้อมเสียก่อน โดยจะมีข้อความ “IIII” (ค่า HEX ที่ส่งออกมาคือ “ 00 49 49 49 49 FF FF FF FF”) ปรากฏให้เห็น ในกรณีที่กำหนด Baudrate เป็นแบบ Auto Baudrate ไว้ (AT+IPR=0) เมื่อทำการ Power-ON จะได้ผลดังตัวอย่าง

IIII

ในกรณีที่กำหนด Baudrate เป็นแบบ Fix Baudrate ไว้(AT+IPR=ค่า Baudrate) เมื่อทำการ สั่งให้โมดูล Power-ON แต่ละครั้งจะได้ผลดังตัวอย่าง

IIII

RDY

+CFUN: 1

+CPIN: READY

Call Ready

หมายเหตุ กรณีที่กำหนด Baudrate เป็นแบบ Auto Baudrate ไว้ โมดูล SIM900B จะรับรู้ Baudrate ที่ใช้โดยการส่งตัวอักษร AT ตัวพิมพ์ใหญ่ให้กับโมดูลก่อน

การติดต่อสื่อสารกับโมดูล SIM900B

การติดต่อสื่อสารกับโมดูล SIM900B ของบอร์ด ET-GSM SIM900B นั้นจะเชื่อมต่อผ่านพอร์ตสื่อสารอนุกรม RS232 โดยใช้หัวต่อแบบ DB9 ตัวเมีย จัดเรียงสัญญาณตามมาตรฐาน RS232-DCE สามารถนำไปเชื่อมต่อกับสัญญาณ RS232-DTE มาตรฐาน โดยใช้สาย DB9 แบบต่อตรง ได้ทันที โดยสัญญาณทั้งหมดที่ DB9 นี้ได้ผ่านวงจร Line Driver เพื่อแปลงสัญญาณระดับ โวลิจ จากโมดูล ให้เป็นสัญญาณระดับมาตรฐาน RS232 เป็นที่เรียบร้อยแล้ว ซึ่งถ้าต้องการนำไปเชื่อมต่อกับ RS232(Com Port) ของคอมพิวเตอร์ PC ก็สามารถทำการเชื่อมต่อกันโดยตรงได้ทันที โดยไม่ต้องทำการสลักสายสัญญาณใดๆ ทั้งสิ้น โดยสัญญาณเชื่อมต่อทางด้านโมดูล SIM900B นั้น จะมีทั้งหมด 8 เส้นสัญญาณ ซึ่งในการเชื่อมต่อใช้งานนั้น จะต่อให้ครบทั้ง 8 เส้น หรือ จะเลือกต่อเพียง 3 เส้น (RXD, TXD และ GND) ก็ได้เช่นเดียวกัน โดยสามารถกำหนดได้จากการ Setup ค่า Configuration และคำสั่งใช้งาน โดยสัญญาณการเชื่อมต่อ RS232 ด้านโมดูล SIM900B จะมีดังนี้

- Pin1 เป็นขา DCD (Data Carrier Detect) ของโมดูล SIM900B ซึ่งเป็น Output จาก SIM900B ที่ได้ผ่านการแปลงระดับสัญญาณเป็น RS232 แล้ว ซึ่งตามปกติจะต่อเข้ากับ DCD Input ของอุปกรณ์ด้าน Host หรือคอมพิวเตอร์ PC
- Pin2 เป็นขา TXD(Transmit Data) ของโมดูล SIM900B ซึ่งเป็น Output จาก SIM900B ที่ได้ผ่านการแปลงระดับสัญญาณเป็น RS232 แล้ว ซึ่งตามปกติจะต่อเข้ากับ RXD(Receive Data) ของอุปกรณ์ด้าน Host หรือคอมพิวเตอร์ PC
- Pin3 เป็นขา RXD (Receive Data) ของโมดูล SIM900B ซึ่งเป็น Input ของ SIM900B สามารถรับสัญญาณระดับ RS232 ได้โดยตรง ซึ่งตามปกติจะต่อเข้ากับ TXD(Transmit Data) จากอุปกรณ์ด้าน Host หรือคอมพิวเตอร์ PC
- Pin4 เป็นขา DTR(Data Terminal Ready) ของโมดูล SIM900B ซึ่งเป็น Input ของ SIM900B ซึ่งตามปกติจะต่อเข้ากับ DTR จากอุปกรณ์ด้าน Host หรือคอมพิวเตอร์ PC

- Pin5 เป็นสัญญาณ GND ของโมดูล SIM900B ต้องต่อเข้ากับ GND ของอุปกรณ์ด้าน Host หรือ คอมพิวเตอร์ PC
- Pin6 ตามปกติแล้วเป็นสัญญาณ DSR (Data Set Ready) แต่ในกรณีของ SIM900B จะไม่ได้ต่อใช้งาน แต่อย่างไรก็ตาม ในบอร์ดได้ทำการป้อนสัญญาณย้อนกลับหรือ Loop Back สัญญาณ DTR (Data Terminal Ready) ซึ่งเป็น Output ส่งมาจาก Host หรือ คอมพิวเตอร์ PC กลับไปแทน โดยจะถูกต่อไปเข้ากับสัญญาณ DSR Input ของอุปกรณ์ด้าน Host หรือคอมพิวเตอร์ PC
- Pin7 เป็นขาสัญญาณ RTS (Request To Send) ของโมดูล SIM900B ซึ่งเป็น Input ของ SIM900B ซึ่งตามปกติจะต่อเข้ากับ RTS ของอุปกรณ์ด้าน Host หรือคอมพิวเตอร์ PC
- Pin8 เป็นขาสัญญาณ CTS (Clear To Send) ของโมดูล SIM900B ซึ่งเป็น Output จาก SIM900B ซึ่งตามปกติจะต่อเข้ากับ CTS ของอุปกรณ์ด้าน Host หรือคอมพิวเตอร์ PC
- Pin9 เป็นขาสัญญาณ RI(Ring Indicator) ของโมดูล SIM900B ซึ่งเป็น Output จาก SIM900B ซึ่งตามปกติจะต่อเข้ากับ RI ของอุปกรณ์ด้าน Host หรือ คอมพิวเตอร์ PC

DB9 Female(SIM900B)		Signal Direction	DB9 Male (Computer PC)	
Pin	Signal		Signal	Pin
1	DCD	→	DCD	1
2	TXD	→	RXD	2
3	RXD	←	TXD	3
4	DTR	←	DTR	4
5	GND	—	GND	5
6	(DSR)	→	DSR	6
7	RTS	←	RTS	7
8	CTS	→	CTS	8
9	RI	→	RI	9

แผนผัง แสดงการต่อสายสัญญาณระหว่าง ET-GSM SIM900B กับ คอมพิวเตอร์ PC

DB9 Female(SIM900B)		Signal Direction	ไมโครคอนโทรลเลอร์
Pin	Signal		Signal
2	TXD	→	RXD
3	RXD	←	TXD
5	GND	—	GND

แผนผัง แสดงการต่อสายสัญญาณระหว่าง ET-GSM SIM900B กับ ไมโครคอนโทรลเลอร์

คำแนะนำ ในกรณีที่ทำการเชื่อมต่อสัญญาณแบบ 3 เส้น (RXD, TXD, GND) ต้องกำหนดเงื่อนไขของ Flow Control ให้กับโมดูล SIM900B เป็น XON/XOFF โดยใช้คำสั่ง “AT+IFC=1,1”

