

ไมโครคอนโทรเลอร์ 8 บิต ตัวใหม่จาก STMicroelectronics

พงษ์ศักดิ์ พร้อมวงศ์, เกียรตินิช นกกรุฑ



มีการพูดถึง MCU 8 bit ตัวใหม่ของทางบริษัท ST มาตั้งแต่ช่วงปลายปีที่แล้วว่าจะมาพร้อมทั้งคุณสมบัติอันมากมาย ด้วยราคาที่ย่อมเยา (ราคาตั้งแต่ 1 ดอลลาร์ถึงครึ่งดอลลาร์สหรัฐที่ MOQ 1kpcs.) จนกระทั่งเมื่อเร็ว ๆ นี้ ถึงได้มีการเปิดตัว MCU ตระกูลใหม่นี้ภายใต้ชื่อ STM8, ความแตกต่างของ MCU จากผู้ผลิตรายนี้ ที่แตกต่างจาก MCU 8 bit ตระกูลอื่นๆที่มีขายอยู่ในท้องตลาดที่ผ่านมาคือ

- 1) จะทำการตลาดเฉพาะโรงงานขนาดใหญ่ ทำให้ไม่แพร่หลายในตลาดสำหรับรายย่อย
- 2) Tools ในการพัฒนา (Programmer/Debugger) และ Evaluation Board ไม่สามารถหาซื้อได้ง่าย
- 3) ไม่มีตำราภาษาไทยไว้ศึกษา และไม่มี web board ไว้ปรึกษาผู้รู้ เมื่อมีข้อสงสัย

แต่ดูเหมือนสถานการณ์เหล่านี้จะเริ่มคลี่คลายขึ้น ในช่วงปีที่ผ่านมา เรารู้จัก MCU จาก ST ในชื่อของ STM32 ซึ่งเป็น ARM 32bit- Cortex-M3 ตัวแรก เริ่มมีผู้ผลิต Evaluation Board ในประเทศ ได้ทำบอร์ดพัฒนาออกมา ทั้ง ETT และ INEX รวมทั้งสามารถซื้อชุด Programmer/Debugger ที่ชื่อ R-Link หรือซื้อปลีกเฉพาะตัว MCU ได้จากบริษัททั้งสอง

แต่ตัวที่ผมตั้งใจจะพูดถึงในวันนี้คือ MCU 8 bit ตัวใหม่ ที่ได้ข่าวว่าจะมาพร้อม ชุดพัฒนาทั้ง Evaluation board, Programmer/Debugger ตัวใหม่ที่ชื่อ ST-Link ประมาณปลายเดือนกรกฎาคมนี้ที่ ETT, ความโดดเด่นของ MCU ตัวนี้เมื่อเทียบกับ MCU 8 bit ในระดับราคาเดียวกัน แทบจะไม่มีคู่แข่ง เพราะมีฟีเจอร์พื้นฐานครบทุกอย่าง ทั้ง Flash Memory (เขียนได้ 10k ครั้ง), RAM ขนาดใหญ่ (1k-6kByte), EEPROM ขนาด 1k-2kByte (เขียนได้ 300k ครั้ง), การสื่อสารแบบอนุกรมทั้ง UART, I2C, SPI, ซึ่งการสื่อสารอนุกรมทุกตัวมี Timer ภายในของตัวเอง ทำให้ไม่ต้องไปใช้ Timer ภายนอกประสงค์, มี Timer ภายนอกประสงค์ ขนาด 16 bit ตั้งแต่ 2-3 ตัว แล้วแต่เบอร์ และมี Timer ภายนอกประสงค์ ขนาด 8 bit พร้อมพรีสเกลเลอร์ 8 bit อีก 1 ตัว, มี Advance Timer1 ซึ่งสามารถขับมอเตอร์ 3 เฟสได้ (ACIM, BLDC), มี High Speed Internal OSC ความละเอียดสูง +/- 2.5% จากโรงงานที่ความถี่สูงสุด 16MHz, ในรุ่น Access Line ต่อกับ External OSC ได้สูงสุด 16MHz จะได้ความเร็ว 12.8MIPs และในรุ่น Performance Line ต่อกับ External OSC ได้สูงสุด 24MHz จะได้ความเร็ว 20MIPs, มี ADC ขนาด 10 bit ที่ความเร็ว 300ksample/sec สามารถมัลติเพล็กซ์ได้ตั้งแต่ 4-16ch, มี AWU: Auto Wake up Timer ไว้ทำการปลุกในโหมดประหยัดพลังงาน รวมทั้งมี Hardware Beeper (ความถี่ 1k, 2k, 4kHz) ในตัว

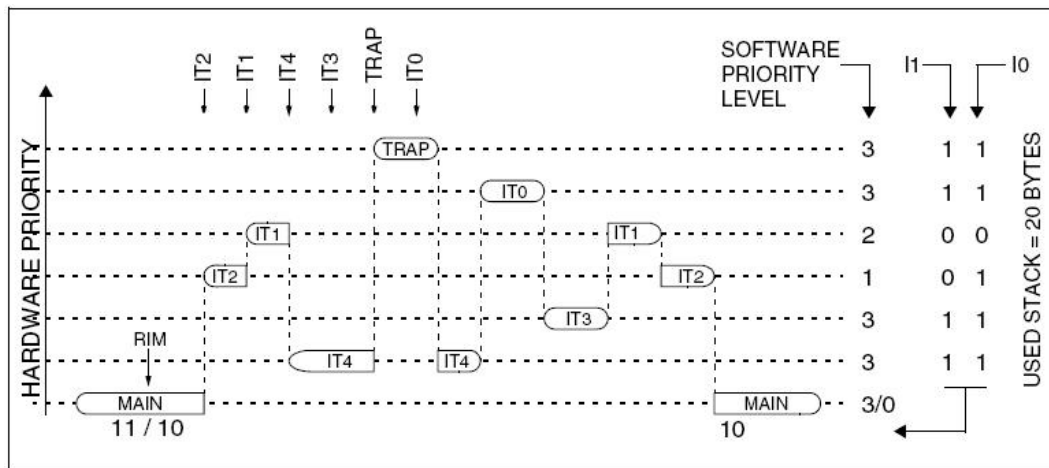
ที่ผมชอบที่สุดคือมีการจัดวาง Interrupt ที่ดีที่สุดในกลุ่ม MCU ขนาด 8 bit, โดยทั่วไป MCU 8 bit ราคาถูกทั่วไป ไม่นิยามให้มีการ อินเทอร์รัพท์ ซ่อน อินเทอร์รัพท์, ถ้ามีอินเทอร์รัพท์ ตัวที่สองเกิดขึ้น ในขณะที่กำลังทำงานใน ISR: Interrupt Service Routine ของอินเทอร์รัพท์ ตัวแรกอยู่ อินเทอร์รัพท์ แพลกของตัวที่สองนั้นจะถูก pending หรือค้างไว้ก่อน จนกว่าจะทำโปรแกรม ISR ของอินเทอร์รัพท์ตัวแรกเสร็จ แต่ของ STM8 สามารถทำอินเทอร์รัพท์ ซ่อน อินเทอร์รัพท์ ได้ ถ้ามีการอนุญาต ตามรูปที่ 1. (STM8 จะมี Interrupt Vector ซึ่งเป็น Hardware Interrupt เหมือนใน MCS-51 แต่มีการใช้ NVIC: Nested Vector Interrupt Control ซึ่งเป็น

จัดกลุ่ม Hardware Interrupt ให้มาอยู่ในกลุ่ม Software Interrupt เดียวกันได้ถ้าต้องการ) รายละเอียดสามารถดาวน์โหลด

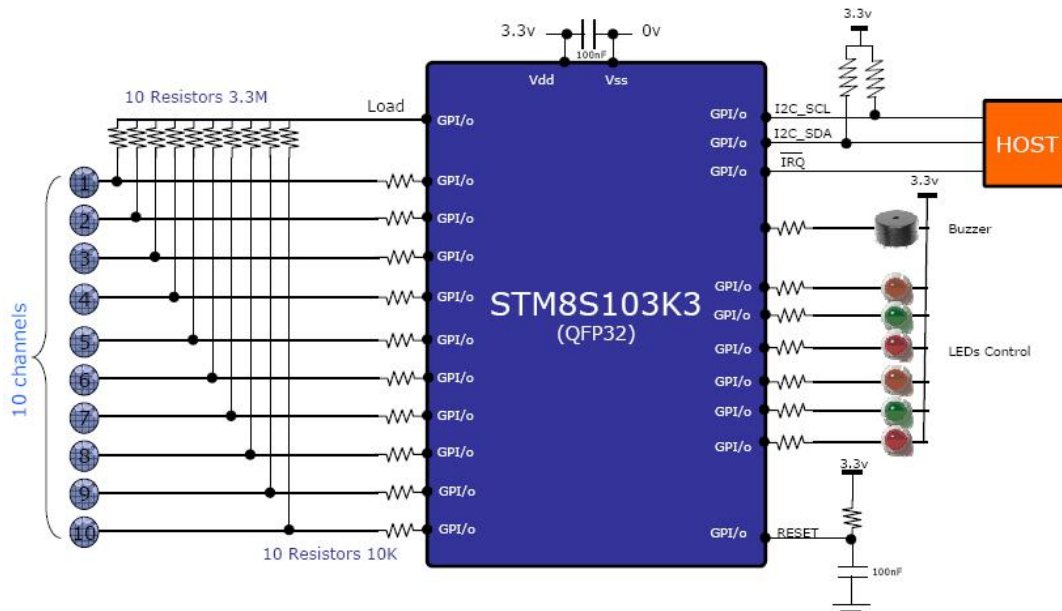
Reference Manual และ Datasheet ได้จากเว็บไซต์ www.st.com

ที่ผมชอบรองลงมาคือใน Firmware Library ของ STM8 มีตัวอย่างการใช้งาน Peripherals แต่ละอย่างค่อนข้างละเอียด อย่างเช่น ในการใช้งาน UART สามารถรับส่งได้ทั้ง Asynchronous และ Synchronous Mode สามารถเชื่อมต่อกับ IrDA มีตัวอย่างทั้งรับและส่ง สามารถติดต่อกับ Smart Card ใน Asynchronous Mode มี LIN รวมทั้งมีโหมด Half duplex แบบ single wire communication (ใช้ขา Tx เพียงขาเดียวรับส่งแบบ Half Duplex) ถ้ายังไม่พอใจ ใน Performance Line ยังแถม UART ตัวที่สองให้ใช้พร้อม CAN bus

และที่น่าสนใจยิ่งขึ้น คือมี Firmware Library ของ Capacitive Touch-Sensing ให้เลือกใช้ได้ฟรีด้วย (ใช้ TIM3, TIM4 และ Program size ขนาด 2kByte ในการทำ Touch-Sensing) โดยไม่จำเป็นต้องต่ออุปกรณ์ภายนอกเลย ตามรูปที่ 2.



รูปที่ 1. แสดงการเกิดอินเทอร์รัพท์ ซ่อน อินเทอร์รัพท์



รูปที่ 2. แสดงตัวอย่างการใช้ STM8S103K3 ทำหน้าที่เป็น Touch-Sensing Slave โดยสแกน Touch pad แล้วแสดงผลออก LED, ส่งเสียงออก Beeper แล้วส่งข้อมูลให้ MCU (Host) ตัวอื่นผ่าน I2C

ในที่นี้ผมขอเลือกยกตัวอย่าง Features ของเบอร์ STM8S105 มาให้ชมกัน

Core

- 16 MHz advanced STM8 core with Harvard architecture and 3-stage pipeline
- Extended instruction set

Memories

- Medium-density Flash/EEPROM:

- Program memory up to 32 Kbytes; data retention 20 years at 55°C after 10 kcycles
- Data memory up to 1 Kbytes true data EEPROM; endurance 300 kcycles
- RAM: up to 2 Kbytes

Clock, reset and supply management

- 3.0 to 5.5 V operating voltage
- Flexible clock control, 4 master clock sources:
 - Low power crystal resonator oscillator
 - External clock input
 - Internal, user-trimmable 16 MHz RC
 - Internal low power 128 kHz RC
- Clock security system with clock monitor
- Power management:
 - Low power modes (Wait, Active-halt, Halt)
 - Switch-off peripheral clocks individually
- Permanently active, low consumption power-on and power-down reset

Interrupt management

- Nested interrupt controller with 32 interrupts
- Up to 37 external interrupts on 6 vectors

Timers

- 2x 16-bit general purpose timers, with 2+3 CAPCOM channels (IC, OC or PWM)
- Advanced control timer: 16-bit, 4 CAPCOM channels, 3 complementary outputs, dead-time insertion and flexible synchronization
- 8-bit basic timer with 8-bit prescaler
- Auto wake-up timer
- 2 watchdog timers: Window watchdog and independent watchdog

Communications interfaces

- UART with clock output for synchronous operation, Smartcard, IrDA, LIN
- SPI interface up to 8 Mbit/s
- I2C interface up to 400 kbit/s

Analog-to-digital converter (ADC)

- 10-bit, ± 1 LSB ADC with up to 10 multiplexed channels, scan mode and analog watchdog I/Os
- Up to 38 I/Os on a 48-pin package including 16 high sink outputs
- Highly robust I/O design, immune against current injection

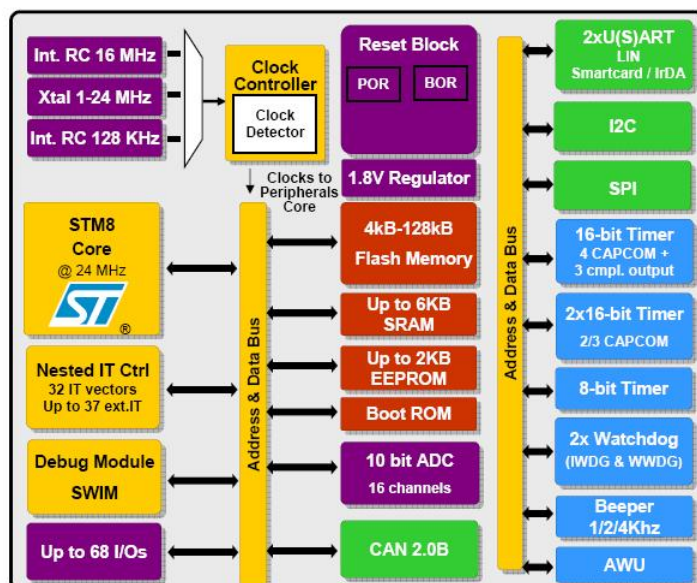
Development support

- Embedded Single Wire Interface Module (SWIM) for fast on-chip programming and non intrusive debugging

STM8S Block Diagram

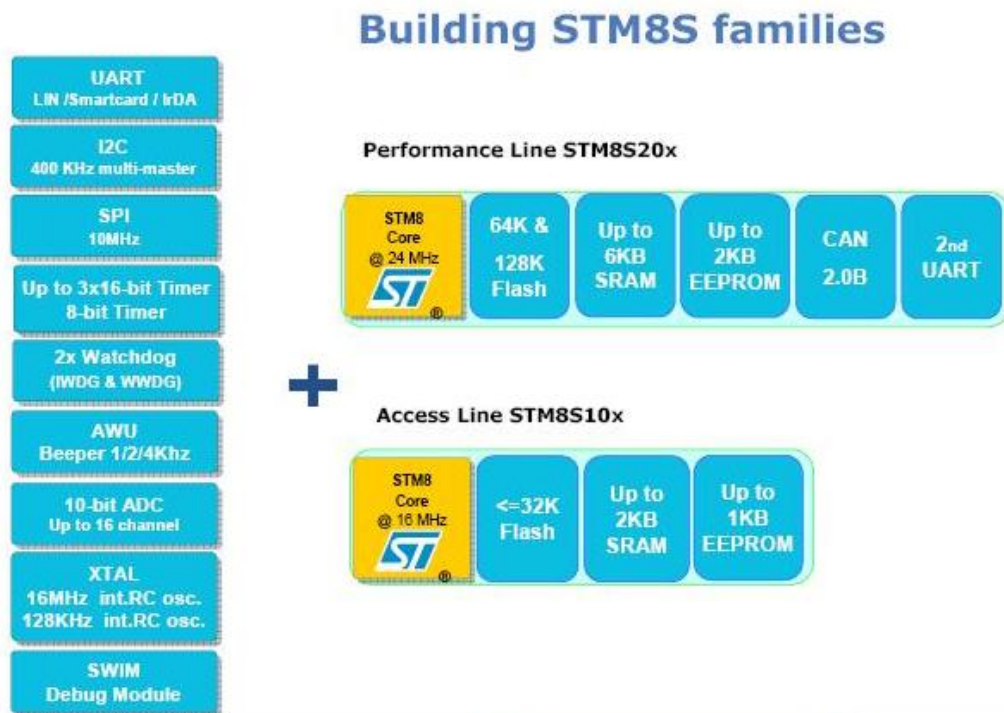
Key features

- 3.0-5.5V
- -40 to +125 °C
- 24MHz core frequency
- 10K cycles for Flash
- 300K cycles for EEprom
- 4 Low power modes (~5µA in Halt mode)
- Trimmable HSI RC 16MHz, +/-1%
- IrDA and Smartcard IF
- SWIM for fast programming (<6s for 128KB)
- LQFP 80, 64, 48, 44, 32
- VQFN 20, 32, 48
- TSSOP 20