คุณสมบัติการทำงานของสัญญาณที่ควรรู้

- RI(Ring Indicator) เป็น Output จากโมดูล SIM900B ตามปกติจะเป็น 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 SIM900B เมื่อต้องการให้โมดูลทำงานต้องให้ขาสัญญาณนี้ได้รับโลจิก LOW ถ้าขา DTR ได้รับโลจิก HIGH โมดูลจะหยุดทำงานและเข้าสู่ Sleep Mode โดยอัตโนมัติ (ถ้ามีการสั่ง Enable Sleep Mode ด้วยคำสั่ง “AT+CSCLK=1” ไว้) ดังนั้นถ้าต้องการให้โมดูลทำงานตลอดเวลาต้องควบคุมให้ขาสัญญาณ DTR ด้านโมดูลได้รับโลจิก LOW โดยการควบคุมจากสัญญาณ DTR ด้านคอมพิวเตอร์ PC หรืออุปกรณ์ที่ควบคุมโมดูลอยู่ให้ทำการ Active สัญญาณ DTR ไว้ตลอดเวลาการเชื่อมต่อ สำหรับกรณีที่นำโมดูล SIM900B ไปเชื่อมต่อกับ RS232 ระบบที่ไม่มีสัญญาณ DTR อยู่เช่น RS232 ของไมโครคอนโทรลเลอร์บางรุ่น ก็อาจเลือกกำหนด Jumper (DTR) ที่อยู่ใกล้กับขั้วต่อ DB9 บนบอร์ดไว้ทางด้าน GND เพื่อให้โมดูลทำงานตลอดเวลา หรือสั่งปิดการทำงานของ Sleep Mode โดยใช้คำสั่ง “AT+CSCLK=0” แล้วบันทึกค่า Configuration นี้ไว้ก็ได้เช่นเดียวกัน

- ADC0(Analog to Digital) เป็น Input แบบ ADC (ขา 52 ของโมดูล SIM900B) สามารถรับสัญญาณ Analog จากภายนอกได้ระหว่าง 0V ถึง 2.8V โดยสามารถสั่งอ่านค่าระดับแรงดันที่ขานี้ได้จากคำสั่ง "AT+CADC?" โดยจะได้ผลลัพธ์เป็นค่าระหว่าง 0 ถึง 2400
- GPIO0:GPIO1 เป็นขาสัญญาณ I/O สามารถเชื่อมต่อกับสัญญาณลอจิกระดับ 2.8V

ตัวอย่างการใช้งาน AT Command เพื่อสั่งงานโมดูล SIM900B

โมดูล GSM/GPRS รุ่น SIM900B ถูกออกแบบให้ทำหน้าที่เหมือน Modem โดยจะใช้การติดต่อสั่งงานและสื่อสารกับโมดูล ผ่านทางพอร์ตสื่อสาร RS232 รองรับ Baudrate ตั้งแต่ 1200-115200 BPS โดยใช้ชุดคำสั่งแบบ AT Command ซึ่งจะมีรูปแบบการใช้งานเหมือนกับ Modem มาตรฐานทั่วไป เพียงแต่จะมีการเพิ่มเติม Option และคำสั่งพิเศษอื่นๆเพิ่มเติมขึ้นมาอีก เพื่อให้เหมาะสมและสอดคล้องกับความสามารถในการทำงานของโมดูลได้อย่างครบถ้วน

สำหรับรายละเอียดการใช้คำสั่ง AT Command ที่จะใช้ สำหรับติดต่อสั่งงานโมดูล SIM900B ไม่ว่าจะเป็น รูปแบบคำสั่ง และ หน้าที่การทำงานของแต่ละคำสั่ง ผู้ใช้สามารถศึกษารายละเอียดต่างๆได้จาก คู่มือคำสั่ง AT Command (ไฟล์เอกสารชื่อ SIM900C_ATC_V1.02.PDF) ในแผ่น CD-ROM ซึ่งในที่นี้จะขอแนะนำถึงวิธีการและรูปแบบการใช้งานคำสั่งแบบย่อๆ แบบพอสังเขป เพื่อเป็นแนวทางให้ผู้เริ่มต้นได้ใช้เป็นแนวทางและประกอบความเข้าใจในการศึกษาการทำงานของคำสั่งต่างๆต่อไป โดยรูปแบบของคำสั่งต่างๆ ที่เป็น 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)

การทดสอบการสั่งงานโมดูล

ดังได้ทราบแล้วว่าในการสั่งงานโมดูล SIM900B นั้น จะใช้วิธีการส่งคำสั่งในรูปแบบของ AT Command ผ่านทางพอร์ตสื่อสารอนุกรมไปให้กับโมดูล ซึ่งตามปกติจะต้องเขียนโปรแกรมเพื่อส่งรหัสคำสั่งต่างๆไปให้กับโมดูลเอง ขึ้นอยู่กับว่าจะใช้อุปกรณ์ใดเป็นตัวควบคุมการทำงานของโมดูล ซึ่งไม่ได้จำกัดว่าเป็นอุปกรณ์แบบใด อาจจะเป็นคอมพิวเตอร์ PC หรือ ไมโครคอนโทรลเลอร์ตระกูลใดๆก็ได้ ขอให้มียุติสื่อสารอนุกรม RS232 อยู่ก็สามารถนำมาเชื่อมต่อเพื่อสั่งงานโมดูล SIM900B ได้แล้ว ส่วนที่ว่าจะเขียนโปรแกรมอย่างไร และจะใช้ภาษาใดในการเขียนนั้น ขึ้นอยู่กับผู้พัฒนาโปรแกรมว่า มีความถนัดอย่างไรและมีพื้นฐานอะไรอยู่บ้าง ซึ่งหลักสำคัญก็คือ ผู้พัฒนาต้องหาคำตอบให้ได้ว่า การจะเขียนโปรแกรมสั่งงานอุปกรณ์ทำการ ส่ง และ รับ ข้อมูล ด้วยพอร์ตสื่อสารอนุกรม RS232 นั้นจะต้องทำอย่างไร ซึ่งจะไม่ขอกล่าวถึงในที่นี้ด้วย

สำหรับในการศึกษาเบื้องต้นนั้น ยังไม่จำเป็นต้องใช้วิธีการเขียนโปรแกรมก็ได้ แต่สามารถใช้โปรแกรมสำเร็จรูปจำพวก Serial Terminal ต่างๆของคอมพิวเตอร์เป็นตัวทดสอบการทำงานเพื่อทำความเข้าใจกับรูปแบบคำสั่งและผลของการทำงานต่างๆให้เข้าใจเสียก่อน ตัวอย่างเช่น ถ้าต้องการจะสั่งให้โมดูล SIM900B โทรออกไปยังโทรศัพท์มือถือหมายเลข 0811234567 นั้น ในอันดับแรกจะต้องศึกษารูปแบบการทำงานของคำสั่งให้เข้าใจเสียก่อน จนสามารถเข้าใจแล้วว่าต้องใช้คำสั่ง “ATD0811234567;” เพื่อสั่งให้โทรออก จากนั้นจึงค่อยปรับเปลี่ยนไปเป็นการเขียนโปรแกรมในภายหลัง ซึ่งผู้ใช้ก็ต้องไปศึกษาหาคำตอบต่อไปอีกว่าการที่จะเขียนโปรแกรมเพื่อสั่งให้อุปกรณ์ส่งคำสั่ง “ATD0811234567;” ออกไปทางพอร์ตสื่อสารอนุกรมนั้นต้องทำอย่างไรบ้าง ซึ่งในที่นี้จะขอแนะนำให้ใช้โปรแกรม Hyper Terminal ของ Windows เป็นเครื่องมือในการทดลองในเบื้องต้นไปก่อน โดย Hyper Terminal 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) จะไม่ถูก

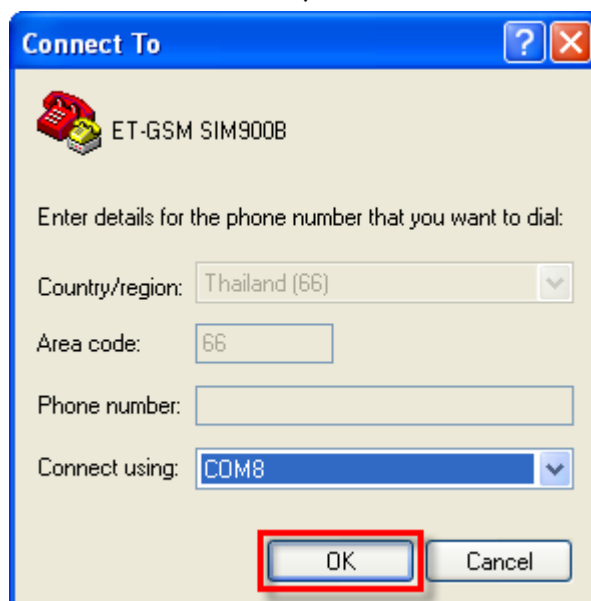
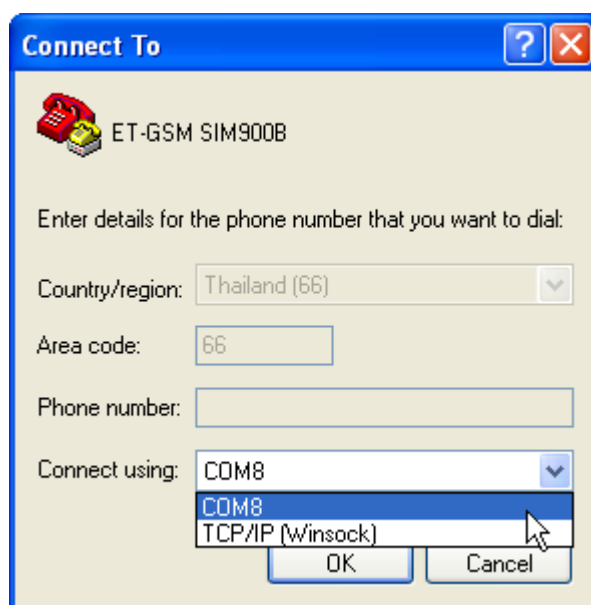
นำมาแสดงผล แต่จะถือว่าเป็นคำสั่ง เช่น เมื่อได้รับ รหัส 0DH โปรแกรม Hyper Terminal จะถือว่าเป็นคำสั่งให้เลื่อน Cursor ของการแสดงผลไว้ในตำแหน่งเริ่มต้นของบรรทัด หรือเมื่อได้รับรหัส 0AH ก็จะทำให้การเลื่อน Cursor ของการแสดงผลให้ขึ้นบรรทัดใหม่แทนดังนี้ เป็นต้น และ ในทางตรงกันข้าม เมื่อเราทำการกดคีย์ใดๆ โปรแกรมก็จะแปลค่าการกดคีย์นั้นให้เป็นรหัส ASCII ของตัวอักษรของตำแหน่งคีย์นั้นๆส่งออกไปยังขา TXD ของพอร์ตสื่อสารอนุกรมโดยอัตโนมัติ โดยการใช้งานโปรแกรม สามารถทำได้ดังนี้

Start → Programs → Accessories → Communication → Hyper Terminal

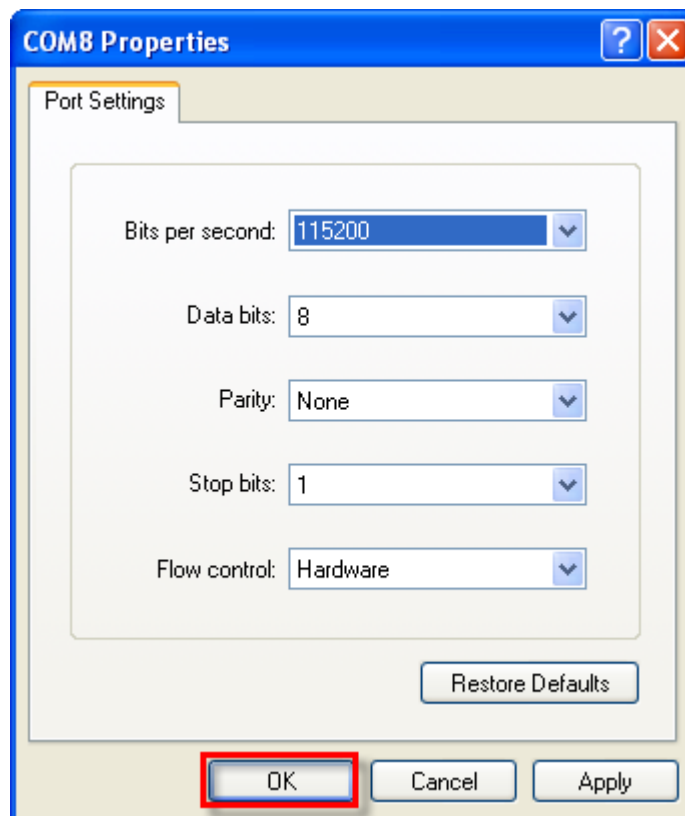
ในขั้นตอนแรกจะปรากฏหน้าต่าง Connection Description ขึ้นมา ให้คลิกเมาส์เลือกรูปแบบของ ICON และกำหนดชื่อของการเชื่อมต่อตามต้องการ แล้วเลือก “OK” ดังตัวอย่าง



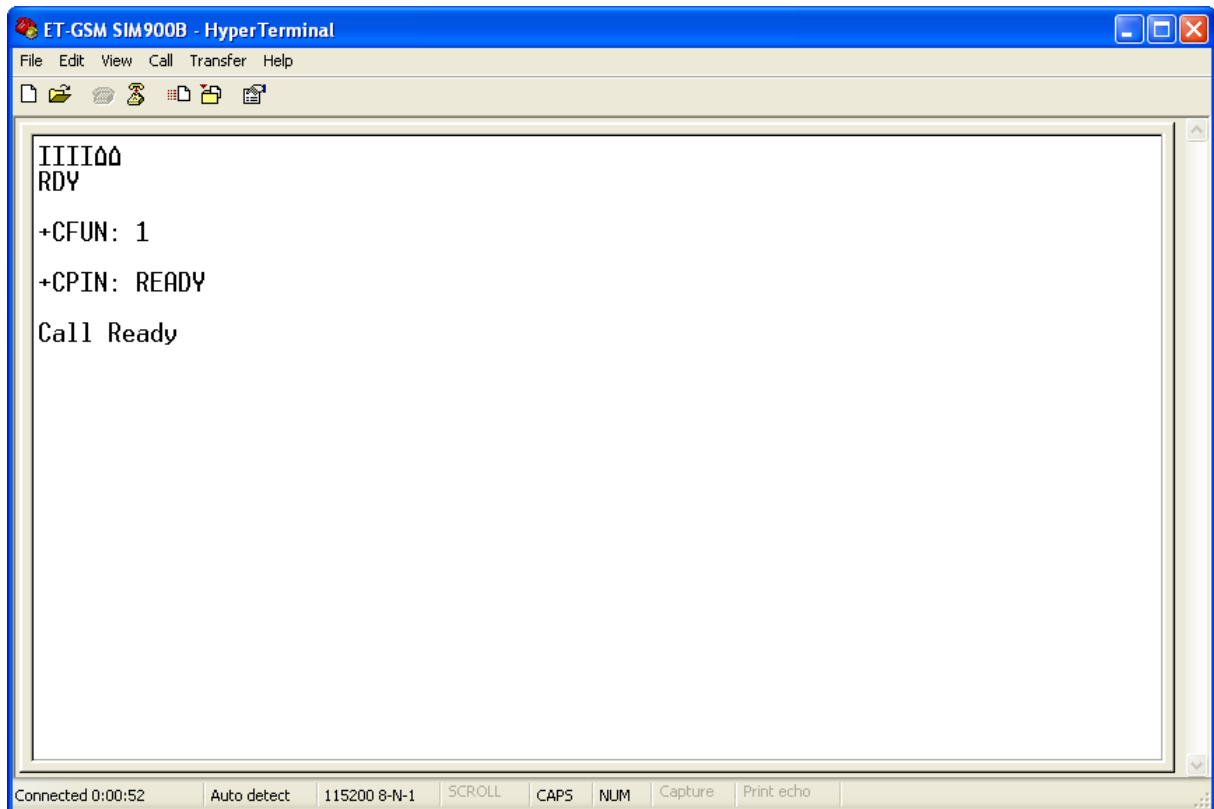
ในขั้นตอนนี้ให้คลิกเมาส์ที่ Connect using แล้วเลื่อนเมาส์ไปยังหมายเลข Comport ที่ใช้ในการเชื่อมต่อกับบอร์ดตามจริง แล้วเลือก “OK” ดังตัวอย่าง