รูปที่ 3. บล็อกไดอะแกรมภายในของ STM8S

บล็อกโคอะแกรมภายในของ STM8 แสดงตามรูปที่ 3. จะสังเกตเห็น Voltage regulator ขนาด 1.8V อยู่ภายในสำหรับเลี้ยง MCU core รวมทั้งมี Boot Loader อยู่ด้วย ทำให้สามารถโปรแกรมผ่าน Serial Communication Port ได้ โดยไม่ต้องใช้ Programmer/Debugger



รูปที่ 4. ไมโครคอนโทรลเลอร์ในตระกูลนี้เบื้องต้น สามารถแบ่งออกได้เป็นสองไลน์ คือ Access Line STM8S10x และ Performance Line STM8S20x

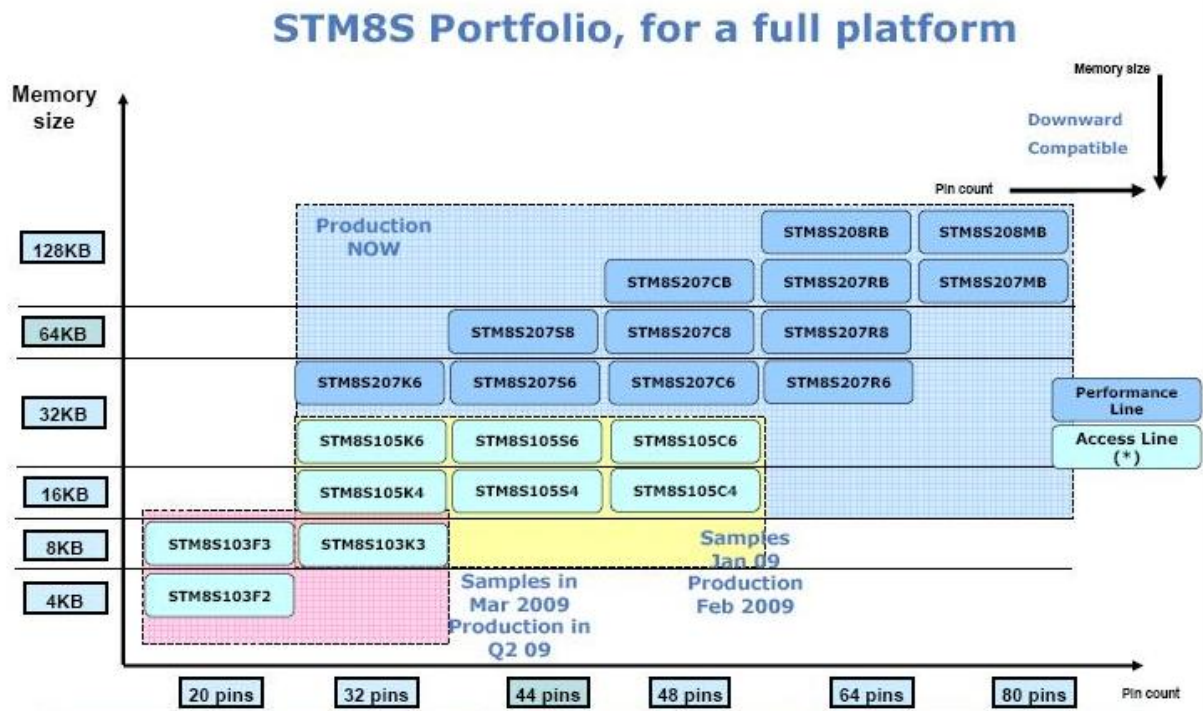
STM8 ในรุ่นแรกนี้ จะแบ่งออกเป็นสองไลน์ตามรูปที่ 4. โดยด้านซ้ายมือคือ Common Features ของทั้งสองไลน์ ส่วนด้านขวามือคือ Features พิเศษของแต่ละไลน์ สิ่งที่มีเหมือนกันคือ ทุกไลน์จะมีอย่างน้อย 1xUART (มี Timer ของตัวเอง), 1xI2C (มี Timer ของตัวเอง), 1xSPI (มี Timer ของตัวเอง), 2 หรือ 3ch 16bit General Purpose Timer(TIM1, TIM2, TIM3), 1ch 8bit Time Base Timer(TIM4), Independent Watch Dog, Window Watch Dog, Auto Wake up Timer, Internal Beeper, 4ch หรือ 7ch หรือ 10ch หรือ 16ch 10bit ADC (300ksample/sec), High Speed Internal (HSI) RC OSC 16MHz ที่ +/- 2.5%, Low Speed Internal (LSI) ที่ความถี่ 128KHz และพอร์ตสำหรับการโปรแกรม/ดีบั๊ก ขนาด 1 เส้น (SWIM: Single Wire Interface Module(ประกอบด้วย Vcc, Data, GND, RST))

ในรุ่น Access Line สามารถใช้ความถี่ High Speed External (HSE) OSC ได้สูงสุดถึง 16 MHz จะได้ความเร็ว 12MIPs, มีหน่วยความจำโปรแกรมตั้งแต่ 4k, 8k, 16k และ 32kByte (เขียนได้ 10kครั้ง), มีหน่วยความจำแรมตั้งแต่ 1k และ 2kByte, มี EEPROM ตั้งแต่ 640 และ 1kByte (เขียนได้ 300kครั้ง)

ส่วนในรุ่น Performance Line มี UART ตัวที่สองเพิ่มเติมเข้ามา รวมทั้งมี CAN 2.0B ให้สามารถใช้ความถี่ High Speed External (HSE) OSC ได้สูงสุดถึง 24 MHz จะได้ความเร็ว 20MIPs

เบอร์ที่น่าสนใจคือ เบอร์ที่มีขนาดหน่วยความจำตั้งแต่ 16kByte ลงมา ตามรูปที่ 5. เพราะ

- 1) Free Compiler ของ Cosmic ใช้ได้ไม่เกิน 16kByte
- 2) เท่าที่ทราบมา (STM8S105K4, STM8S105S4, STM8S105C4) ราคาไม่ถึง 1 USD ที่ 1kpcs. (แต่สามารถซื้อปลีกได้ที่ ETT หรือขอ sample ได้ที่ ST Thailand)



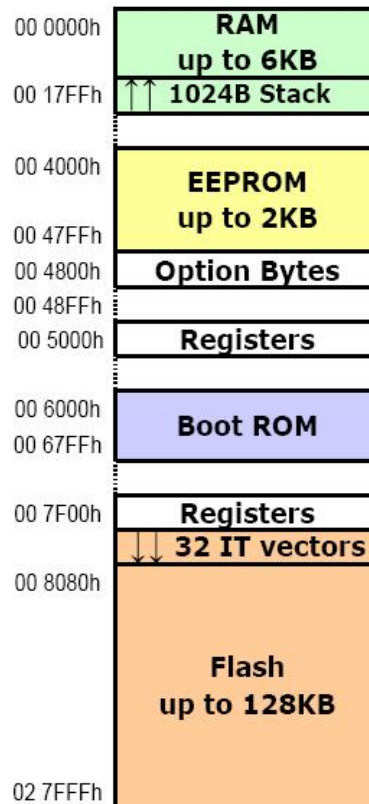
รูปที่ 5. รายละเอียดของเบอร์ต่างในตระกูล STM8

Device	Pin count	No. of maximum GPIO (I/O)	Ext. Interrupt pins	Timer CAPCOM channels	Timer complementary outputs	A/D Converter channels	High sink I/Os	Flash Program memory (bytes)	Data EEPROM (bytes)	RAM (bytes)	Peripheral set	
STM8S105C6	48	38	35	9	3	10	16	Medium density	32K	1024	2K	Advanced control timer (TIM1), General-purpose timers (TIM2 and TIM3), Basic timer (TIM4) SPI, I ² C, UART Window WDG, Independent WDG, ADC
STM8S105C4	48	38	35	9	3	10	16		16K	1024	2K	
STM8S105S6	44	34	31	8	3	9	15		32K	1024	2K	
STM8S105S4	44	34	31	8	3	9	15		16K	1024	2K	
STM8S105K6	32	25	23	8	3	7	12		32K	1024	2K	
STM8S105K4	32	25	23	8	3	7	12		16K	1024	2K	

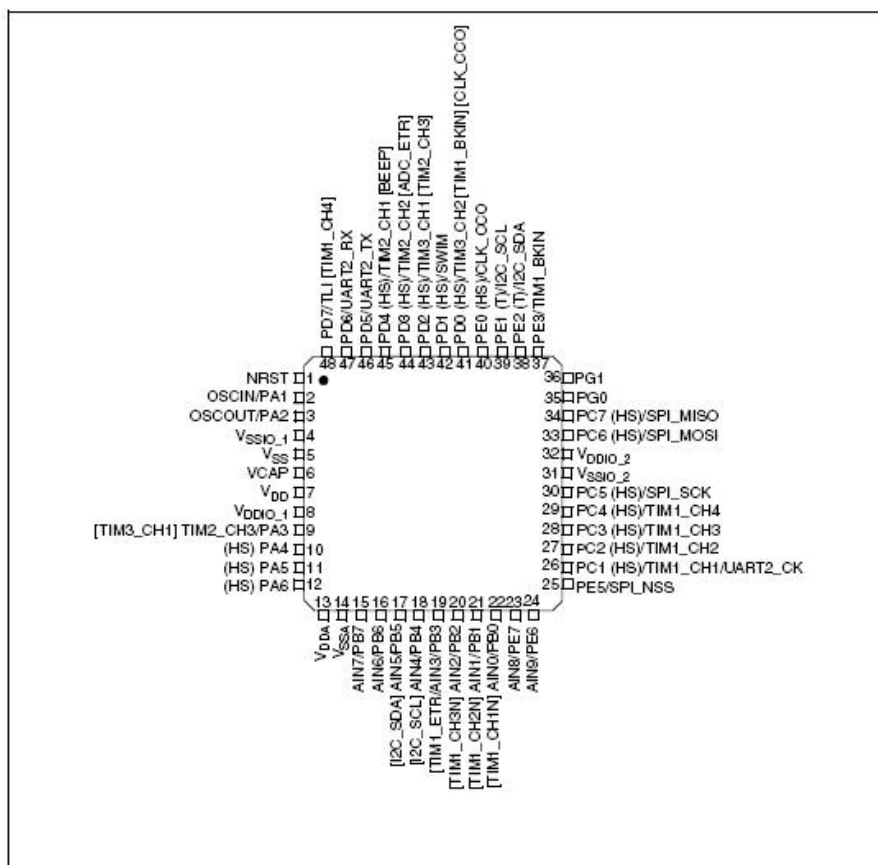
รูปที่ 6. ตัวอย่าง Features ต่างๆของ Access Line

จะขอยกตัวอย่างอนุกรม STM8S105 ตามรูปที่ 6. ซึ่งผมเห็นว่าเป็นอนุกรมที่น่าสนใจที่สุด (รองลงมาคือ อนุกรม STM8S103) แสดงถึง จำนวนขา, จำนวน I/Os, จำนวนขา External Interrupt, จำนวนช่อง ADC, ขนาดโปรแกรม, ขนาด EEPROM และขนาด RAM สามารถเลือกใช้ได้ตามความเหมาะสมของงาน

เนื่องจาก MCU ตระกูลนี้ใช้สถาปัตยกรรมแบบ Harvard ทำให้การมองหน่วยความจำเสมือนเป็นแผ่นเดียวกัน ตามรูปที่ 7. โดยจะเริ่มจากแอดเดรส 0x0000 ซึ่งเป็นส่วนของ RAM และเป็นบริเวณเดียวกับ Stack, แอดเดรส 0x4000 เป็นที่อยู่ของ EEPROM และ Option Byte, แอดเดรส 0x5000 เป็นที่อยู่ของ SFR: Special Function Register, แอดเดรส 0x6000 เป็นที่เก็บ Boot ROM Monitoring, แอดเดรส 0x8000 เป็นที่เก็บ Interrupt Vector และ Program Memory

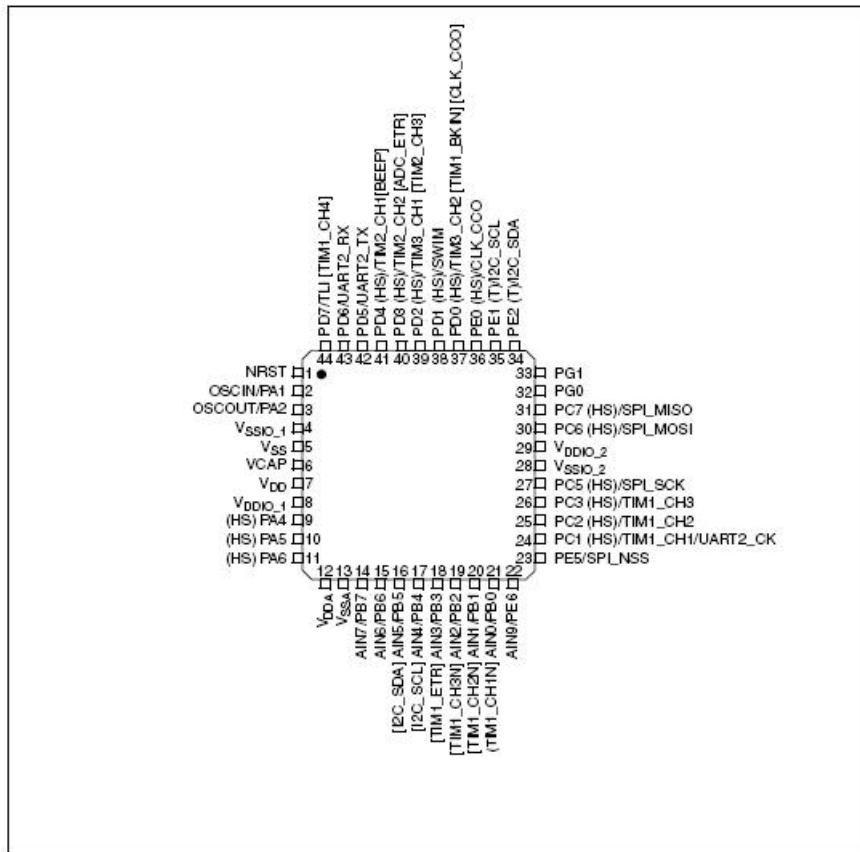


รูปที่ 7. ลักษณะการจัดวางหน่วยความจำ จะมีลักษณะเรียงเป็นเส้นเดียวกัน และมี Boot Loader อยู่ในตัวแล้วด้วย



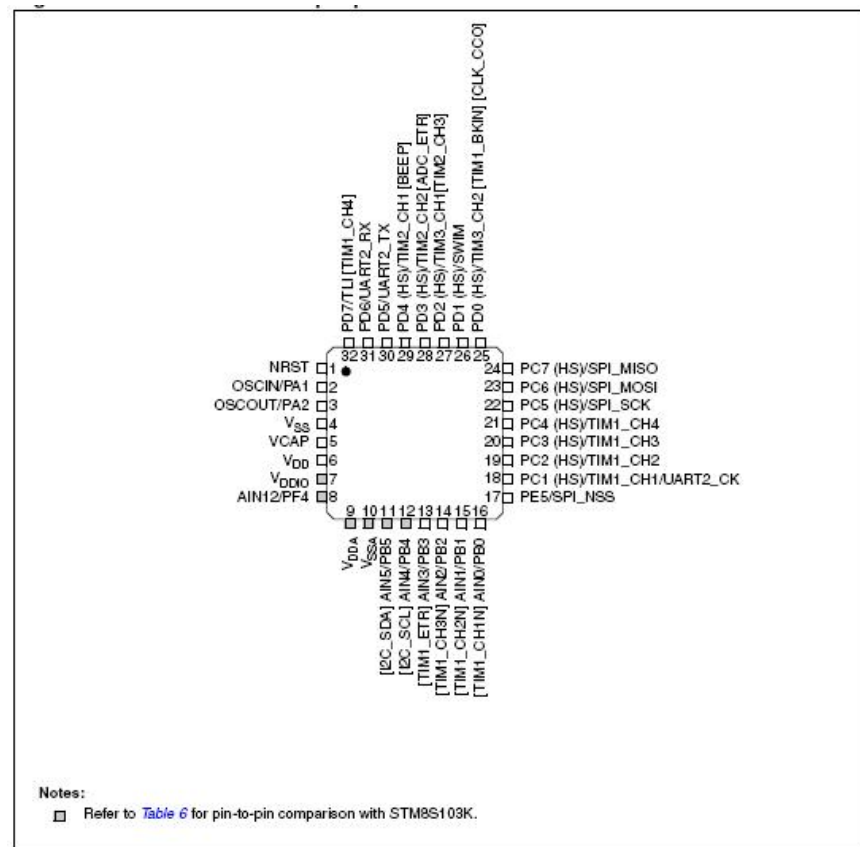
1. (HS) high sink capability.
2. (T) True open drain (P-buffer and protection diode to VDD not implemented).
3. [] alternate function remapping option (If the same alternate function is shown twice, it indicates an exclusive choice not a duplication of the function).

รูปที่ 8. การจัดเรียงขาของ STM8S105C4 (48 Pins, 38 I/Os)



1. (HS) high sink capability.
2. (T) True open drain (P-buffer and protection diode to V_{DD} not implemented).
3. [] alternate function remapping option (If the same alternate function is shown twice, it indicates an exclusive choice not a duplication of the function).

รูปที่ 9. การจัดเรียงขาของ STM8S105S4 (44 Pins, 34 I/Os)

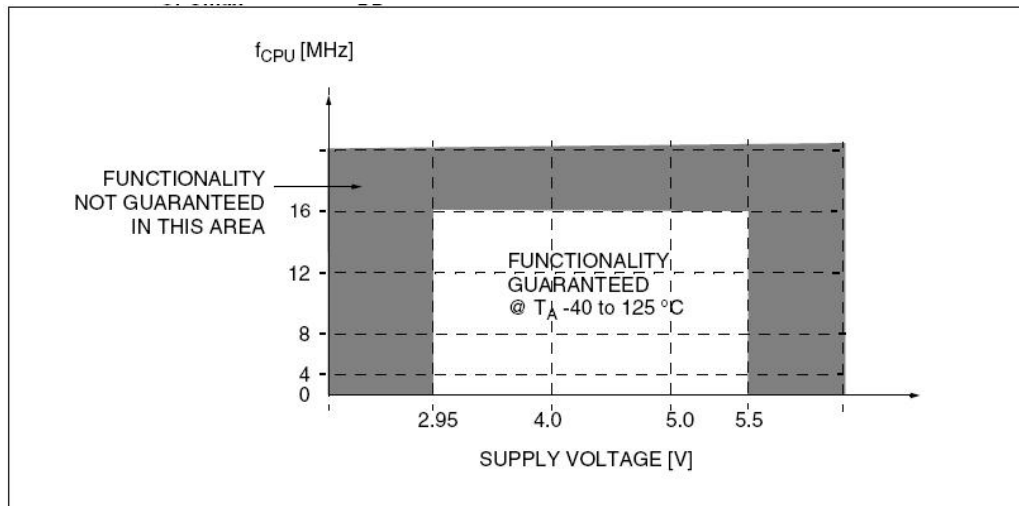


1. (HS) high sink capability.
2. [] alternate function remapping option (If the same alternate function is shown twice, it indicates an exclusive choice not a duplication of the function).

รูปที่ 10. การจัดเรียงขาของ STM8S105K4 (32 Pins, 25 I/Os)

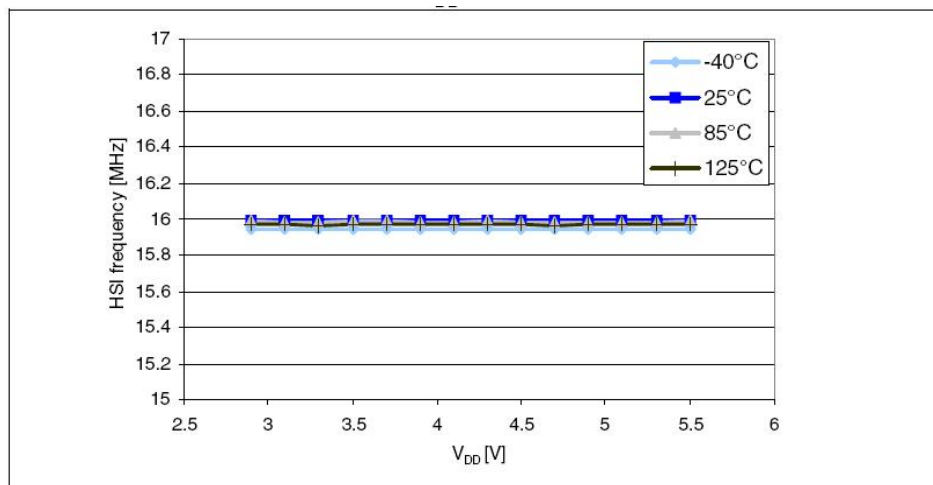
ขาของ MCU ในอนุกรม STM8S105 ตั้งแต่ 48 pin, 44 pin และ 32 pin แสดงตามรูปที่ 8. ถึง 10.

สังเกตว่า บางขาที่อยู่ในเครื่องหมาย [] จะเป็นขาที่สามารถ remap ได้ ด้วยการเซตค่าใน Option Byte, ส่วนคำว่า (HS) หมายถึง High Sink สามารถรับกระแสได้สูงถึง 20mA, โดย V_{DDIO} หมายถึงไฟเลี้ยงที่จ่ายให้ขา I/Os แยกต่างหากจากขา V_{CC} (เพื่อให้ I/Os สามารถจ่ายกระแสได้มากขึ้น และไม่ไปโหลดที่ขา V_{CC} เพียงอย่างเดียว) เบอร์ที่มีขาเยอะๆ ก็จะมี V_{DDIO} หลายๆ ชุด เพื่อแบ่งกันจ่ายกระแส, ขา V_{CAP} มีไว้ต่อกับ Capacitor ขนาด 0.47uF-1.0uF ภายนอก เพราะภายในของ STM8 มี Voltage Regulator ขนาด 1.8V อยู่ข้างใน เพื่อเป็นไฟเลี้ยง Core MCU โดยขา V_{CAP} คือขาเอาต์พุตของ 1.8V Regulator นั้นเอง ส่วนขา V_{DDA} และขา V_{SSA} เป็นแรงดันอ้างอิงสำหรับการแปลง ADC ที่ต้องแยกออกมาต่างหาก เพื่อเป็นการป้องกันการกระเพื่อมของไฟเลี้ยงไปกวน ADC, ขา NRST หมายถึง รีเซตที่แรงดันต่ำ (วงจร Power-on รีเซตสามารถ ต่อ C ขนาด 0.1uF ที่ขา NRST ลงกราวด์)

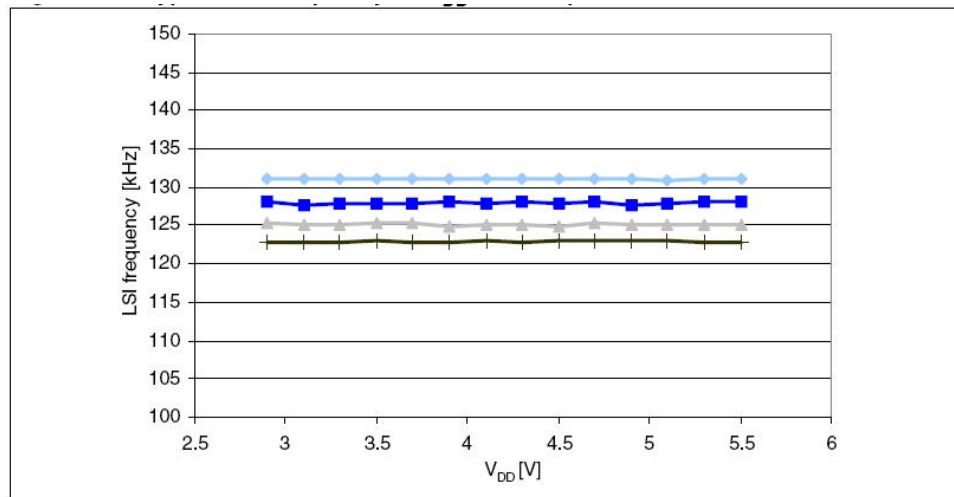


รูปที่ 11. ช่วงการทำงานตั้งแต่แรงดันไฟเลี้ยง 2.95V-5.5V ที่อุณหภูมิ (-40)-(125) องศาเซลเซียส

สำหรับบางงานที่ต้องการใช้ High Speed Internal OSC และต้องการรันที่แรงดัน 3.3V ก็สามารถทำงานได้ดังรูปที่ 11. โดยมีความถี่เบี่ยงเบนเพียงเล็กน้อย ตามรูปที่ 12. ส่วนรูปที่ 13. แสดงความถี่ที่เปลี่ยนไปของ Low Speed Internal RC OSC ซึ่งจะต่ออยู่กับวงจร Watch Dog, Auto Wake Up Timer และ Beeper



รูปที่ 12. แสดงความถี่ของ High Speed Internal (HSI) ตั้งแต่ช่วงแรงดันไฟเลี้ยง 2.95V-5.5V จะเห็นได้ว่ามีค่าค่อนข้างคงที่



รูปที่ 13. แสดงความถี่ของ Low Speed Internal (LSI) ตั้งแต่ช่วงแรงดันไฟเลี้ยง 2.95V-5.5V

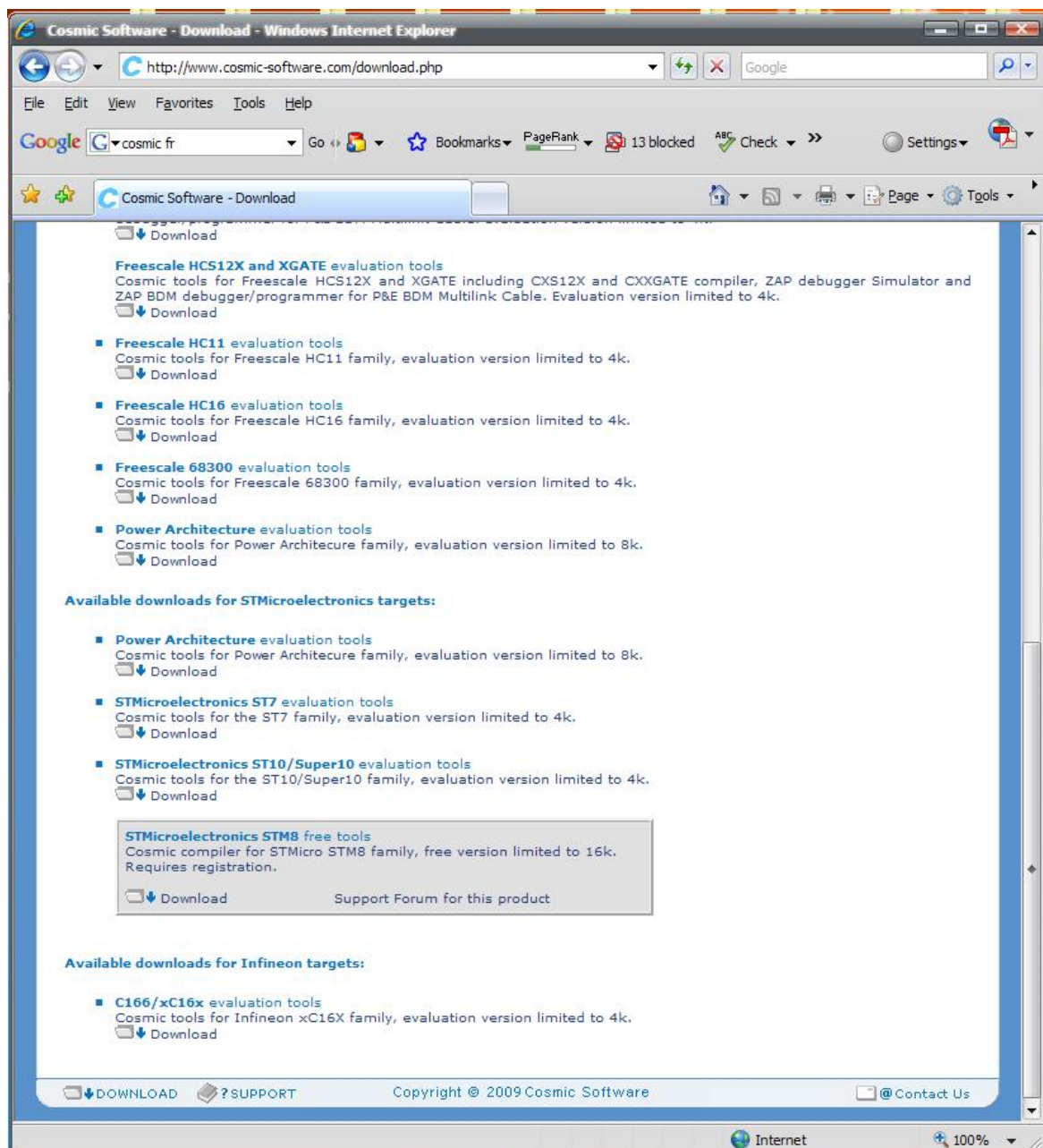


รูปที่ 14. เว็บไซต์ของบริษัท COSMIC Software ให้เลือกที่แท็บ DOWNLOAD

Download Program

สำหรับผู้สนใจจะลองใช้งาน ให้เริ่มจากการดาวน์โหลดโปรแกรม Compiler จากบริษัท COSMIC Software ที่ www.cosmic-software.com ตามรูปที่ 14. เลือกรูปธงชาติอเมริกาเพื่อเปลี่ยนเป็นภาษาอังกฤษ (COSMIC Software เป็นบริษัทของฝรั่งเศส) จากนั้นเลือกที่แท็บ DOWNLOAD ด้านขวามือ

จากนั้นให้เลื่อนลงมาจนพบ STMicroelectronics STM8 free tools เลือกที่ DOWNLOAD ตามรูปที่ 15. จะพบหน้าจอสำหรับรีจิสเตอร์ ตามรูปที่ 16. ให้ใส่รายละเอียดต่างๆไป ไม่ต้องกังวลเพราะขั้นตอนนี้ไม่เสียเงิน จากนั้นจะสามารถดาวน์โหลดโปรแกรมสำหรับติดตั้งชื่อ cxstm8_16k ให้ทำการติดตั้ง ตามรูปที่ 17. เสร็จแล้วจะได้ไอคอนตามรูปที่ 18. ให้ดับเบิลคลิกที่ไอคอนเพื่อเข้าสู่โปรแกรม COSMIC compiler editor ตามรูปที่ 19. แล้วให้เลือกที่เมนู Help > About Idea... จะได้รูปที่ 20. ให้กดที่ปุ่ม License Info. จะปรากฏข้อมูลสำหรับแต่ละเครื่องตามรูปที่ 21. ให้ copy ข้อมูล Product Information ในกรอบล่างที่ถูก Highlight ไว้ นำมาวางบน Notepad ตามรูปที่ 22. ซึ่งข้อมูลตามตัวอย่าง ประกอบด้วย Product information ที่ copy มาจาก Cosmic Compiler, ส่วน User information ให้ระบุข้อมูลของเราลงไป จากนั้นให้ save แล้วส่งไปที่ sales@cosmic.fr ให้รอการตอบกลับซึ่งใช้เวลาประมาณ 2 วัน

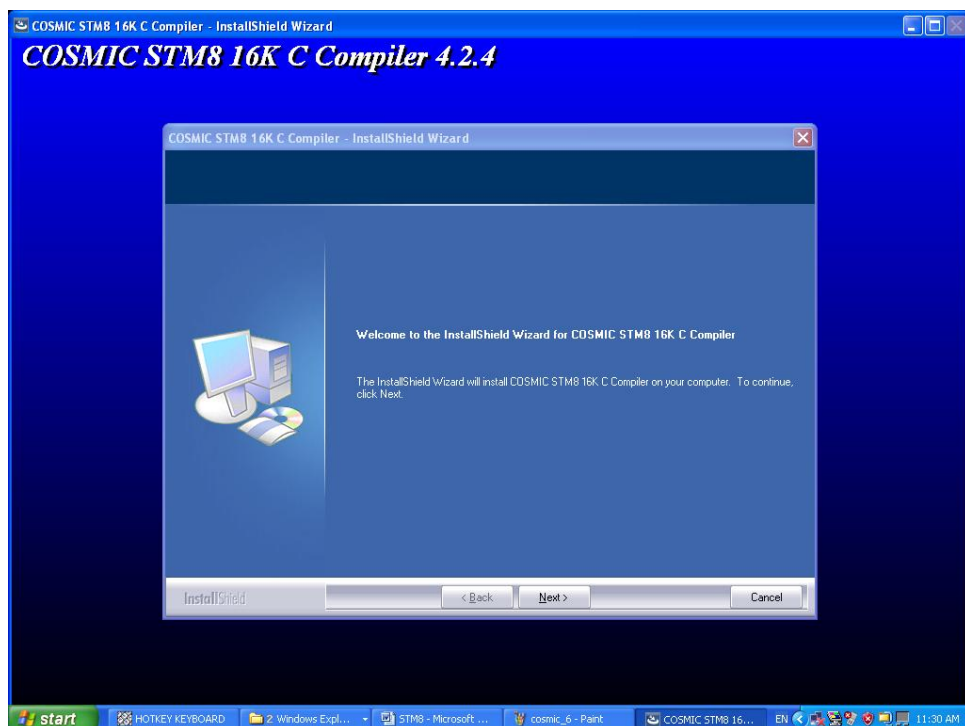


รูปที่ 15. เลื่อนลงมาจนพบ STMicroelectronics STM8 free tools เลือกที่ DOWNLOAD

เมื่อทางบริษัท COSMIC ตอบกลับมา พร้อมไฟล์แนบ license.lic ให้ copy ไปไว้ที่ C:\Program Files\COSMIC\CXSTM8_16k\License\ ตามรูปที่ 23. เพียงเท่านี้เราก็สามารถใช้ COSMIC Compiler ได้ฟรีถึง 16kByte

The screenshot shows a web browser window titled "Cosmic Software - Download - Windows Internet Explorer". The address bar shows "http://www.cosmicsoftware.com/download_stm8_16k.php". The page content includes the COSMIC logo, navigation links (ABOUT US, NEWS & EVENTS, CONTACT US, PRODUCTS & SERVICES, SUPPORT, DOWNLOAD), and a section titled "Download of the FREE stm8 16k version". Below this, there is a registration form with fields for Name, Company, Address, ZIP Code, City, Country (a dropdown menu), Phone, Fax, and E-mail. There are "Submit" and "Clear" buttons at the bottom of the form. The footer of the page includes links for DOWNLOAD, SUPPORT, and CONTACT US, along with a copyright notice for 2008 COSMIC Software.