ในขั้นตอนนี้ให้เลือกค่า Baud rate ให้ตรงและสอดคล้องกับที่กำหนดให้กับโมดูลไว้ หรือในกรณีที่กำหนดค่า Baud rate ของโมดูลเป็น Auto-Baud rate ไว้ก็สามารถกำหนดค่าใดๆ ที่โมดูลสามารถรองรับได้ระหว่าง 1200,2400,4800,9600,14400,19200,28800,38400,57600,115200 ส่วน Data ให้เลือกเป็น 8 Bit ,Parity =None, Stop bits=1, Flow Control = Hardware แล้วเลือก “OK” ดังตัวอย่าง



ซึ่งหลังจากกำหนดการเชื่อมต่อต่างๆเรียบร้อยแล้ว ถ้าทุกอย่างถูกต้องให้ทดลองทำการต่อสายสัญญาณ RS232 ระหว่างบอร์ดกับ Comport ของคอมพิวเตอร์ PC แล้วจ่ายไฟเลี้ยงวงจรให้กับบอร์ด ซึ่งถ้าทุกอย่างถูกต้องจะเห็น LED VBAT สีแดงบนบอร์ดติดสว่างให้เห็น จากนั้นให้สั่ง Power-ON ตัวโมดูล โดยการกดสวิตช์ ON/OFF ค้างไว้ประมาณ 1 วินาที จะสังเกตเห็น LED POWER และ STATUS ติดสว่างขึ้น จากนั้น LED NETLIGHT ก็จะเริ่มกะพริบเป็นจังหวะตลอดเวลา แสดงว่าโมดูลเริ่มต้นทำงานแล้ว ส่วนที่หน้าจอของ Hyper Terminal จะปรากฏข้อความการทำงานให้เห็น ให้รอจนพบคำว่า "Call Ready" ซึ่งหมายถึงโมดูลทำการค้นหาและเครือข่ายได้เรียบร้อยแล้ว จากนั้นก็จะสามารถส่งงานโมดูลด้วยคำสั่งต่างๆได้ตามต้องการดังตัวอย่าง



รูปแสดงลักษณะของหน้าจอ Hyper Terminal เมื่อโมดูลพร้อมทำงาน

ตัวอย่างการใช้งานคำสั่ง

สมมติว่าเราต้องการใช้คำสั่งสำหรับกำหนดค่าอัตราความเร็วของการสื่อสารของโมดูล ซึ่งจะต้องใช้คำสั่ง “AT+IPR” โดยเราสามารถสั่งงานคำสั่งนี้ได้หลายรูปแบบดังตัวอย่างต่อไปนี้ คือ

ถ้าเราจำไม่ได้ว่าค่าพารามิเตอร์ของคำสั่งหรืออัตรา Baudrate ที่สามารถกำหนดได้ มีค่าอะไรบ้าง และกำหนดอย่างไร เราก็สามารถใช้รูปแบบการทดสอบคำสั่ง โดยใช้คำสั่ง “AT+IPR=?” และจบด้วย Enter เพื่อสอบถามได้ โดยโมดูลจะตอบรับด้วย “+IPR:” พร้อมกับพิมพ์ค่าพารามิเตอร์ของคำสั่งที่มีอยู่ทั้งหมดให้ทราบ คือ 0,300,1200,...,115200 ดังตัวอย่าง

```
AT+IPR=?<Ent>
```

```
+IPR: (),(0,1200,2400,4800,9600,14400,19200,28800,38400,57600,115200)
```

```
OK
```


ถ้าต้องการทราบว่าในขณะนี้ ค่า Baudrate 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-Baudrate โดยโมดูลจะทำการปรับค่าความเร็ว Baudrate โดยอัตโนมัติ ในตอนเริ่มต้นการทำงานครั้งแรก แต่ถ้าเราต้องการกำหนดค่า Baudrate เป็นค่าคงที่ไปเลย เพื่อให้โมดูลใช้อัตราความเร็วนี้ตลอด ก็สามารถใช้รูปแบบคำสั่ง กำหนดค่าการทำงาน ได้ เช่น ถ้าต้องการกำหนด Baudrate เป็น 115200 ก็จะใช้รูปแบบคำสั่งเป็น “AT+IPR=115200” ดังตัวอย่าง

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

```
OK
```

ซึ่งหลังจากสั่งเปลี่ยนค่า Baudrate เป็น 115200 แล้ว ต่อจากนี้ไปก็สามารถสื่อสารกับโมดูลด้วยค่าความเร็วนี้ได้ตลอด แต่ถ้าสั่ง รีเซ็ตโมดูลใหม่ ค่า Baudrate ก็จะกลับไปเป็นแบบ Auto-Baudrate อีก เนื่องจากเรายังไม่ได้สั่งบันทึกค่าให้เป็น Configuration ถาวรไว้ ซึ่งสามารถทำได้โดยใช้คำสั่ง “AT&W”

```
AT&W<Ent>
```

```
OK
```

การกำหนด Flow Control

โมดูล SIM900B สามารถกำหนด Flow Control หรือ รูปแบบการตรวจสอบความพร้อมในการสื่อสารและรับส่งข้อมูลได้ด้วย ซึ่ง Flow Control จะมีความสำคัญเป็นอย่างมาก เนื่องจากการประมวลผลของอุปกรณ์ต่างๆ มีความช้าเร็วที่แตกต่างกัน เมื่อมีการรับส่งข้อมูลที่มีจำนวนข้อมูลมากๆ แบบต่อเนื่องนั้น ถ้าฝ่ายรับไม่พร้อมรับข้อมูลแต่ฝ่ายส่งยังคงส่งข้อมูลออกไป ก็จะทำให้ข้อมูลสูญหายและเกิดความผิดพลาดขึ้นได้ โดย SIM900B เองรองรับการตรวจสอบความพร้อมหรือ Flow Control ได้ 2 แบบ คือ

- Software Flow Control (XON/XOFF Flow Control) เป็นการตรวจสอบความพร้อมด้วย Software โดยจะใช้รหัส XOF(13H) เป็นตัวสั่งหยุดการส่งข้อมูลจากฝ่ายส่ง และใช้รหัส XON(11H) เพื่อบอก

หรืออนุญาตให้ฝ่ายส่งเริ่มต้นส่งข้อมูลลำดับต่อไปมายังโมดูลได้ โดยใช้ Flow Control แบบนี้ เหมาะกับการเชื่อมต่อกับอุปกรณ์ที่ไม่มีสัญญาณตรวจสอบความพร้อม 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 XON/XOFF จะใช้คำสั่ง “AT+IFC=1,1”

การกำหนด Flow Control เป็น RTS/CTS จะใช้คำสั่ง “AT+IFC=2,2”

การกำหนด Format ข้อมูลของ RS232

เราสามารถกำหนด Format ของข้อมูล ได้ว่าจะใช้รูปแบบการส่งข้อมูลเป็นอย่างไร ใช้ขนาดข้อมูลเป็นกี่บิต ใช้บิตตรวจสอบความผิดพลาด Parity หรือไม่ และต้องการใช้ Stop Bit เป็นกี่บิต ซึ่งตามปกติทั่วไปแล้วจะใช้ 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”

การ Setup และตรวจสอบค่า Configuration

ตามปกติแล้วการทำงานของโมดูล SIM900B นั้นจะสามารถกำหนดรูปแบบการทำงานได้มากมายหลายลักษณะ เช่น เงื่อนไขในการติดต่อสื่อสารกับโมดูล ผู้ใช้สามารถเปลี่ยนแปลงค่าต่างๆได้มากมาย ไม่ว่าจะเป็นค่า Baud rate หรือรูปแบบของการ Handshake ต่างๆ ที่จะใช้ในการสื่อสาร เป็นต้น ดังนั้นจึงจำเป็นต้องมีการกำหนดรูปแบบการทำงานของโมดูลให้ตรงกับความต้องการ ซึ่งตามปกติแล้วเงื่อนไขต่างๆเหล่านี้จะมีค่าที่แน่นอนอยู่ค่าหนึ่งเสมอหลังการรีเซ็ต หรือ Power ON โดยโมดูลจะกำหนดค่าเงื่อนไขต่างๆให้กับตัวมันเองในตอนเริ่มต้นการทำงานด้วยค่าที่กำหนดไว้ใน Configuration ที่ถูกบันทึกไว้แล้ว แต่อย่างไรก็ตามผู้ใช้สามารถเปลี่ยนแปลงแก้ไขค่า Configuration ต่างๆได้เองตามต้องการ ซึ่งวิธีการกำหนดเงื่อนไขการทำงานให้กับโมดูลนั้นสามารถทำได้ 2 แบบ