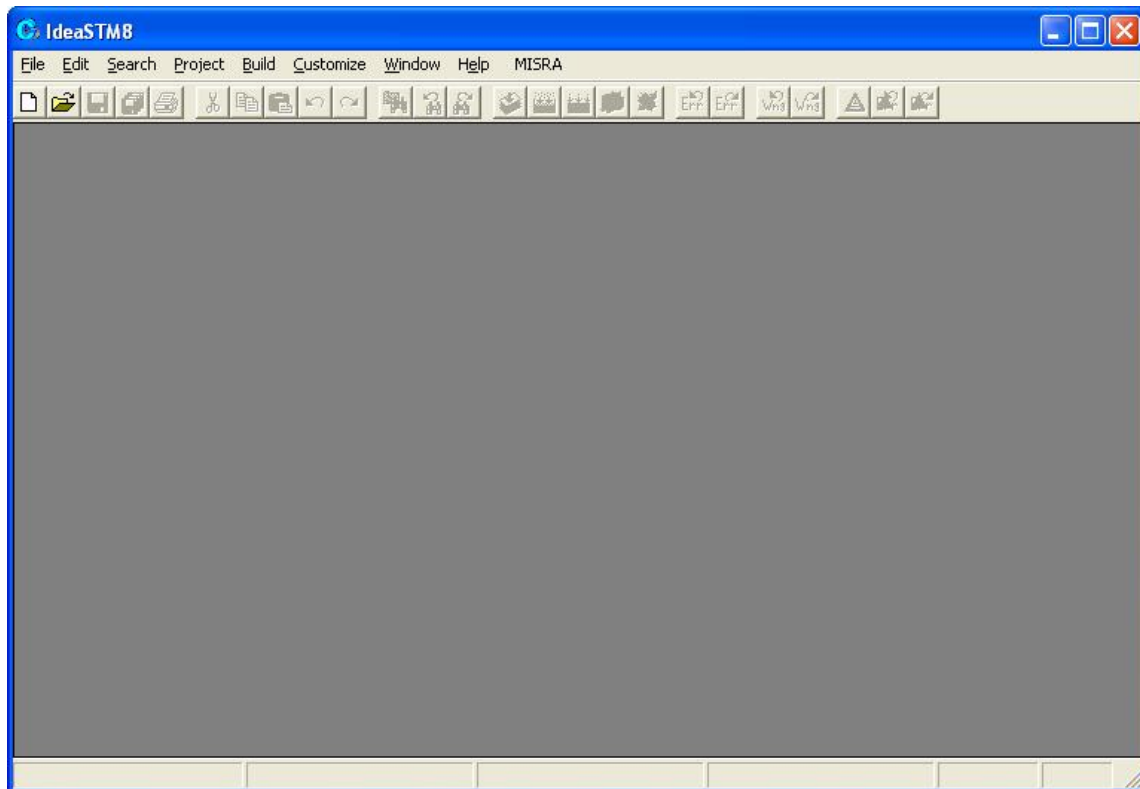
รูปที่ 16. หน้าต่างให้ Register (CITY=Bangkok, Country=Thailand) ที่มีเครื่องหมายดอกจันให้ใส่ข้อมูลลงไป



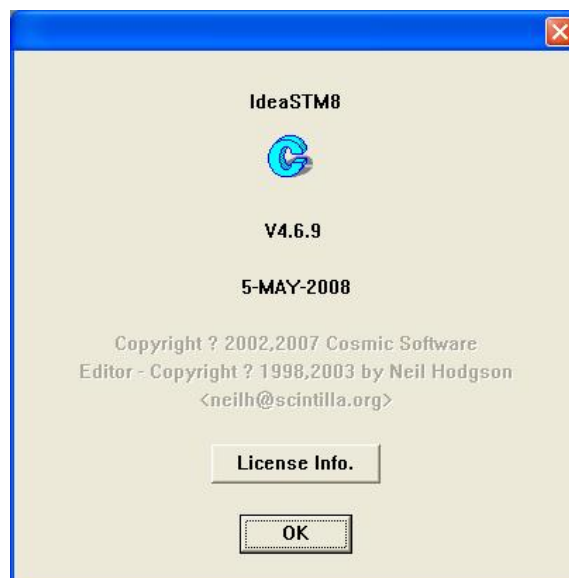
รูปที่ 17. ให้ลงโปรแกรม cxstm8_16k ที่ดาวน์โหลดมาได้



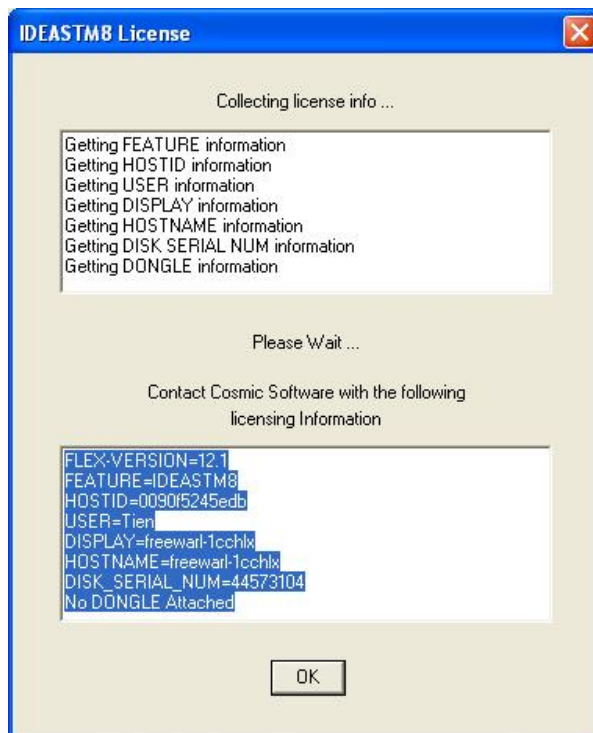
รูปที่ 18. ให้ดับเบิลคลิกที่ไอคอน Cosmic CxSTM8 16K4.2.7



รูปที่ 19. เลือกที่เมนู Help > About Idea...



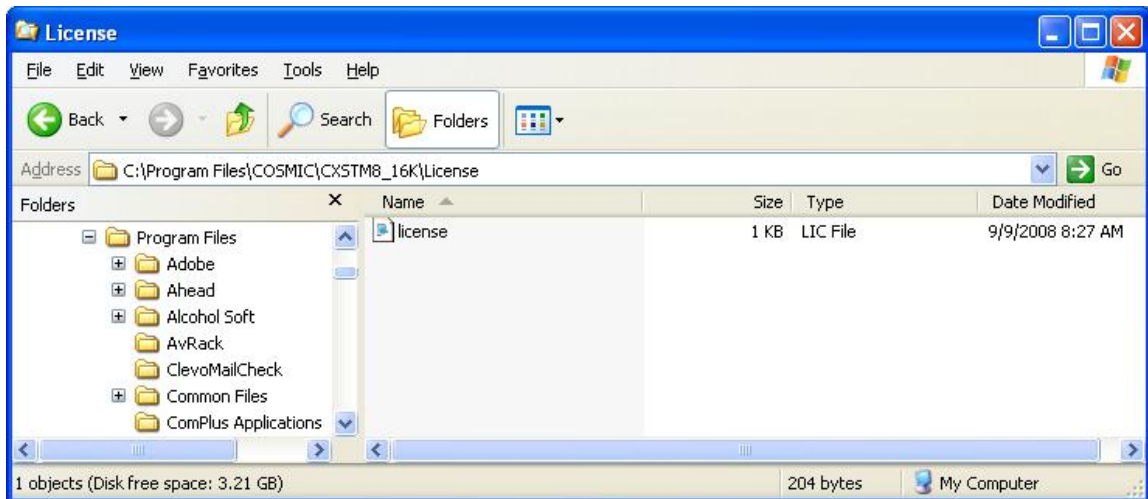
รูปที่ 20. เลือกที่ปุ่ม License Info.



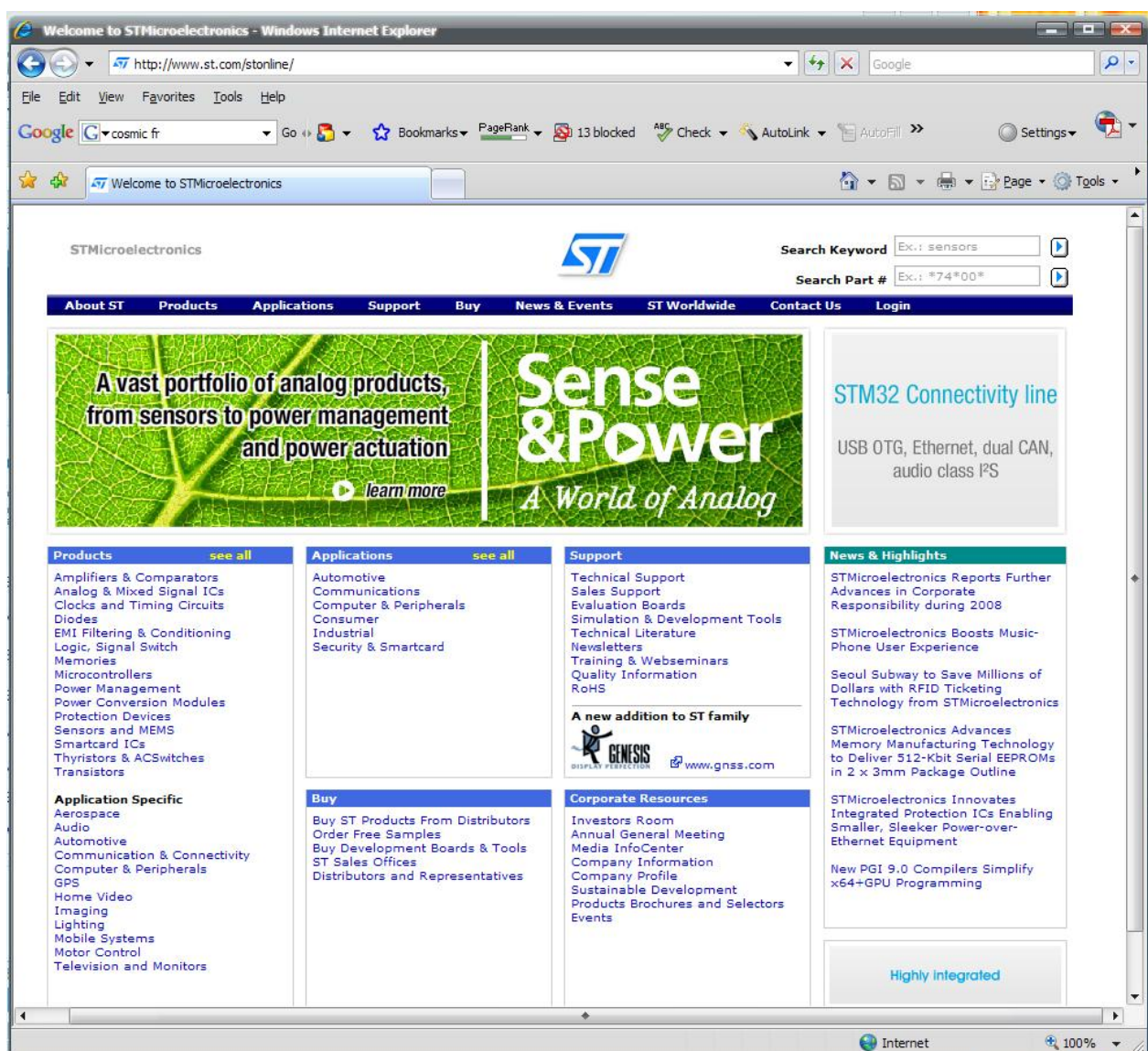
รูปที่ 21. ให้ copy ข้อมูล Product Information ในกรอบล่างที่ถูก Highlight



รูปที่ 22. ใช้โปรแกรม Notepad สร้างข้อมูลตามตัวอย่าง ประกอบด้วย Product information ที่ copy มาจาก Cosmic Compiler, ส่วน User information ให้ระบุข้อมูลของเราลงไป จากนั้นให้ save แล้วส่งไปที่ sales@cosmic.fr ให้รอการตอบกลับซึ่งใช้เวลาประมาณ 2 วัน



รูปที่ 23. เมื่อได้รับไฟล์ license.lic มาแล้ว ให้ copy ไปไว้ที่ C:\Program Files\COSMIC\CXSTM8_16K\License\ เพียงเท่านั้น
เราก็จะสามารถใช้ COSMIC Compiler ได้ฟรีถึง 16kByte



รูปที่ 24. เว็บไซต์ของบริษัท STMicroelectronics ให้เลือกที่แท็บ Products > Microcontroller

แต่เราจะไม่ได้ใช้ Editor ของ COSMIC เราเพียงแต่ต้องการใช้เฉพาะส่วน Compiler เท่านั้น

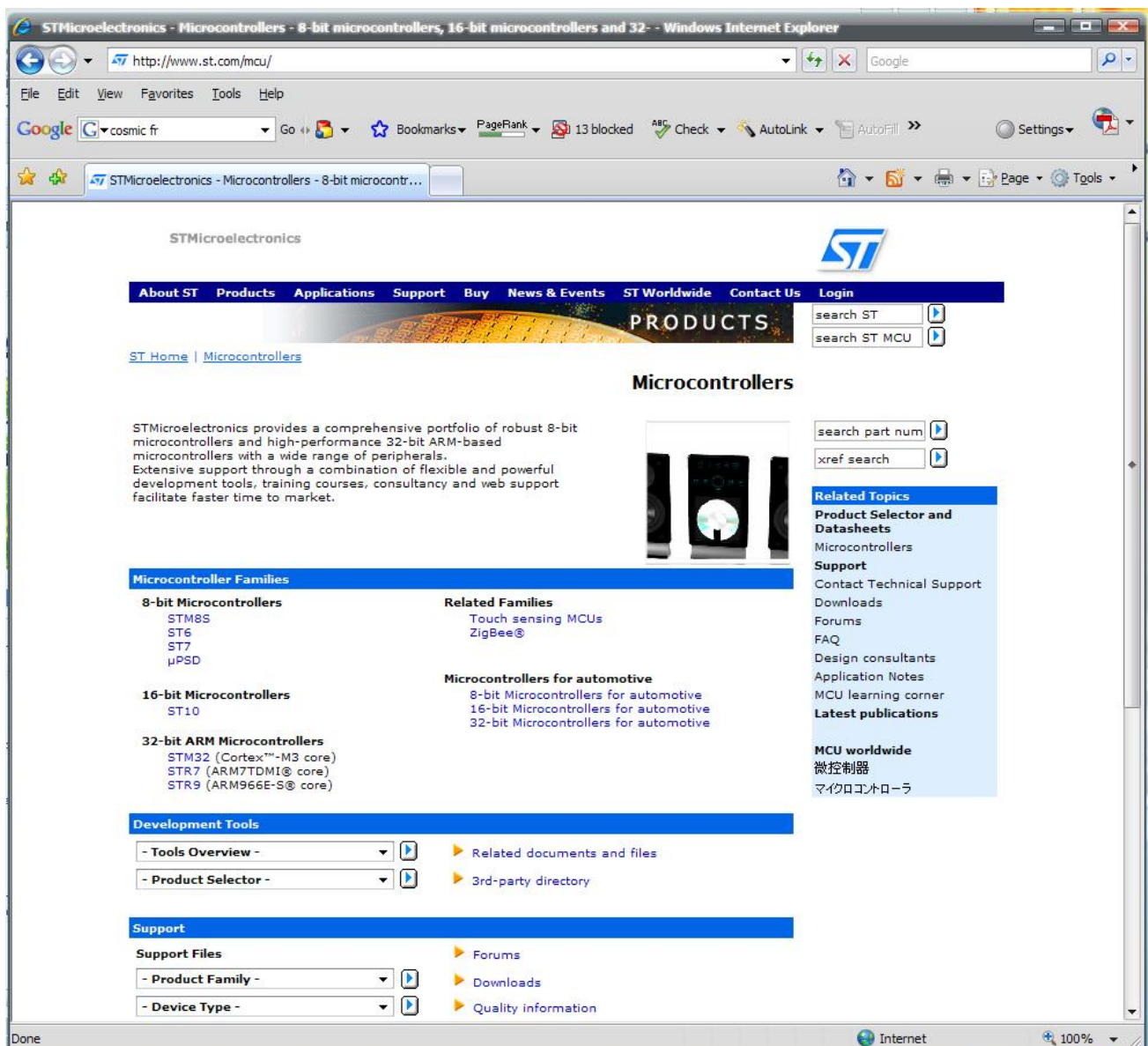
ต่อมาคือการดาวน์โหลดโปรแกรม Editor และ Programmer/Debugger จากทาง ST, ให้เข้าไปที่เว็บไซต์ www.st.com ตามรูปที่ 24. ให้เลือกที่แท็บ Products > Microcontroller จะได้หน้าดังตามรูปที่ 25. ในหัวข้อ 8-bit Microcontrollers ให้เลือกที่ STM8S จะได้รูปที่ 26. ให้เลือกที่ Documents and Files for STM8S family ก็จะได้รูปที่ 27.

ให้ทำการโหลด Datasheet ของ MCU เบอร์ต่างๆ ตามรูปที่ 28.

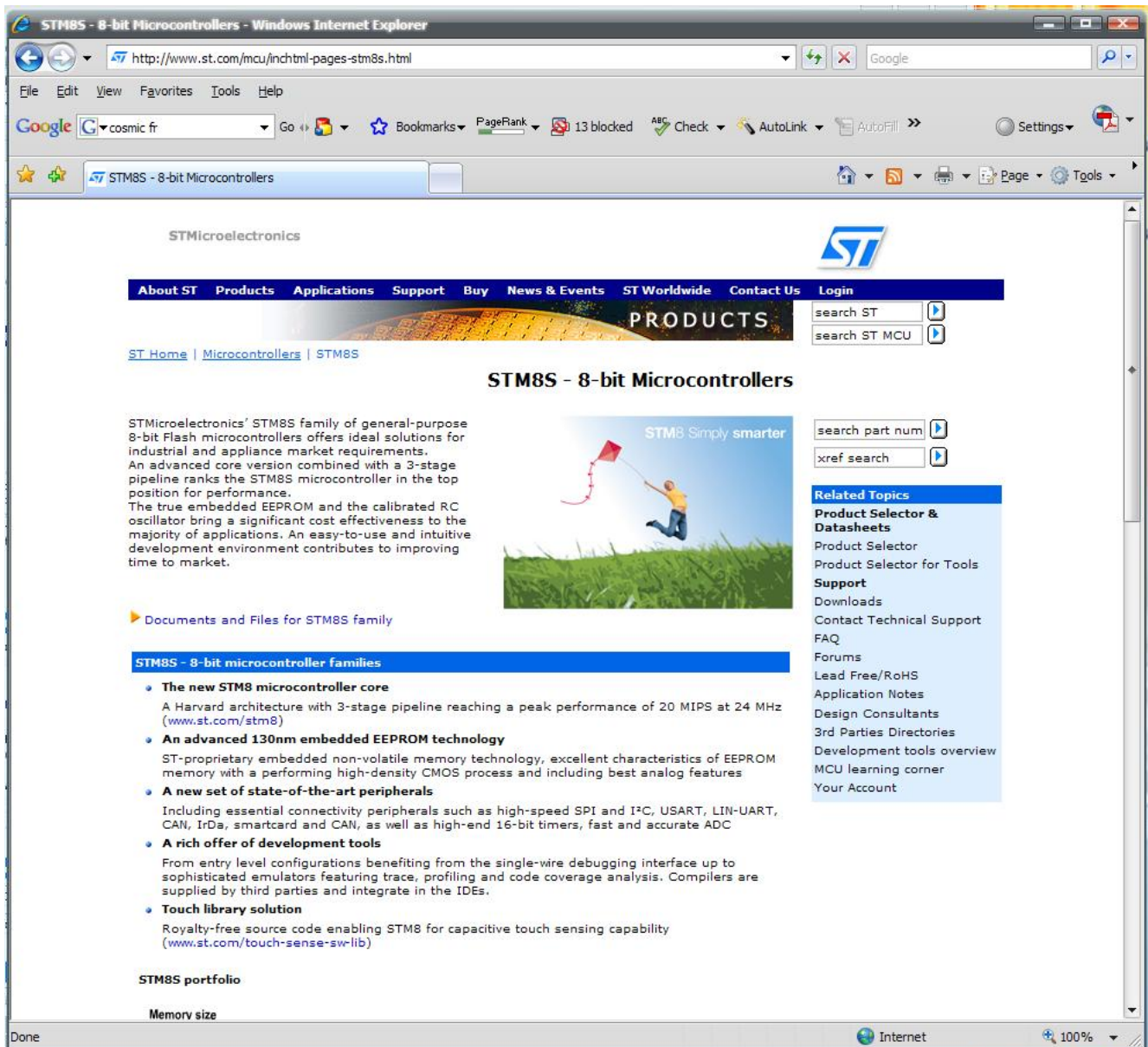
เลื่อนลงมาที่กรอบ Firmware ตามรูปที่ 29. แล้วดาวน์โหลด STM8S FWLib: STM8S firmware library ซึ่งเป็นไลบรารีสำหรับ Peripherals ที่ ST เตรียมไว้ให้ และดาวน์โหลด STM8S FWLib: STM8S Touch Sensing Library ซึ่งเป็นไลบรารีสำหรับความสามารถ Capacitive Touch-Sensing ที่ ST เตรียมไว้ให้

ให้เลื่อนลงมาที่กรอบ Reference Manual ตามรูปที่ 30. แล้วดาวน์โหลด RM0016: STM8S microcontroller family

สุดท้ายให้เลื่อนลงมาที่กรอบ Software for Tools ตามรูปที่ 31. แล้วดาวน์โหลด ST toolset จากนั้นให้เลื่อนลงมาที่กรอบ Software Patches แล้วดาวน์โหลด STVD-4.1.1 Patch



รูปที่ 25. ในหัวข้อ 8-bit Microcontrollers ให้เลือกที่ STM8S



รูปที่ 26. ให้เลือกที่ Documents and Files for STM8S family

STM8 - 8-bit Microcontrollers - Technical Literature and Support Files - Windows Internet Explorer

http://www.st.com/mcu/familiesdocs-113.html

File Edit View Favorites Tools Help

Google cosmic fr Go Bookmarks PageRank 13 blocked Check AutoLink AutoFill Settings

STM8 - 8-bit Microcontrollers - Technical Literature and Support Files

STMicroelectronics

About ST Products Applications Support Buy News & Events ST Worldwide Contact Us Login

PRODUCTS

search ST search ST MCU

ST Home | Microcontrollers

Documents and files for family STM8 - 8-bit Microcontrollers

Documents and files for family STM8 - 8-bit Microcontrollers

Please report broken links !

Available Documents
 Advertizing | Application Note | Brochure / Flyer | Certification | Datasheet | Errata Sheet | Firmware | Presentation - Marketing | Press Release | Programming Manual | Reference Manual | Software for Tools | Technical Note | User Manual | User manual Evaluation board |

Development Tools Section :
 Data Briefing | Reference Schematics | Release Note | Software for Tools | Software Patches | User Manual |

Advertizing

Reference	Description	Version	Date	Size	File	File
	STM8S PRODUCT PRESENTATION USED DURING PRESS CONFERENCES		Mar-2009	1213 KB		

Application Note

Reference	Description	Version	Date	Size	File	File
AN2687	STM8S20xxx LCD software driver	1	Apr-2009			
AN2752	Getting started with the STM8S	3	Apr-2009			
AN2645	Migration and compatibility guidelines for STM8S microcontroller applications	1	Mar-2009			
AN2737	Basic in-application programming example using the STM8 I2C and SPI peripherals	3	Mar-2009			
AN2659	STM8S in-application programming (IAP) using a customized bootloader	1	Feb-2009			
AN2869	Guidelines for designing touch sensing applications	1	Feb-2009			
AN2927	RC acquisition principle for touch sensing applications	2	Feb-2009			
AN2822	STM8S high speed internal oscillator calibration	1	Jan-2009			
AN2867	Oscillator design guide for ST microcontrollers	1	Jan-2009			
AN2860	EMC guidelines for STM8S microcontrollers	2	Jan-2009			

Related Topics

Product Selector and Datasheets

Microcontrollers

Support

Contact Technical Support

Downloads

Forums

FAQ

Design consultants

Application Notes

MCU learning corner

Latest publications

MCU worldwide

微控制器

マイクロコントローラ

Internet 100%

รูปที่ 27. หน้าต่างของ Documents and Files for family STM8 - 8- bit Microcontrollers

STM8 - 8-bit Microcontrollers - Technical Literature and Support Files - Windows Internet Explorer

http://www.st.com/mcu/familiesdocs-113.html

File Edit View Favorites Tools Help

Google cosmic fr Go 13 blocked Check AutoLink AutoFill Settings

STM8 - 8-bit Microcontrollers - Technical Literature an...

Reference	Description	Version	Date	Size	File	File
	VDE certification for STM8S ClassB fw		Apr-2009	1902 KB		

Reference	Description	Version	Date	Size	File	File
STM8S103xx	Access line, 16 MHz STM8S 8-bit MCU, up to 8 Kbytes Flash, data EEPROM, 10-bit ADC, 3 timers, UART, SPI, PC	3	Jun-2009			
STM8S103F3						
STM8S103F2						
STM8S103K3						
STM8S105xx	Access line, 16 MHz, STM8S 8-bit MCU, up to 32 Kbytes Flash, integrated EEPROM, 10-bit ADC, timers, UART, SPI, PC	8	Jun-2009			
STM8S105S6						
STM8S105S4						
STM8S105K6						
STM8S105K4						
STM8S105C6						
STM8S105C4						
STM8S903K3	16 MHz STM8S 8-bit MCU, up to 8 Kbytes Flash, 1 Kbyte RAM, 640 bytes EEPROM, 10-bit ADC, 2 timers, UART, SPI, PC	1	Apr-2009			
STM8S207xx	Performance line, 24 MHz, 8-bit MCU, up to 128 Kbytes Flash, integrated EEPROM, 10-bit ADC, timers, 2 UARTs, SPI, PC, CAN	7	Jan-2009			
STM8S208xx						
STM8S208RB						
STM8S208MB						
STM8S207S8						
STM8S207RB						
STM8S207R8						
STM8S207R6						
STM8S207MB						
STM8S207K6						
STM8S207CB						
STM8S207C8						

Reference	Description	Version	Date	Size	File	File
STM8S207xx	STM8S performance line revision X device limitations	1	Dec-2008			
STM8S208xx						

Reference	Description	Version	Date	Size	File	File
STM8/128-EVAL firmware	STM8/128-EVAL demonstration firmware	1.0.3	Jun-2009			
STM8S FWLib	STM8S firmware library	1.1.1	Jun-2009			
	STM8S Touch Sensing Library	1.1.0	Feb-2009			

Reference	Description	Version	Date	Size	File	File
	Touch sensing software suite presentation	1.0	Mar-2009	1291 KB		

Internet 100%

รูปที่ 28. ให้เลื่อนลงมาที่รอบ Datasheet แล้วดาวน์โหลด STM8S103, STM8S105, STM8S903 และ STM8S208

STM8 - 8-bit Microcontrollers - Technical Literature and Support Files - Windows Internet Explorer

http://www.st.com/mcu/familiesdocs-113.html

File Edit View Favorites Tools Help

Google cosmic fr Go 13 blocked Check AutoLink AutoFill Settings

STM8 - 8-bit Microcontrollers - Technical Literature an...

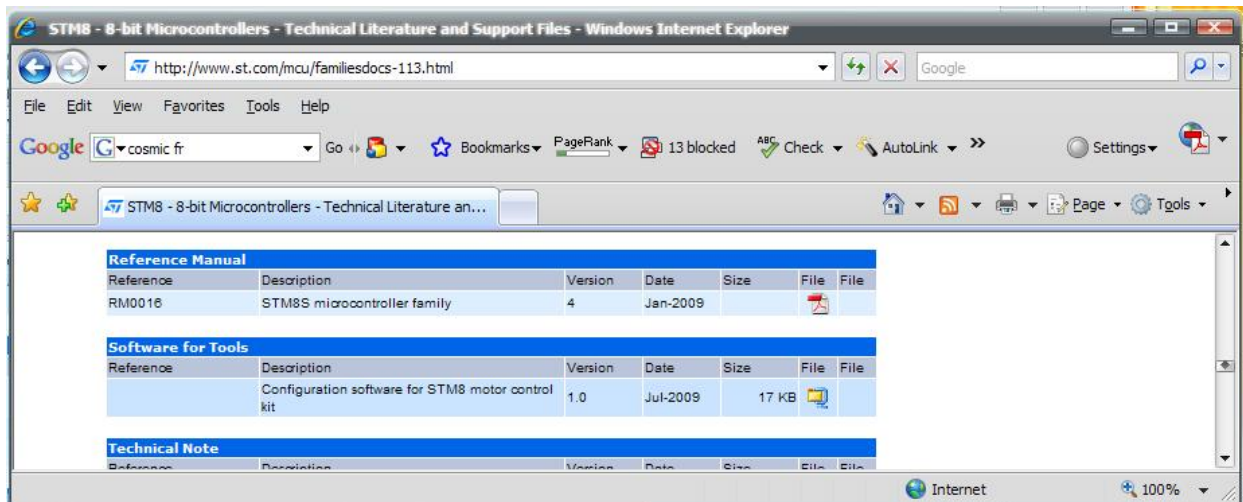
Reference	Description	Version	Date	Size	File	File
STM8S207xx	STM8S performance line revision X device limitations	1	Dec-2008			
STM8S208xx						

Reference	Description	Version	Date	Size	File	File
STM8/128-EVAL firmware	STM8/128-EVAL demonstration firmware	1.0.3	Jun-2009			
STM8S FWLib	STM8S firmware library	1.1.1	Jun-2009			
	STM8S Touch Sensing Library	1.1.0	Feb-2009			

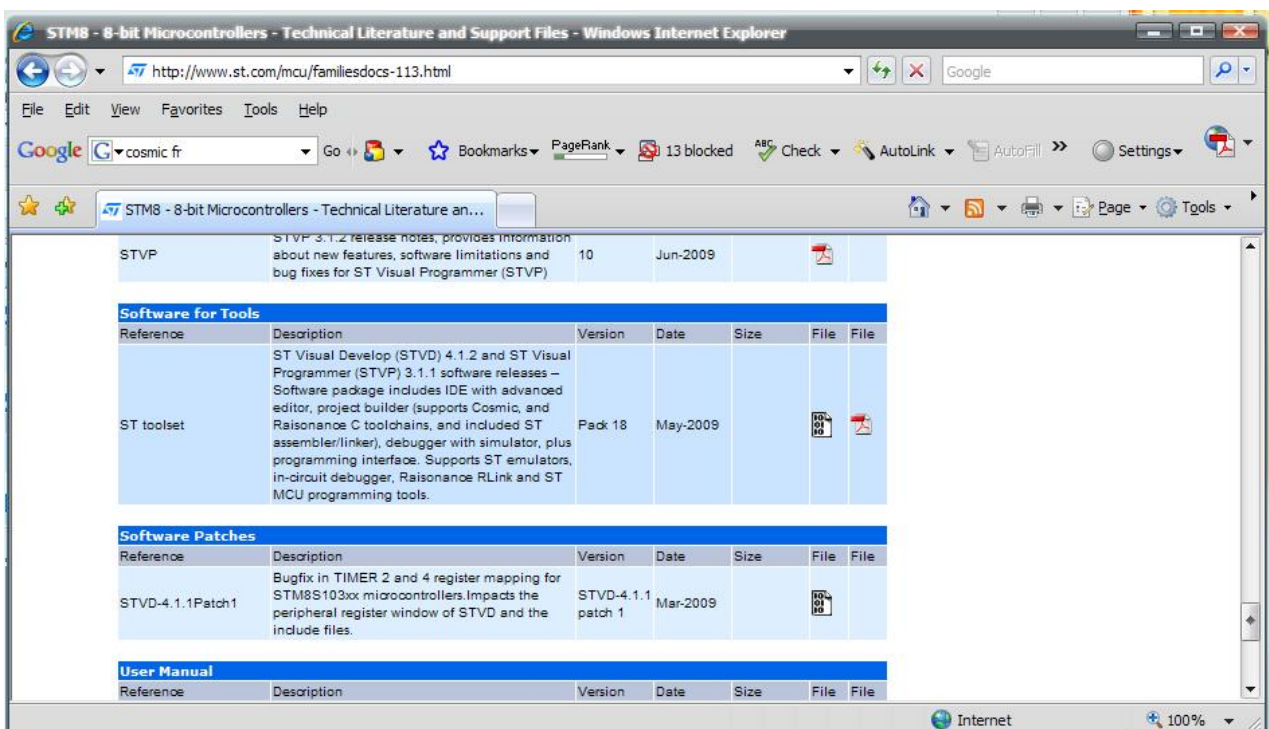
Reference	Description	Version	Date	Size	File	File
	Touch sensing software suite presentation	1.0	Mar-2009	1291 KB		

Internet 100%

รูปที่ 29. ให้เลื่อนลงมาที่รอบ Firmware แล้วดาวน์โหลด STM8S FWLib: STM8S firmware library ซึ่งเป็นไลบรารีสำหรับ Peripherals ที่ ST เตรียมไว้ให้ และดาวน์โหลด STM8S FWLib: STM8S Touch Sensing Library ซึ่งเป็นไลบรารีสำหรับความสามารถ Capacitive Touch-Sensing ที่ ST เตรียมไว้ให้



รูปที่ 30. ให้เลื่อนลงมาที่กรอบ Reference Manual แล้วดาวน์โหลด RM0016: STM8S microcontroller family



รูปที่ 31. ให้เลื่อนลงมาที่กรอบ Software for Tools แล้วดาวน์โหลด ST toolset จากนั้นให้เลื่อนลงมาที่กรอบ Software Patches แล้วดาวน์โหลด STVD-4.1.1 Patch