- **การกำหนดค่าแบบถาวร** จะเป็นการสั่งบันทึกค่าเงื่อนไขการทำงานต่างๆของโมดูลตามรูปแบบที่เรากำหนดไว้ในหน่วยความจำถาวรภายในตัวโมดูล โดยใช้คำสั่ง “AT&W” ซึ่งหลังจากโมดูลเริ่มต้น

ทำงานใหม่ หรือ หลังการรีเซ็ตโมดูลแต่ละครั้ง ค่าการทำงานต่างๆของโมดูลจะถูกกำหนดเงื่อนไขตามที่เรากำหนดไว้แล้วเสมอ

- **การกำหนดค่าแบบชั่วคราว** เป็นการใช้คำสั่ง AT Command ต่างๆ เพื่อกำหนดเงื่อนไขการทำงาน ให้กับโมดูล แต่ไม่มีการสั่งบันทึกค่า Configuration ด้วยคำสั่ง “AT+W” ซึ่งการทำงานของโมดูลก็จะปรับเปลี่ยนไปตามการสั่งงานในขณะนั้นๆ แต่เมื่อสั่งรีเซ็ตการทำงานของโมดูล หรือ มี การ Power ON ใหม่คุณสมบัติการทำงานของโมดูลจะถูกเปลี่ยนกลับเป็นค่าเดิมอีก

โดยเราสามารถใช้คำสั่ง AT Command ในการสั่ง ตรวจสอบ และ บันทึกค่า Configuration ต่างๆให้กับโมดูล SIM900B 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)

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: 15,0
OK
```

การตรวจสอบ รหัสผลิตภัณฑ์

```
ATI<Ent>
SIM900B R11.0
OK
```

การตรวจสอบ รหัสผู้ผลิต

```
AT+GMI<Ent>  
SIMCOM_Ltd  
OK
```

การตรวจสอบ รหัสรุ่น

```
AT+GMM<Ent>  
SIMCOM_SIM900B  
OK
```

การตรวจสอบ รหัส Version Firmware

```
AT+GMR<Ent>  
Revision:1137B04SIM900B32_ST  
OK
```

การตรวจสอบ Serial Number ของโมดูล

```
AT+GSN<Ent>  
012207000000015  
OK
```

การตรวจสอบ รหัสเครือข่าย SIM ผู้ให้บริการ

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

การโทรออก การรับสาย และการยกเลิกการโทร

- ใช้คำสั่ง “ATD” เพื่อส่งโทรออก โดยรูปแบบการใช้คำสั่งให้ตามด้วยเบอร์ปลายทาง
- ใช้คำสั่ง “ATDL” เพื่อส่งโทรออกด้วยหมายเลขโทรออกครั้งสุดท้าย
- ใช้คำสั่ง “ATA” เพื่อรับสายเรียกเข้า โดยเมื่อมีสายเรียกเข้าจะมีเสียงเรียกเข้าที่ หูฟังของ Handset ให้ทราบ ถ้าต้องการรับสายให้ใช้คำสั่ง “ATA” เพื่อรับสายได้ทันที ซึ่งหลังจากส่งรับสายแล้วผู้ใช้จะสามารถพูดคุยกับปลายสายได้ทันที โดยใช้ Handset หรือชุด ปากพูดหูฟังของโทรศัพท์บ้าน
- ใช้คำสั่ง “ATH” เพื่อส่งวางสาย หรือยกเลิกการโทรออก

ตัวอย่างการโทรออก ซึ่งเป็นการสื่อสารด้วย Voice จะต้องป้อนคำสั่งด้วยเครื่องหมายเซมิโคลอน (;) และจบคำสั่งด้วย Enter เช่นถ้าต้องการโทรออกไปยังเบอร์ 0811234567 จะเป็นดังนี้

```
ATD0811234567;<Ent>
```

```
OK
```

ในกรณีที่ส่งโทรออกแล้วไม่มีการรับสาย หรือ สายไม่ว่างโมดูลจะรายงานผลด้วยข้อความ “BUSY” ดังตัวอย่าง

```
ATD0812505187;<Ent>
```

```
OK
```

```
BUSY
```

ตัวอย่างการสั่งตรวจสอบยอดเงินของระบบ 1-2-CALL ซึ่งถ้าเป็นโทรศัพท์มือถือปกติจะใช้วิธีการพิมพ์เครื่องหมาย “*121#” แล้วส่งโทรออก แต่สำหรับโมดูล SIM900B จะต้องใช้คำสั่ง “ATD” สำหรับส่งโทรออกแล้วตามด้วยเครื่องหมายแทน ดังตัวอย่าง

```
ATD*121#<Ent>
```

```
+CUSD: 0,"You have 67.00 baht, valid until 29/04/08",15
```

```
OK
```

ตัวอย่างการรับสายเรียกเข้า เมื่อมีสายเรียกเข้าโมดูล SIM900B จะมีข้อความ “RING” และสร้างเสียงเรียกเข้าเป็นจังหวะที่ หูฟังของ Handset ให้ทราบ ถ้าผู้ใช้ต้องการรับสาย ให้ใช้คำสั่ง “ATA” เพื่อส่งรับสาย หรือใช้คำสั่ง “ATH” เพื่อวางสายหรือยกเลิกไม่รับสาย ดังตัวอย่าง

```
RING
```

```
ATA<Ent>
```

```
OK
```

ตัวอย่างการรับข้อความ SMS

ตามปกติแล้วโมดูล SIM900B 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” เพื่อสั่งแสดงข้อความทั้งหมดที่เก็บไว้ในหน่วยความจำ โดยสามารถเลือกประเภทของข้อความได้ เช่น ข้อความใหม่ ข้อความทั้งหมด
- ใช้คำสั่ง “AT+CMGD” เพื่อสั่งลบข้อความออกจากหน่วยความจำ เช่น ถ้าต้องการสั่งลบข้อความลำดับที่3 ก็ให้ใช้คำสั่งเป็น “AT+CMGD=3”

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

```
+CMTI: "SM",3
AT+CMGR=3<Ent>
+CMGR: "REC UNREAD","+66812505187",,"07/11/19,13:29:25+28"
Hello 12345
OK
```

ถ้ามีการสั่งอ่านข้อความเดิมซ้ำใหม่สถานะของข้อความจะเปลี่ยนเป็น “REC READ” แทน เพื่อแสดงให้ทราบว่าข้อความนี้ถูกอ่านไปแล้วดังตัวอย่าง

```
AT+CMGR=3<Ent>
+CMGR: "REC READ","+66812505187",,"07/11/19,13:29:25+28"
Hello 12345
OK
```


ตัวอย่างการส่งข้อความ SMS

ในการส่งข้อความ SMS นั้นจะใช้คำสั่ง “AT+CMGS” ในการสั่งงาน โดยในกรณีที่ให้ Text Mode นั้นให้ใช้รูปแบบคำสั่งเป็น “AT+CMGS=”+เบอร์ผู้รับ” โดยเบอร์ของผู้รับต้องใส่รหัสประเทศนำหน้าแทนศูนย์ด้วยเสมอ ซึ่งในกรณีที่เป็นประเทศไทยจะใช้รหัสประเทศเป็น “66” ดังนั้นถ้าต้องการส่งข้อความ SMS ให้กับเบอร์ที่ใช้งานอยู่ในประเทศไทย เช่น 081-1234567 ก็จะต้องกำหนดหมายเลขของเบอร์ผู้รับปลายทางเป็น 6681-1234567 แทน ซึ่งในกรณีนี้จะได้รับรหัสเบอร์ผู้รับข้อความเป็น “+66811234567” ซึ่งเมื่อโมดูล SIM900B ได้รับคำสั่ง “AT+CMGS” เรียบร้อยแล้วมันจะตอบรับด้วยการส่งเครื่องหมาย “>” กลับมาบอก ซึ่งหลังจากนี้เป็นต้นไปผู้ใช้อีกก็สามารถจะทำการพิมพ์ข้อความต่างๆที่ต้องการจะส่งให้กับโมดูลได้ทันที โดยให้ปิดท้ายข้อความด้วยรหัส “Ctrl+Z” ตามด้วย “Enter” เช่นถ้าต้องการส่งข้อความ SMS ให้กับหมายเลข 0811234567 ด้วยข้อความ “Hello Test SMS” จะเป็นดังนี้