ให้ทำการลงโปรแกรม sttoolset ซึ่งจะได้อีกก่อน ST Visual Develop (STVD) และ ST Visual Programmer (STVP) ตามรูปที่ 48. จากนั้นให้ลงโปรแกรม stvd-4.1.1patch1 เมื่อลงโปรแกรม STVD เสร็จ โปรแกรม STVD จะเข้าไปค้นหา COSMIC Compiler มาเก็บไว้ใช้งานเอง

ต่อมาให้ลง example ของ Peripherals และ Template project โดยการระเบิดไฟล์ stm8sfwlib_v1_1_0.zip ออกมาแล้ว move ไปไว้ที่ C:\Program Files\STMicroelectronics\

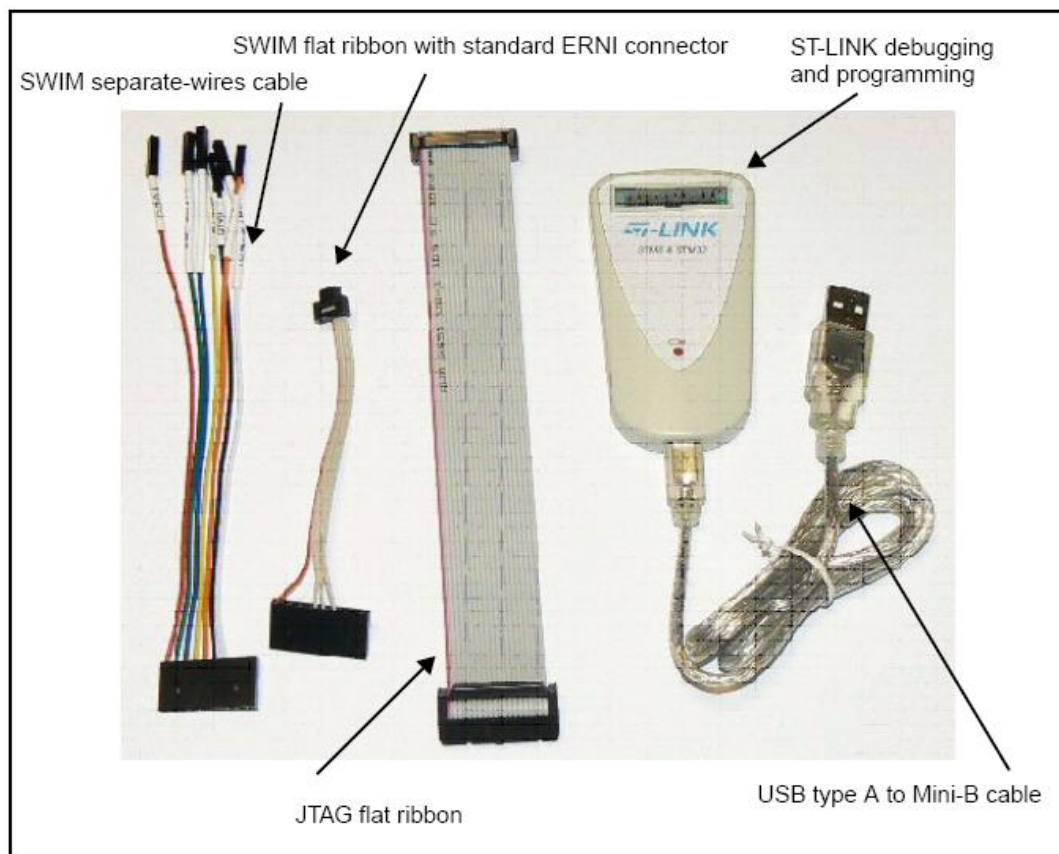
ST-Link

สำหรับ Tools ในการพัฒนาเราจะใช้ ST-Link สำหรับเป็น Programmer/Debugger ตามรูปที่ 32 (หาซื้อได้ที่ ETT) โดยการเชื่อมต่อกับ PC ผ่านทาง USB พอร์ต (ST-Link สามารถใช้ได้ทั้ง MCU 8 bit (SWIM) และ 32 bit (JTAG) ของทาง ST) รายละเอียดอ่านได้ที่ C:\Program Files\STMicroelectronics\st_toolset\stvd\ST-LINK.pdf

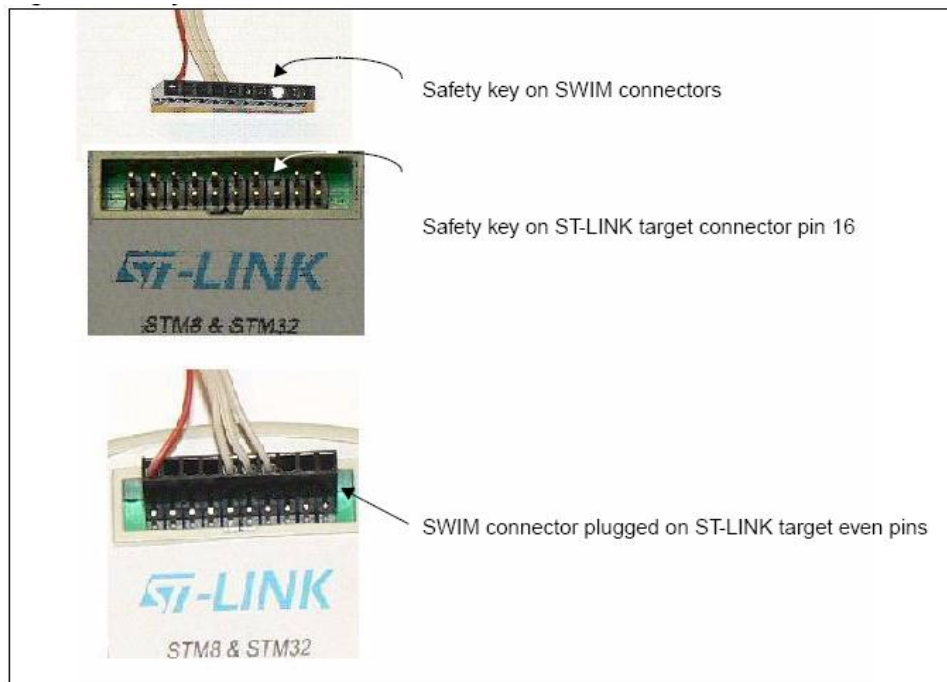
ในชุดของ ST-Link จะประกอบด้วยอุปกรณ์ตามรูปที่ 33. การต่อใช้งาน ST-Link แสดงตามรูปที่ 34.ถึง 38.



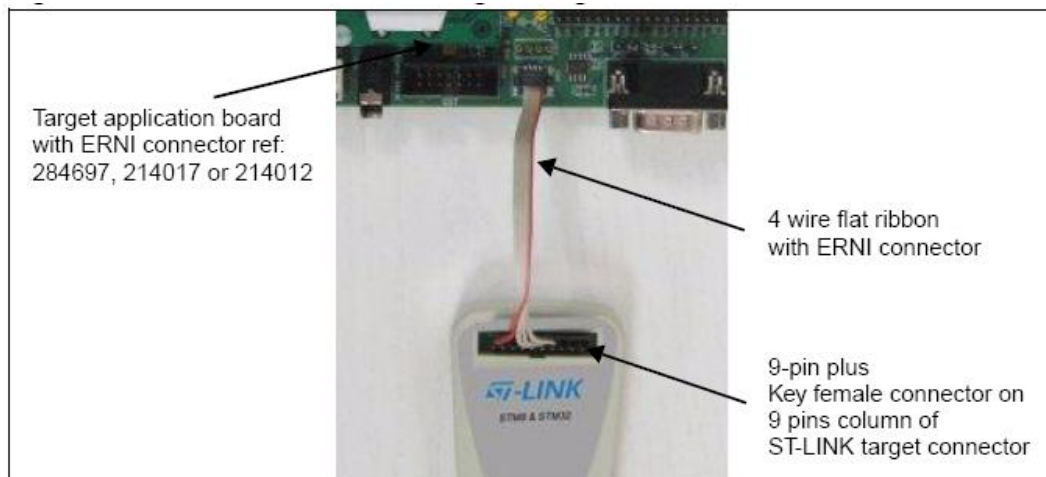
รูปที่ 32. หน้าตาของเครื่อง Programmer/Debugger ของแท่งจาก ST ได้ข่าวว่าสามารถหาซื้อได้ที่ ETT ราคาประมาณ 1,000 บาท



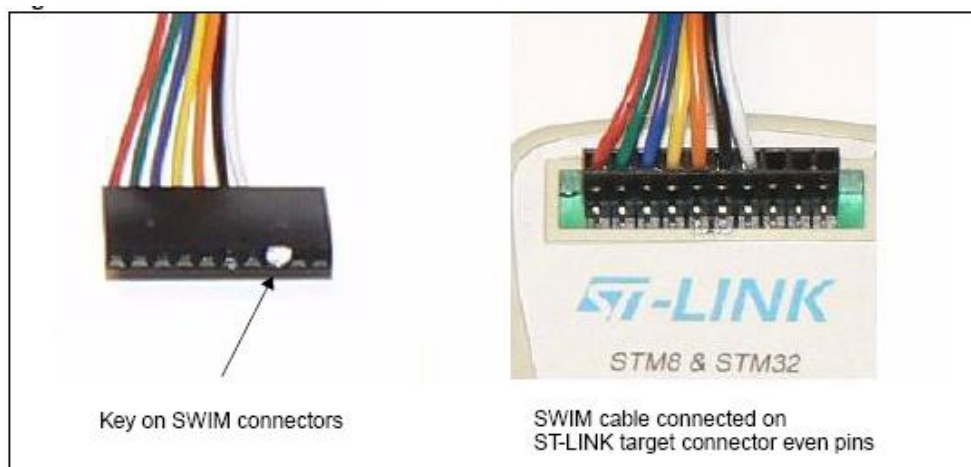
รูปที่ 33. อุปกรณ์ในชุด ST-Link จากซ้ายไปขวา ประกอบด้วย สายเสียบแยกสำหรับ STM8 SWIM (สามารถเสียบเข้า Male IDE Connector ได้), สายแพรเสียบ 4 เส้น สำหรับ STM8 SWIM (แต่คอนเนคเตอร์เป็นแบบเล็กพิเศษ), สายแพร 20 เส้นสำหรับ JTAG (ARM 32bit MCU), ตัวบอดี้ของ ST-Link พร้อมสาย USB



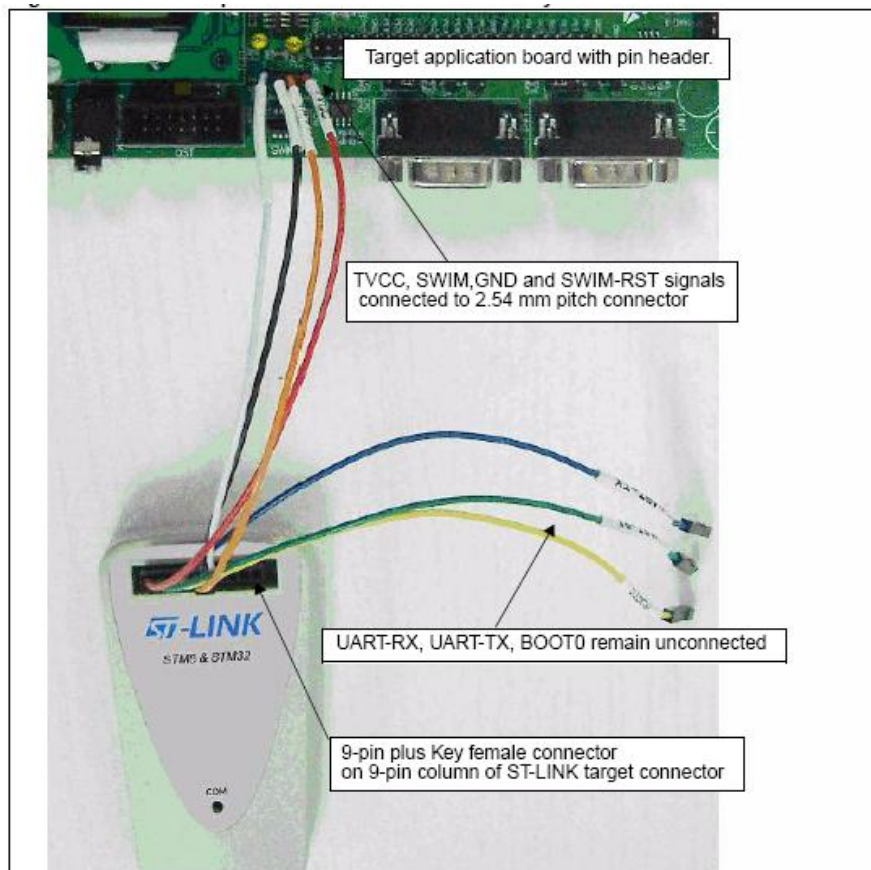
รูปที่ 34. แสดงการต่อสายแพรแบบ 4 เส้น เข้ากับ ST-Link



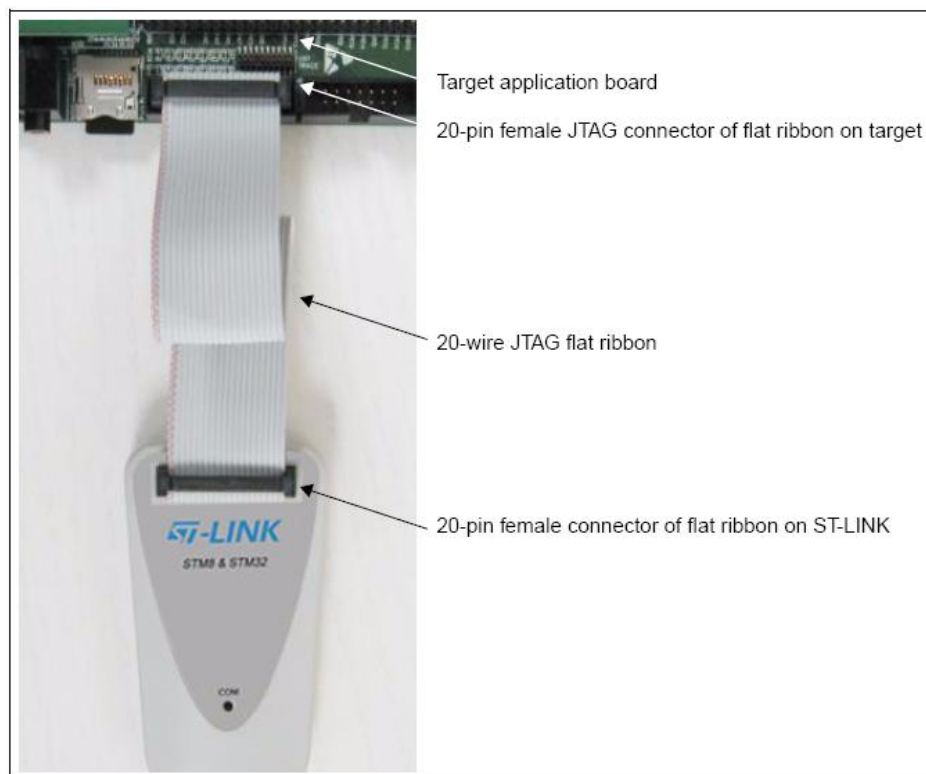
รูปที่ 35. แสดงการต่อสาย SWIM แบบ 4 เส้น เข้าที่พอร์ทของ STM8 Evaluation Board



รูปที่ 36. แสดงการต่อสายแยกแบบ 7 เส้น เข้ากับ ST-Link



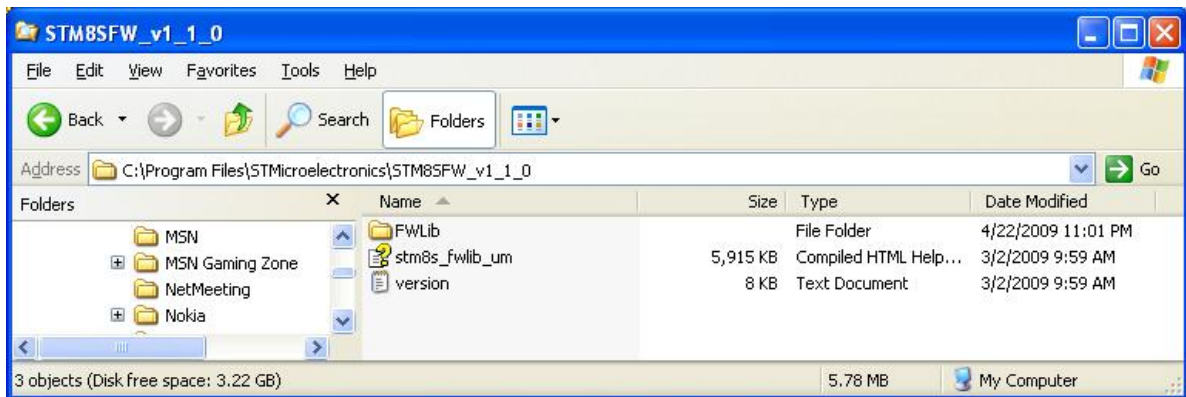
รูปที่ 37. แสดงการต่อสายแยกแบบ 7 เส้น ระหว่าง ST-Link กับบอร์ดที่พัฒนา, สังเกตว่าใช้เพียง 4 เส้น คือ VCC, Data, GND, NRST ในขณะที่ UART-RX, UART-TX, BOOT0 สงวนไว้ใช้ในอนาคต



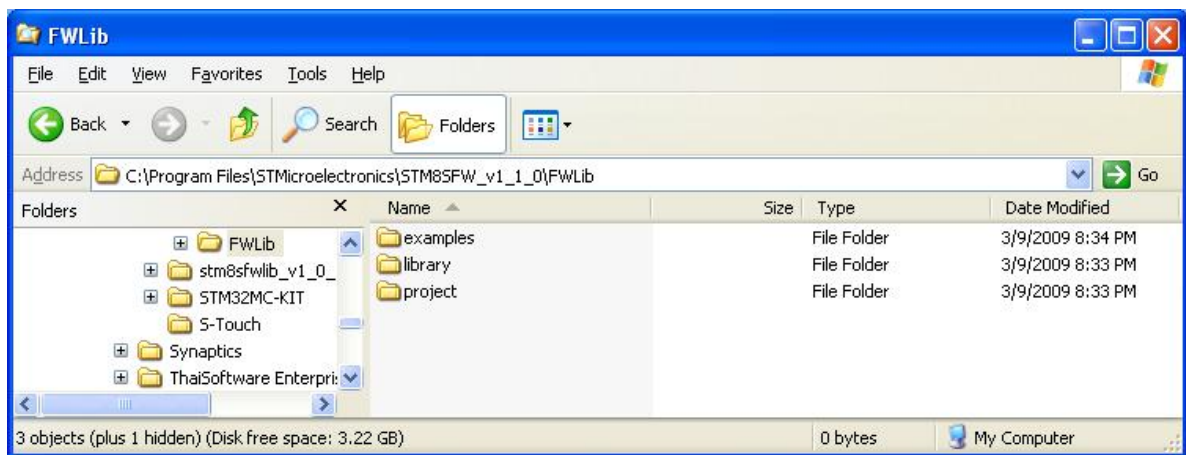
รูปที่ 38. แสดงการต่อ ST-Link เข้ากับบอร์ดที่พัฒนา กรณีที่ใช้กับ ARM 32bit MCU (STM32)

STM8 Firmware Library

ต่อมาเราจะลองลง Firmware Library ของ STM8 ตามรูปที่ 39. โดยภายในโฟลเดอร์ STM8SFW_v1_1_0\FWLib จะประกอบด้วย โฟลเดอร์ examples จะเก็บตัวอย่างการใช้ Peripherals ทั้งหมดของ STM8 ในขณะที่โฟลเดอร์ library และ โฟลเดอร์ project เป็น” Template Project” สำหรับ copy ไปใช้งานตามรูปที่ 40.