```
AT+CMGS="+66811234567"<Ent>
> Hello Test SMS<Ctrl+Z><Ent>
+CMGS: 6
OK
```

รหัสข้อความ SMS ภาษาไทย

สำหรับข้อความ SMS ที่เป็นภาษาไทยนั้น จะไม่สามารถแสดงผลด้วยโปรแกรม Terminal ปรกติได้ ทั้งนี้ก็เนื่องมาจากว่าระบบตัวอักษรที่ใช้ในโปรแกรม Terminal นั้นจะใช้รหัส ASCII ปรกติที่มีขนาดเพียง 1 ไบท์ แต่สำหรับรหัสภาษาไทยที่ใช้ในระบบสื่อสารของโทรศัพท์มือถือต่าง ๆ นั้น จะใช้รหัสพิเศษเฉพาะที่เรียกว่า “Unicode” ซึ่งตัวอักษร 1 ตัวจะประกอบไปด้วยข้อมูลจำนวน 2 ไบท์ โดยรหัส Unicode ของภาษาไทยนั้นจะมีค่าอยู่ระหว่าง 0E00H...0E7FH สำหรับภาษาอังกฤษนั้นถ้าเป็น Unicode จะใช้รหัสตัวอักษรขนาด 2 Byte เช่นเดียวกันกับภาษาไทย โดยจะมีค่าอยู่ระหว่าง 0000H..007FH โดยตามปรกติแล้วถ้าข้อความที่เป็นภาษาอังกฤษอย่างเดีย্বরหัสของตัวอักษรที่ใช้ใน SMS จะเป็นแบบ ASCII คือ ใช้รหัส ขนาด 1 ไบท์ โดยตัวรหัส 00H ไบท์แรกใน Unicode ที่ทิ้งไป เช่น A แทนที่จะเป็นรหัส 0041H ก็เหลือเพียง 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	ฌ	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	ฎ	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	ิ	ี	ึ	ุ	เ	อ	โ	ๅ	ๆ	็	่	้
0E30	0E31	0E32	0E33	0E34	0E35	0E36	0E37	0E38	0E39	0E3A	0E3B	0E3C	0E3D	0E3E	0E3F
เ	แ	โ	ใ	ไ	า	ๆ	๔	๕	๖	๗	๘	๙	๐	๑	๒
0E40	0E41	0E42	0E43	0E44	0E45	0E46	0E47	0E48	0E49	0E4A	0E4B	0E4C	0E4D	0E4E	0E4F
๓	๔	๕	๖	๗	๘	๙	๐	๑	๒	๓	๔	๕	๖	๗	๘
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

หลักการถอดรหัสตัวอักษร Unicode

สำหรับรหัสตัวอักษรที่เป็น Unicode นั้น จะเห็นได้ว่าแต่ละตัวอักษรจะประกอบไปด้วยรหัส Code จำนวน 2 ไบต์เสมอ โดยตัวแรกเป็นตัวบอกรหัส Table ว่าเป็น Unicode ของภาษาใด โดยถ้าเป็นรหัส Unicode ของภาษาอังกฤษ ไบต์แรกจะมีค่าเป็น 00H ส่วนไบต์ที่ 2 จะเป็นรหัสตัวอักษร ซึ่งมีค่าตรงกันกับรหัส ASCII ส่วนภาษาไทยนั้น ไบต์แรกจะมีค่ารหัสเป็น 0EH ส่วนไบต์ที่ 2 จะเป็นรหัสตัวอักษร ซึ่งจากการทดสอบรับข้อความรหัสตัวอักษรจาก SMS พบว่า ถ้าใช้ภาษาอังกฤษอย่างเดียว รหัสของตัวอักษรจะเป็นแบบรหัส ASCII คือ 1 ตัวอักษร จะมีรหัส 1 ไบต์ แต่เมื่อมีการใช้ข้อความที่มีทั้งภาษาไทยและภาษาอังกฤษรวมกันพบว่ามีรหัสตัวอักษรภาษาอังกฤษเป็นแบบ Unicode ด้วย

ดังนั้นจึงน่าจะพอสรุปได้ว่า ถ้าใช้ข้อความที่เป็นภาษาไทย ในระบบ SMS จะใช้รหัสตัวอักษรที่เป็นแบบ Unicode เสมอ แต่สำหรับภาษาอังกฤษนั้น ในระบบโทรศัพท์จะสามารถเลือกใช้ได้ทั้งระบบ Unicode และ ASCII Code โดยถ้าเป็น Unicode จะใช้รหัสตัวอักษรขนาด 2 Byte เช่นเดียวกับภาษาไทย โดยจะมีค่าอยู่ระหว่าง 0000H..007FH โดยมีรหัส 00H เป็นข้อมูลไบต์แรก ซึ่งถ้าข้อความเป็นภาษาอังกฤษอย่างเดียว รหัสของตัวอักษรที่ใช้ใน SMS จะเป็นแบบ ASCII คือ ใช้รหัส ขนาด 1 ไบต์ โดยตัวรหัส 00H ไบต์แรกใน Unicode ที่ไป เช่น A แทนที่จะเป็นรหัส 0041H ก็จะเหลือเพียง 41H เป็นต้น แต่สำหรับข้อความที่มีทั้งภาษาไทยและภาษาอังกฤษรวมกันพบว่ามีรหัส Code ตัวอักษรเป็นแบบ Unicode ด้วยเช่นเดียวกับภาษาไทย

ดังนั้นในการถอดรหัสตัวอักษรต้องพิจารณาถึงจุดนี้ด้วย โดยมีข้อสังเกตว่า ถ้าพบรหัสตัวอักษรที่มีค่าระหว่าง 20H-7FH แสดงว่าเป็นรหัสแบบ ASCII สามารถนำไปแสดงผลได้เลย แต่ถ้าพบว่ารหัสเป็น 00H แสดงว่าเป็นรหัสแบบ Unicode ภาษาอังกฤษ ซึ่งรหัส Code ที่เป็นรหัสตัวอักษรจะอยู่ในรหัสข้อมูลไบต์ถัดไป และถ้าพบรหัสเป็น 0EH แสดงว่าเป็นรหัส Unicode ภาษาไทย ซึ่งรหัส Code ที่เป็นรหัสตัวอักษรจะอยู่ในข้อมูลไบต์ถัดไป เช่นเดียวกัน

ตัวอย่างเช่น ถ้าเราทดลองส่งข้อความ SMS ด้วยคำว่า “สวัสดี Jack” ไปให้กับโมดูล SIM900B และสมมติว่า SIM900B รับข้อความนั้นไว้ และจัดเก็บไว้เป็นข้อความที่ 1 ถ้าใช้โปรแกรม Hyper Terminal หรือ Terminal อื่นๆ ที่แสดงผลเป็น 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 แต่จะเป็นคำสั่งให้เลื่อน Cursor มาไว้ในตำแหน่งเริ่มต้นของบรรทัดและขึ้นบรรทัดใหม่เป็นต้น ซึ่งในที่นี้ผู้เขียนจะขอแสดง

ข้อมูลที่ได้รับในรูปแบบของ 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 มีข้อความส่งเข้ามาใหม่และถูกเก็บไว้ในหน่วยความจำ โดยเป็นข้อความลำดับที่1 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" <...ข้อความที่ได้รับได้...>
--

+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

ซึ่งจากตัวอย่างการทดลองถ้าแสดงผลข้อมูลที่ได้รับได้จากโมดูลในรูปแบบ HEX String ผลการสั่งอ่านข้อความจะได้ผลดังตัวอย่าง

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

เมื่อลองพิจารณาถึงรหัสส่วนที่เป็นข้อความจะเห็นได้ว่า รหัสของข้อความทั้งหมดจะเป็นรหัสแบบ Unicode โดยอักขรตัวแรกจะเป็น 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 ก
 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 SIM900B และสมมติว่า SIM900B 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 มีข้อความส่งเข้ามาใหม่และถูกเก็บไว้ในหน่วยความจำ โดยเป็นข้อความลำดับที่2 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
41 44 22 2C 22 2B 36 36 38 31 31 32 33 34 35 36
37 22 2C 2C 22 30 37 2F 31 31 2F 32 32 2C 31 31
3A 33 34 3A 30 36 2B 32 38 22 0D 0A
48 65 6C 6C 6F 20 4A 61 63 6B 0D 0A
0D 0A
4F 4B 0D 0A
..
+CMGR: "REC UNRE
AD", "+6681123456
7", ", "07/11/22, 11
:34:06+28"..
Hello Jack..
..
OK..
```


ซึ่งจะเห็นได้ว่ารหัสของข้อความใน SMS จะเป็นแบบ ASCII ปกติ 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

ตัวอย่างการส่ง SMS ภาษาไทย ของระบบ TRUE

- SMS Message Center ของ TRUE = "+66891009120"
- ส่งให้เบอร์ "+66811234567"
- ข้อความ "กขค"

มีขั้นตอนดังนี้

Call Ready

AT+CMGF=1

OK

AT+CSCS="UCS2"

OK

AT+CSCA="002B00360036003800390031003000300039003100320030",145

OK

AT+CSMP=17,167,0,24

OK

AT+CMGS="002B00360036003800310031003200330034003500360037"

>0E010E020E04[ctrl+z]

+GMGS: 1

OK

ตัวอย่างการใช้ SIM Command (SIM Application Toolkit : STK)

SIM Application Tool Kit เป็นชุดคำสั่งพิเศษของโมดูล SIM900B ใช้สำหรับสั่งงาน SIM ผ่านทางชุดคำสั่งพิเศษใน SIM บางรุ่นที่รองรับสร้าง Application จากผู้ใช้ได้ ซึ่งตามปกติแล้วผู้ที่ทำการพัฒนา Application ต่างๆให้กับ SIM ได้นั้น จะต้องมีความรู้ความเข้าใจเกี่ยวกับ SIM เป็นอย่างดี โดยข้อกำหนดต่างๆสามารถศึกษาเพิ่มเติมได้จาก Technical Reference ของ "GSM11.14" ซึ่งสำหรับผู้ให้บริการ SIM ในประเทศไทยเอง ในปัจจุบันก็ได้มีการสร้าง Application บรรจุไว้ใน 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 ที่สร้างไว้ใน SIM ตัวอย่างเช่น SIM ของระบบ DTAC ในส่วนที่มีการพัฒนา Application บน SIM ไว้เป็นที่เรียบร้อยแล้ว อันได้แก่ SIM สำหรับผู้ให้บริการเติมเงินมือถือผ่านมือถือ หรือ SIM Happy Online ซึ่งจะเห็นได้ว่าในส่วนของผู้ใช้นั้นไม่จำเป็นต้องทราบรายละเอียดของการทำงานใน Application บน SIM แต่อย่างใด ผู้ใช้งานเพียงแต่ศึกษาข้อกำหนดและวิธีการในการเลือกเมนูและป้อนข้อมูลต่างๆให้ถูกต้องเท่านั้น ตัวอย่างเช่น วิธีการสั่งเติมเงินให้กับมือถือจะมีขั้นตอนดังนี้

1. เข้าสู่เมนู Happy Refill
2. เลือกภาษาเพื่อทำการเติมเงิน
3. เลือกเมนูเติมเงิน
4. ระบุเบอร์มือถือที่ต้องการเติมเงิน
5. เลือกราคาหรือจำนวนเงินในการเติมเงิน
6. ใส่รหัสผ่าน

7. ยืนยันการเติมเงิน

ซึ่งหลังจากที่ผู้ใช้ทำการรายการต่างๆครบทั้ง 7 ขั้นตอนแล้ว โปรแกรม Application ที่บรรจุไว้ภายใน SIM ก็ จะสั่งงานให้โทรศัพท์ส่ง SMS ไปยัง Server ผู้ให้บริการเพื่อดำเนินการเติมเงินให้กับเบอร์ที่ระบุไว้ ซึ่งจะเห็น ได้ว่าในส่วนของผู้ใช้งานเอง จะไม่มีโอกาสทราบได้เลยว่า ข้อความ SMS ที่ส่งออกไปเพื่อร้องขอการเติมเงิน นั้นเป็นอย่างไร และส่งออกไปยังหมายเลขใด ข้อมูลต่างๆที่ผู้ใช้ทำการป้อนเข้าไปในนั้น มีการนำไปแก้ไข ดัดแปลง หรือ ผสมรวมกับข้อมูลอื่นๆ หรือ มีการเข้ารหัสข้อมูลอย่างไรบ้าง ผู้ใช้เองจะไม่มีโอกาสได้ทราบ เลย ทำให้ระบบการทำงาน ถูกปกปิดเป็นความลับ และ มีความปลอดภัย มากขึ้น เพราะไม่ต้องกังวลว่าจะ มีใครพยายามส่งข้อความ SMS ปลอมเข้ามายัง Server ของผู้ให้บริการเติมเงิน เพื่อขอเติมเงินบ้าง

ตัวอย่างการตรวจสอบยอดเงินจากกระเป๋าเงินสด True Money ของ SIM True

เราจะมาทดลองทำความเข้าใจกระบวนการทำงานของ Application ใน SIM กันสัก 1 ตัวอย่าง โดยใน ที่นี้จะขอยกตัวอย่างการใช้ Application ใน SIM ระบบเติมเงิน ของ True ซึ่งได้ทำการ Download Application ของ True Money ให้กับ SIM เรียบร้อยแล้ว โดยในอันดับแรกเราจะทดลองใช้โทรศัพท์มือถือ ในการทำการรายการกันดูก่อน จากนั้นจึงจะทดลองให้โมดูล SIM900B ส่งทำการรายการแบบเดียวกันเพื่อให้ผู้ ใช้ได้เห็นการทำงานจะได้ทำความเข้าใจได้ดีขึ้น โดยในกรณีใช้โทรศัพท์มือถือทำการรายการจะมีขั้นตอนดังนี้

1. เมื่อเลือกเมนูไปยัง True Money จะมีตัวเลือกรายการหัวข้อย่อยใน Menu ให้เลือก 6 ตัวเลือก คือ
 - 1 check balance
 - 2 transfer
 - 3 refill
 - 4 change password
 - 5 register
 - 6 about
2. สมมติว่าเราต้องการตรวจสอบยอดเงิน ก็ต้องเลือกกดคีย์หมายเลข “1” เพื่อเลือกทำการรายการใน Submenu ลำดับที่ 1 ซึ่งก็คือ “check balance”
3. เมื่อโปรแกรม Application ในโทรศัพท์มือถือรับรู้การกดคีย์ มันจะสั่งให้ Application ใน SIM ปฏิบัติงานตามหัวข้อที่ได้เลือกไว้ โดยในกรณีนี้จะปรากฏข้อความ password ที่หน้าจอแสดงผล ของโทรศัพท์และรอรับการกดคีย์เพื่อป้อนรหัสผ่าน ในขั้นตอนนี้ผู้ใช้ต้องทำการป้อนรหัสผ่านเป็น ตัวเลข 4 หลักตามที่ลงทะเบียนกับผู้ให้บริการไว้