รูปที่ 39. แสดงการลง Firmware Library ของ STM8 (ตัวอย่างการใช้งาน Peripherals และ Template Folder) เมื่อ unzip ไฟล์ stm8sfwlib.zip ออกมาจะได้ โฟลเดอร์ C:\Program Files\STMicroelectronics\STM8SFW_v1_1_0\ ตามรูป



รูปที่ 40. ภายใน FWLib จะประกอบด้วย 3 โฟลเดอร์ โดย examples จะเก็บตัวอย่างการใช้ Peripheral ทั้งหมดของ STM8 ในขณะที่ library และ project เป็น”โปรเจกต์เปล่า” สำหรับ copy ไปใช้งานได้เลย (Template Project)

- ในโฟลเดอร์ example จะประกอบด้วยตัวอย่างการใช้งาน Peripherals ของ STM8 ทั้งหมด ตามรูปที่ 41. ประกอบด้วย
- ADC2 สอนการใช้งาน ADC ทั้งในโหมด ต่อเนื่อง และ โหมดแปลงครั้งเดียว
 - AWU สอนการใช้งาน Auto Wake Up Timer
 - BEEP สอนการใช้งาน Hardware Beeper เพื่อตั้งความถี่ 1k,2k หรือ 4kHz
 - CAN สอนการใช้งาน CAN bus 2.0B
 - CLK สอนการใช้ Clock gating สำหรับ Peripherals ทุกตัว
 - Common สอนการใช้งาน LCD และ SD card
 - EXTI สอนการใช้งาน External Interrupt (ทุกพอร์ตของ STM8 สามารถกำหนดเป็นขา External Interrupt ได้)
 - FLASH สอนการใช้งานเขียน/อ่าน Flash Memory (IAP: In Application Programming) และ EEPROM
 - GPIO สอนการใช้งาน General Purpose I/Os
 - I2C สอนการใช้งาน I2C ทั้ง Master และ Slave โหมด
 - ITC สอนการใช้งาน Interrupt Controller
 - IWDG สอนการใช้งาน Independent Watch Dog
 - RST สอนการใช้งาน Reset แบบต่างๆ

SPI สอนการใช้งาน SPI

TIM1 สอนการใช้งาน Advance Timer 1 สำหรับขับเคลื่อนมอเตอร์ 3 เฟส แบบมี dead time ในตัว

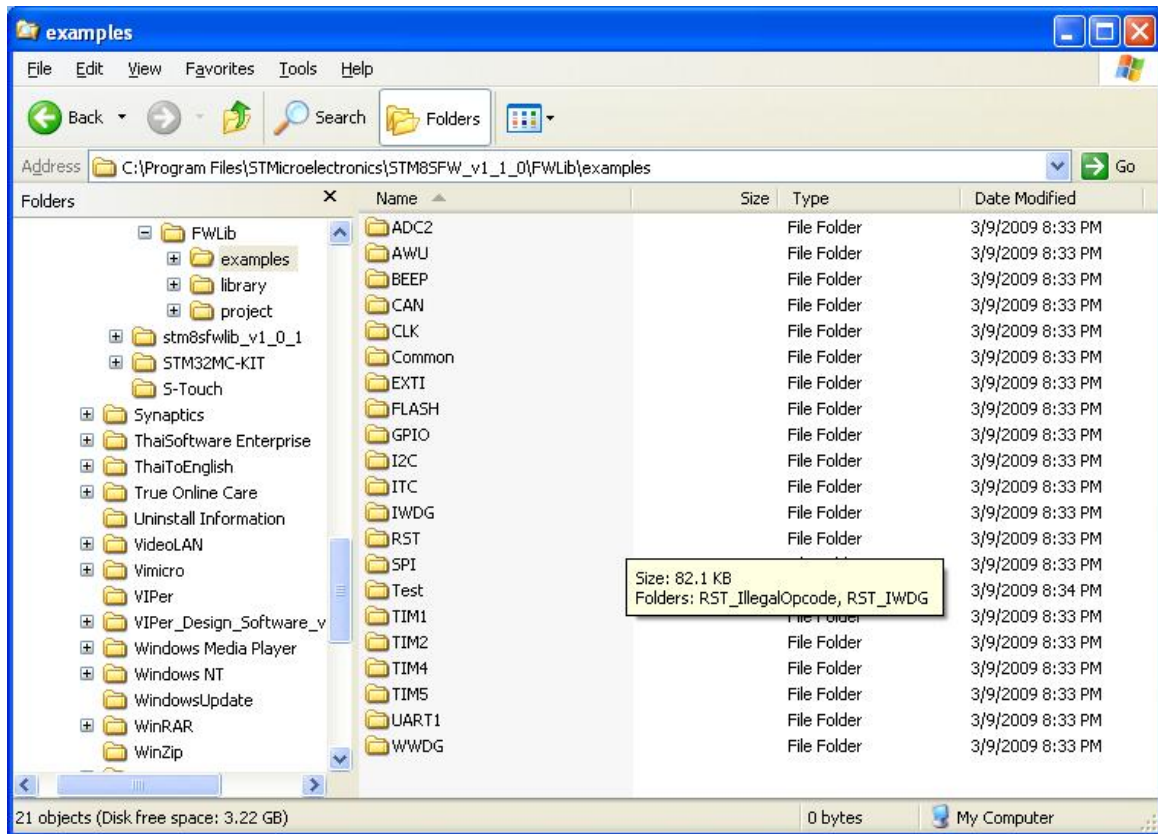
TIM2 สอนการใช้งาน General Purpose Timer 2 และ 3 ขนาด 16 bit

TIM4 สอนการใช้งาน General Purpose Timer 4 ขนาด 8 bit พร้อม prescaler ขนาด 8 bit

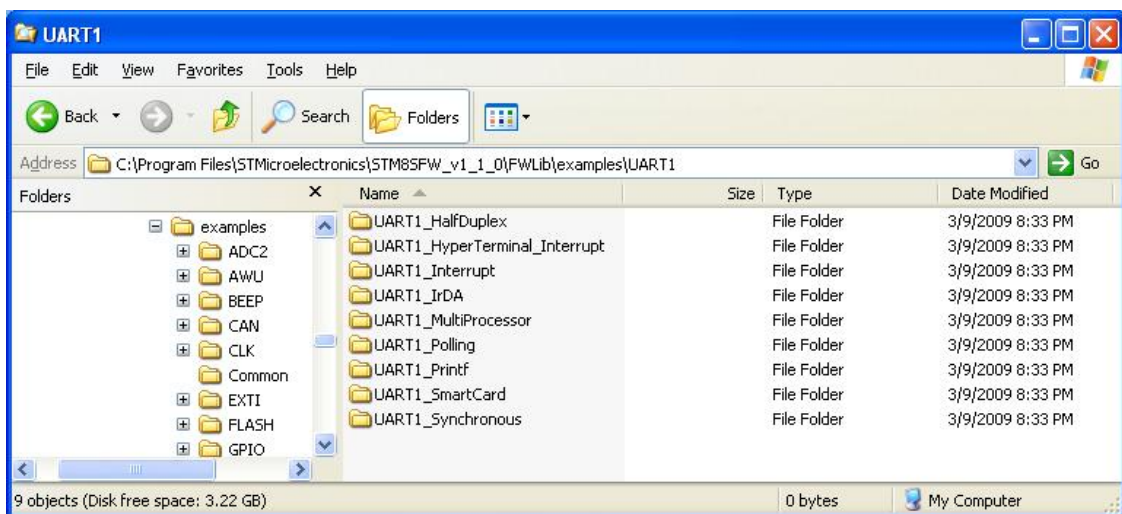
TIM5 สอนการใช้งาน General Purpose Timer 5 ขนาด 8 bit มีในบางเบอร์

UART1 สอนการใช้งาน UART ทั้ง Asynchronous, Synchronous, Single wire Half-Duplex, Smart card, IrDA

WWDG สอนการใช้งาน Window Watch Dog



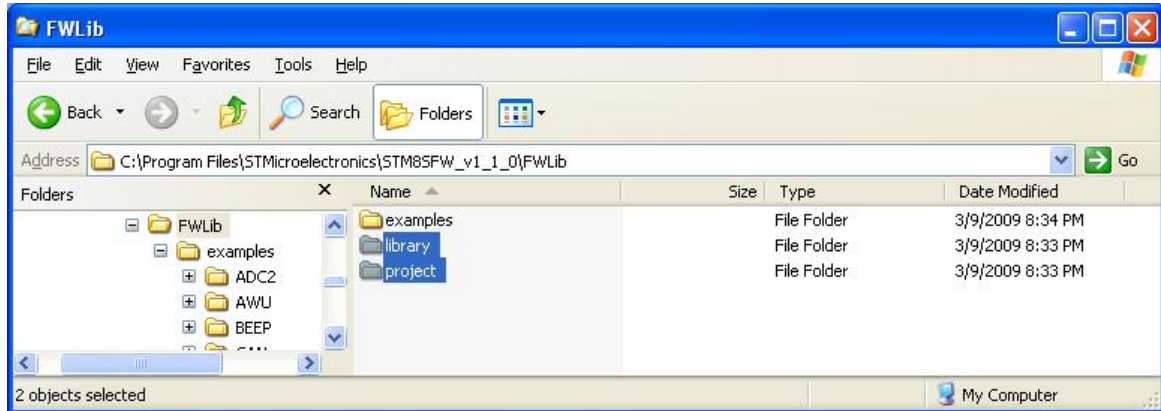
รูปที่ 41. แสดงตัวอย่าง Peripherals ต่างๆ ในไฟล์เดอร์ examples



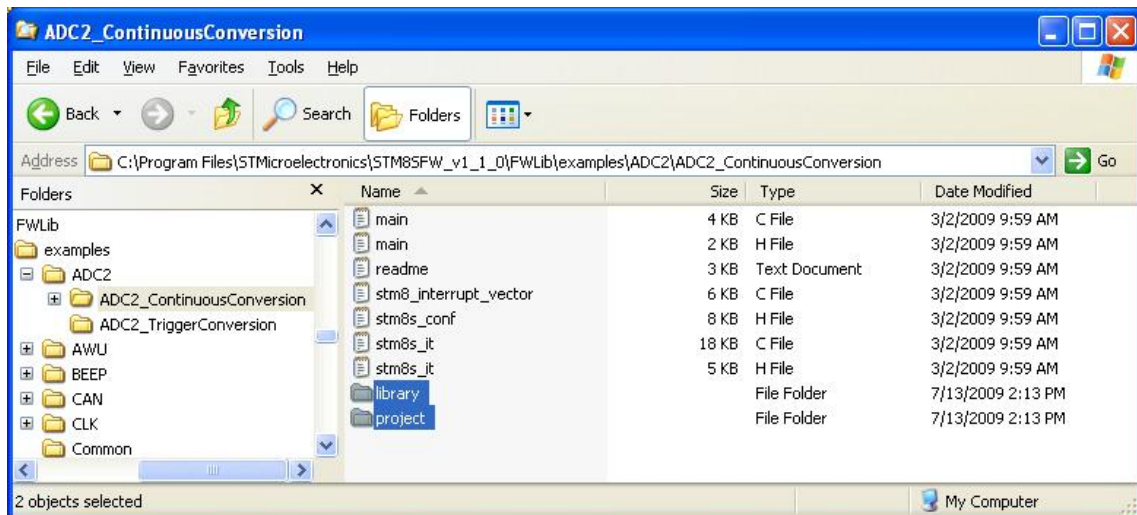
รูปที่ 42. แสดงตัวอย่างย่อยของ Peripheral UART1 ซึ่งประกอบด้วย 9 การทดลอง

ในแต่ละไฟล์เดอร์ของ Peripherals ก็จะมีตัวอย่างย่อยให้ศึกษาอีกมากมาย ยกตัวอย่างเช่น ตัวอย่างย่อยของการใช้งาน UART1 ตามรูปที่ 42. จะพบว่ามากถึง 9 การทดลอง ทั้ง IrDA, SmartCard, Synchronous และ Single wire Half Duplex

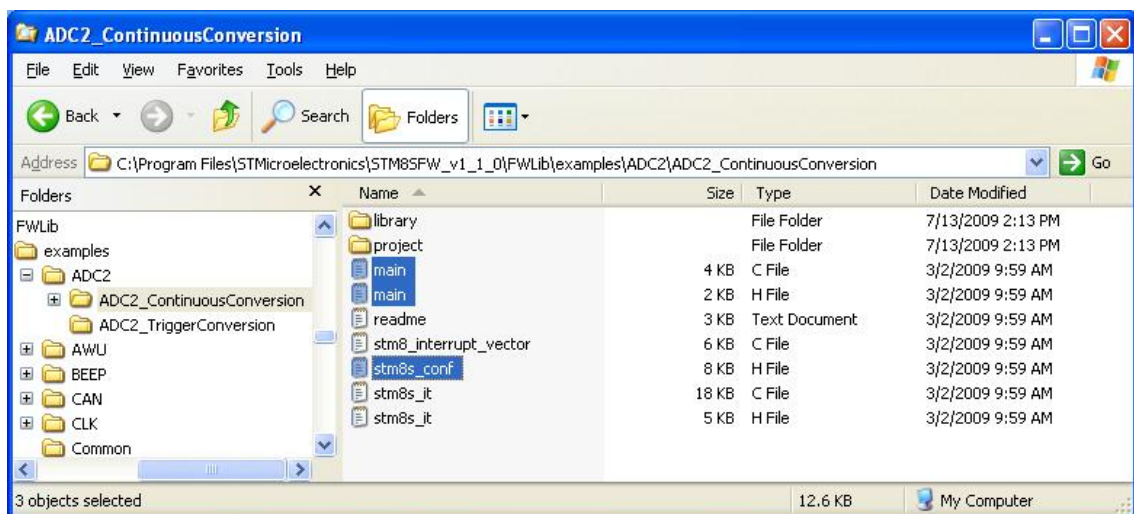
การสร้างโปรเจกของ ST จะแตกต่างจาก MCU เจ้าอื่นๆที่มักจะมี Wizard ไว้คอยช่วยสร้างโปรเจก แต่ของ ST จะใช้ลักษณะเป็น Template Project กล่าวคือเป็น”โปรเจกว่าง” ที่ Link แต่ละส่วนไว้ให้อยู่แล้ว เราเพียงแค่ copy เอาไปใช้งานได้เลย ขอให้ศึกษาจากตัวอย่างต่อไปนี้ สมมุติว่าเราต้องศึกษาการใช้งาน ADC2 แบบ Continuous Conversion (ซึ่งเป็นตัวอย่างอยู่ใน FWLib\examples\ADC2\ADC2_ContinuousConversion\) ให้เราทำการ copy โฟลเดอร์ library และโฟลเดอร์ project ที่อยู่ใน FWLib\ ตามรูปที่ 43. ไปใส่ไว้ในโฟลเดอร์ FWLib\examples\ADC2\ADC2_ContinuousConversion\ ตามรูปที่ 44.