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

จากกระบวนการข้างต้นจะเห็นได้ว่าในกรณีที่ผู้ใช้ โทรศัพท์มือถือ เป็นตัวทำรายการนั้นจะทำได้ง่ายมาก เนื่องจากโทรศัพท์มือถือเองมี Application รองรับการทำงานเหล่านี้ไว้อยู่แล้ว และมีคีย์บอร์ดพร้อมหน้าจอสำหรับแสดงรายการต่างๆให้เห็นได้โดยสะดวก แต่สำหรับในกรณีที่ผู้ใช้โมดูล SIM900B เป็นตัวทำรายการนั้น จะไม่สามารถทำได้โดยตรง เพราะเรายังไม่มี Application สำเร็จรูปรองรับการทำงานกับ SIM เหมือนกับโทรศัพท์มือถือ แต่สิ่งที่เราสามารถทำได้ก็คือ การใช้ชุดคำสั่งต่างๆของโมดูล SIM900B สำหรับติดต่อกับ SIM อันได้แก่ การเข้าถึง Application ใน SIM การขอดูรายการตัวเลือกในเมนูของ Application ใน SIM การสั่งให้โมดูลปฏิบัติงานตามหัวข้อ(เลือก Submenu) ต่างๆเหล่านี้เป็นต้น

ตัวอย่าง ขั้นตอนการตรวจสอบยอดเงินของ "True Money" ด้วย SIM900B

AT+CMGF=1<Ent>	;Setup SIM300C = Text Mode
OK	
AT+CMEE=2<Ent>	;Enable result code ,use verbose values
OK	
AT+STRT=200<Ent>	;Setup Response Time Out = 200 Second
OK	
AT+STPD=5,1F7FFF7F7F<Ent>	;Send Setup SIM Tool Kit Command
OK	
+STC: 25	;Wait Setup Menu Command
AT+STGC=25<Ent>	;Read Parameter of Setup Menu Command
+STGC: 25,8,0,0,0,"Menu"	;Display SIM Main Menu
+STGC: 5,"Fun & Smart"	
+STGC: 6,"Setting2Play"	
+STGC: 8,"True Money"	
+STGC: 9,"True Product"	
+STGC: 10,"True Payment"	
+STGC: 12,"True Transfer"	
+STGC: 11,"Other Service"	
+STGC: 13,"Mobile Top Up"	
OK	
AT+STCR=25,0<Ent>	;Request Response of Setup Menu
OK	
+STC: 81	;End of proactive session
AT+STMS=8<Ent>	;Select Sub Menu = 8 (True Money)
OK	
+STC: 24	;Wait Select Item Command
AT+STGC=24<Ent>	;Read Parameter of Select Item Command
+STGC: 24,6,0,0	;Display Item
+STGC: 1,"check balance"	
+STGC: 2,"transfer"	
+STGC: 3,"refill"	
+STGC: 4,"change password"	
+STGC: 5,"register"	
+STGC: 6,"about"	
OK	
AT+STCR=24,0,1<Ent>	;Request Response of Select Item

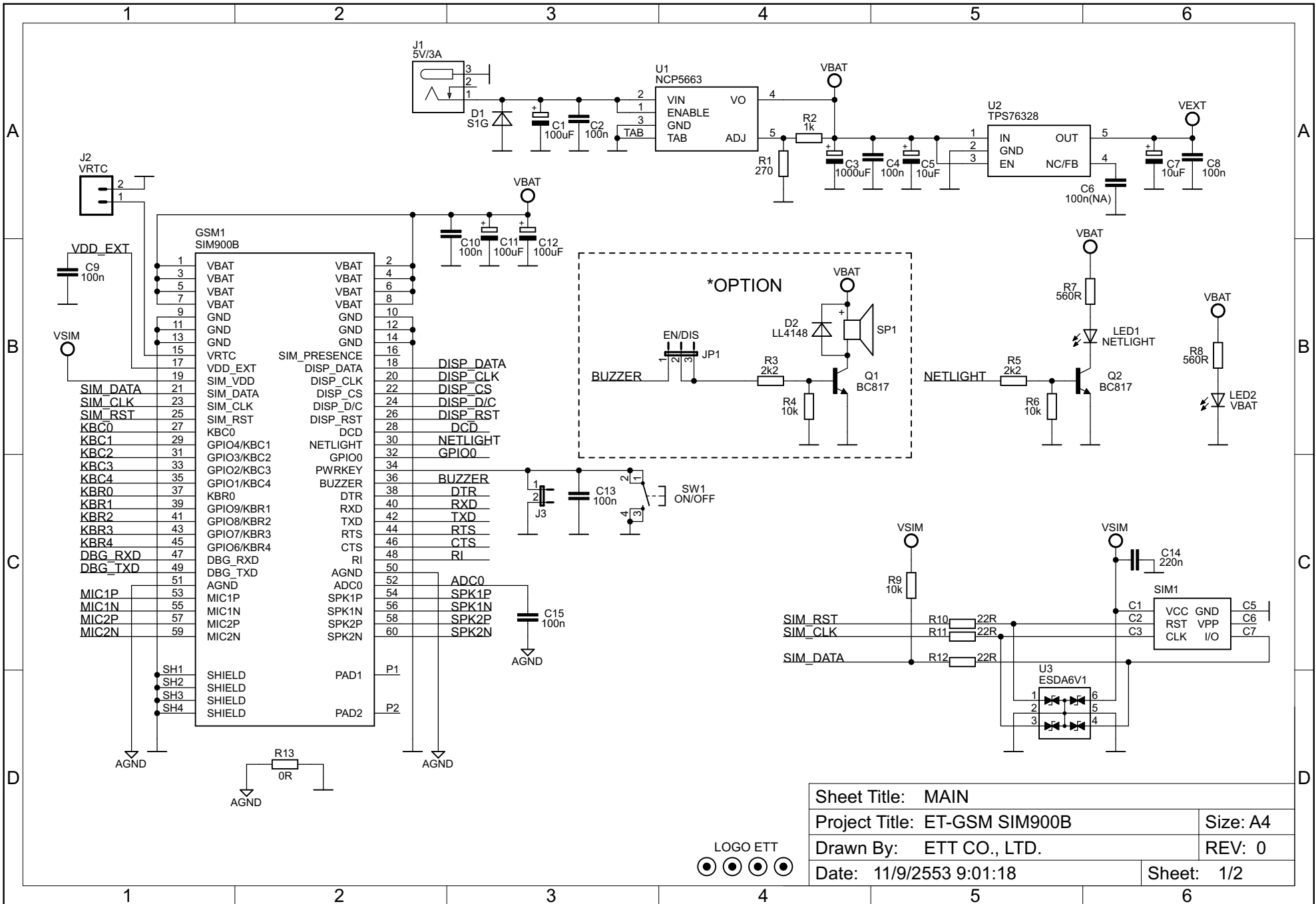
```

OK
+STC: 23 ;Wait Get Input Command
AT+STGC=23<Ent> ;Read Parameter of Get Input Command
+STGC: 23,4,"password:",1,0,0,1,4 ;Display Input Choice
OK
AT+STCR=23,0,4,"1234"<Ent> ;Enter Password = 1234
OK
+STUD: 13 ;Send SMS unsolicited proactive command
+STC: 81 ;End of proactive session
+CMTI: "SM",5 ;SMS Receive = Number 2
AT+CMGR=5<Ent> ;Read SMS Number 5
+CMGR: "REC UNREAD","True Money",,"07/11/22,15:29:45+07"
True Money A/C 0891465362 is 300.00 Baht.
OK

```

แสดง ตัวอย่างการเข้าถึง Application Menu ของ SIM ระบบ True

หมายเหตุ	ข้อความในกรอบสีแดง หมายถึง คำสั่งที่ส่งให้โมดูล SIM900B
	ข้อความในกรอบสีเหลือง หมายถึง ข้อความที่ตอบกลับมาจากโมดูล SIM900B



Sheet Title: MAIN		
Project Title: ET-GSM SIM900B		Size: A4
Drawn By: ETT CO., LTD.		REV: 0
Date: 11/9/2553 9:01:18		Sheet: 1/2