รูปที่ 43. การนำ Template project ไปใช้งาน ให้ copy โฟลเดอร์ library และโฟลเดอร์ project ไปพร้อมกันทั้งคู่



รูปที่ 44. จากนั้นมาวางไว้ในบริเวณที่จะใช้งาน ตามรูปจะลองใช้งานตัวอย่าง ADC2_ContinuousConversion



รูปที่ 45. ให้ copy ไฟล์ main.c, main.h และ stm8s_conf.h ไปทับใน sub project ตามรูป

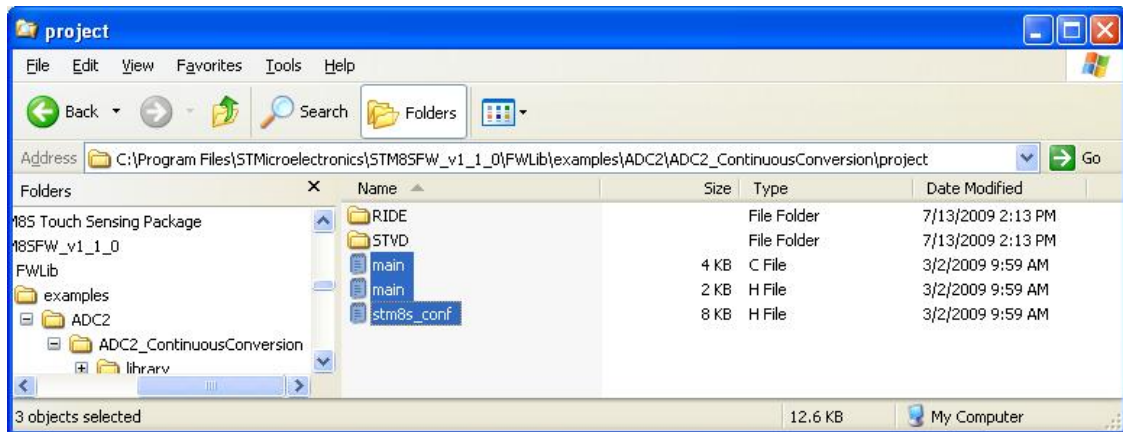
จากนั้นให้ copy ไฟล์ main.c, main.h และ stm8s_conf.h ใน

FWLib\examples\ADC2\ADC2_ContinuousConversion\ ตามรูปที่ 45. ไปทับไฟล์ใน โฟลเดอร์

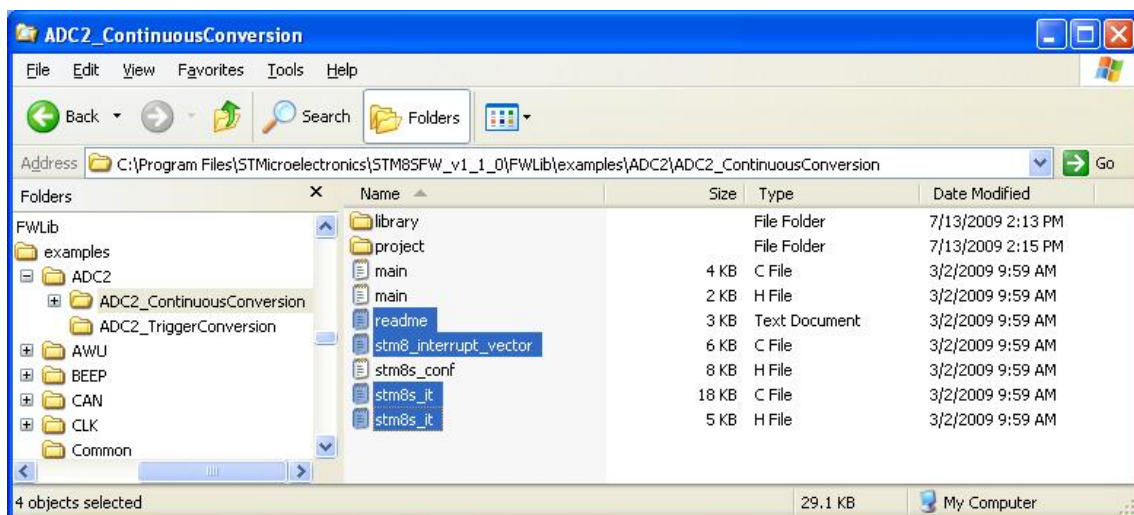
FWLib\examples\ADC2\ADC2_ContinuousConversion\project\ ตามรูปที่ 46. จากนั้นให้ copy ไฟล์ที่เหลือใน

FWLib\examples\ADC2\ADC2_ContinuousConversion\ ซึ่งประกอบด้วย readme.txt, stm8_interrupt_vector.c, stm8s_it.c และ stm8s_it.h ตามรูปที่ 47. ไปทับไฟล์ใน โฟลเดอร์

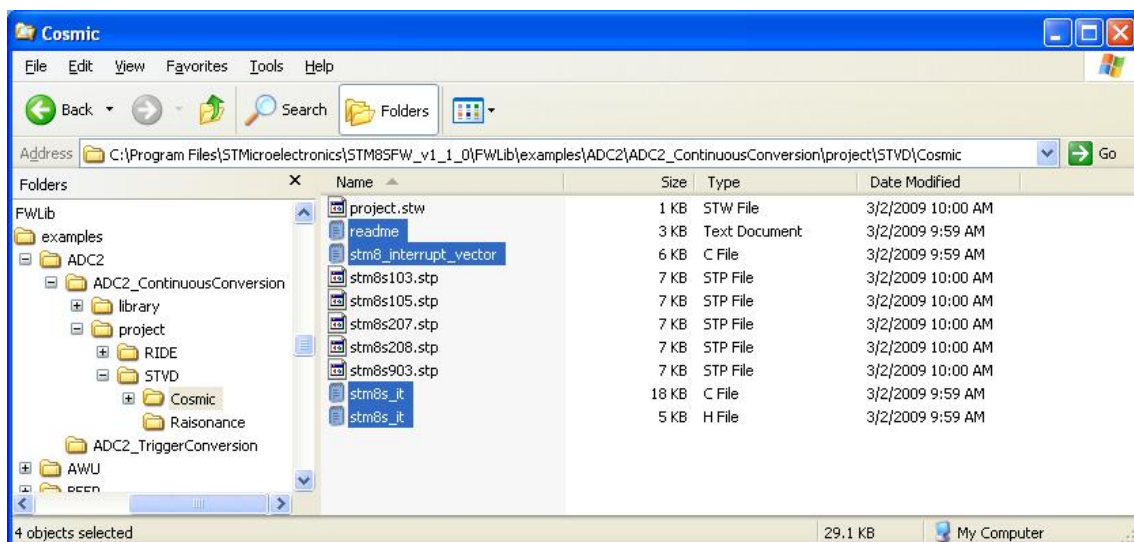
FWLib\examples\ADC2\ADC2_ContinuousConversion\project\STVD\COSMIC\ ตามรูปที่ 48.



รูปที่ 46. ให้ paste ไฟล์ทั้ง 3 ทับไฟล์เก่า ตามรูป



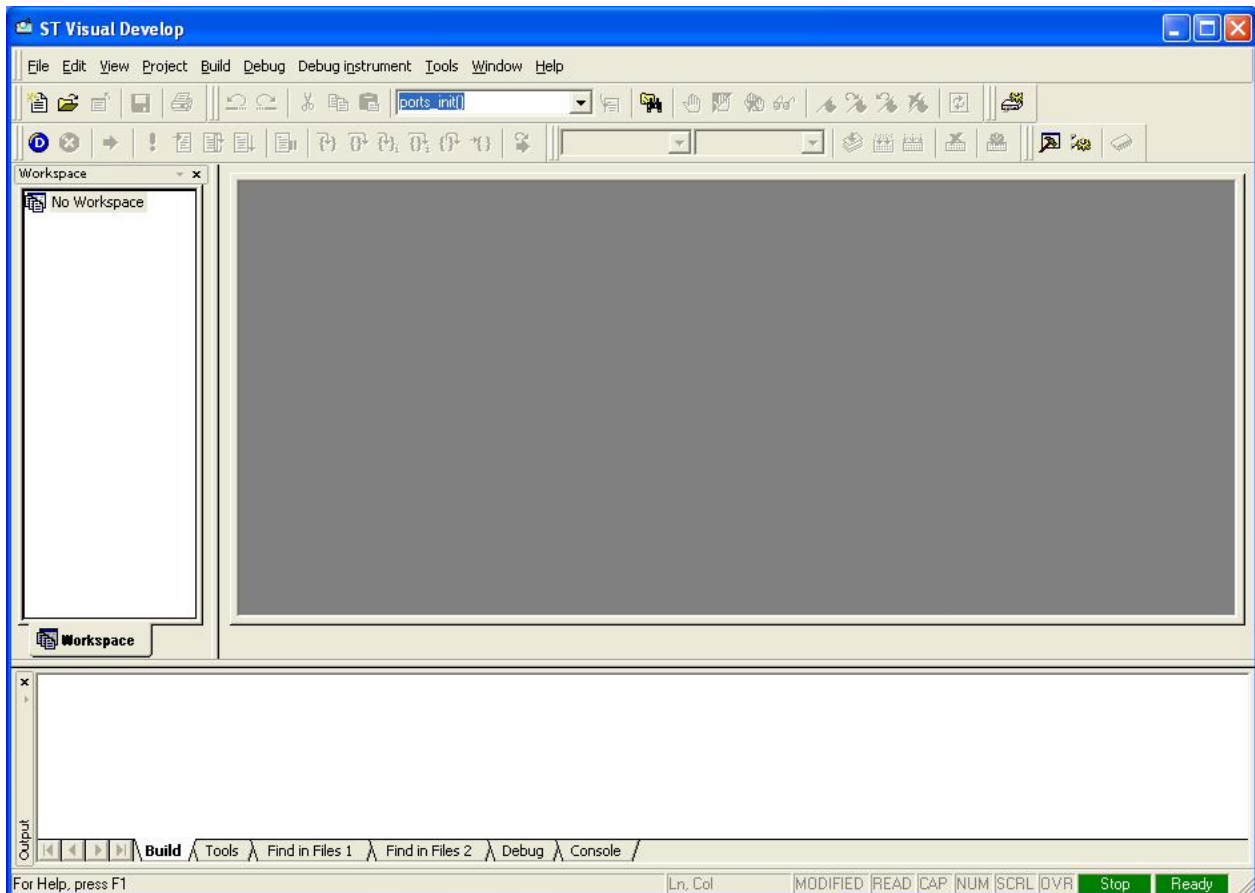
รูปที่ 47. จากนั้นให้ copy ไฟล์ที่เหลือ ประกอบด้วย readme.txt, stm8_interrupt_vector.c, stm8s_it.c และ stm8s_it.h



รูปที่ 48. ไปหับไว้ใน sub project\STVD\Cosmic ตามรูป



รูปที่ 49. แสดงโปรแกรม ST Visual Develop สำหรับเป็นศูนย์กลางการพัฒนา STM8 (เป็น Editor, เป็น Programmer/Debugger)



รูปที่ 50. เมื่อดับเบิลคลิกที่ไอคอน ST Visual Develop จะได้หน้าจอสำหรับพัฒนาดังนี้ ให้เลือกเมนู File > Open Workspace...

ST Visual Develop (STVD)

ต่อมาให้ดับเบิลคลิกที่ไอคอน ST Visual Develop (STVD) ตามรูปที่ 49. จะได้หน้าต่างโปรแกรมตามรูปที่ 50. ไปที่เมนู File > Open Workspace... แล้วเลือกไฟล์ project.stw ในโฟลเดอร์ตามรูปที่ 51. จะปรากฏ workspace ซึ่งประกอบด้วยหลายๆ project ซ้อนกันอยู่ใน workspace เดียวกัน ตามรูปที่ 52. (ST จะทำการแยก MCU แต่ละเบอร์เป็นโปรเจกต์ต่างๆ แยกกัน ถ้าต้องการใช้ MCU เบอร์ไหน ให้คลิกขวาที่โปรเจกต์ของ MCU เบอร์นั้น แล้วเลือก Set As Active Project สังเกตว่าโปรเจกต์นั้นจะกลายมาเป็นอักษรตัวหนา) ให้เปิดไฟล์ readme.txt ขึ้นมา ตามรูปที่ 52. ในไฟล์ readme.txt จะบอกเราว่า การทดลองนี้ต้องการสอนอะไรเรา ต้อง Add ไฟล์ *.c ใน library ตัวไหนบ้างเข้ามาในการทดลองนี้ ยกตัวอย่างเช่น readme.txt ของการทดลองนี้บอกเราว่า

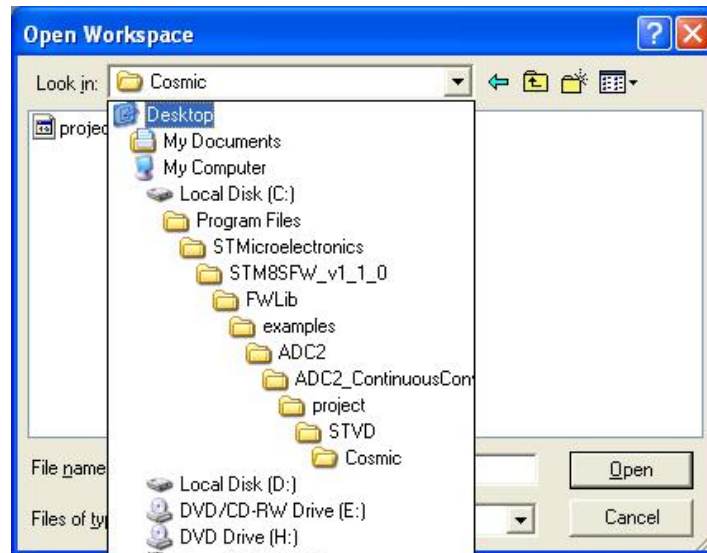
@par How to use it ?

In order to make the program work, you must do the following :

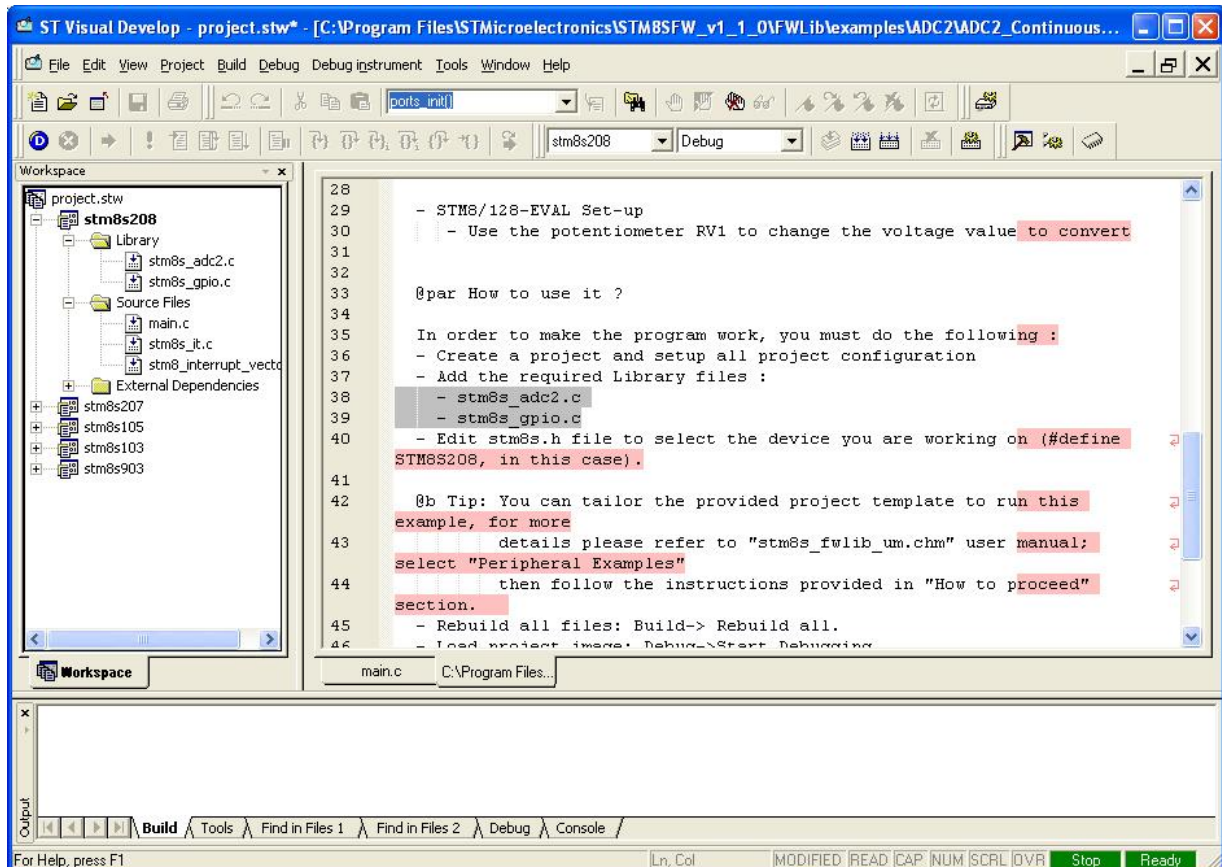
- Create a project and setup all project configuration
- Add the required Library files :
 - stm8s_adc2.c
 - stm8s_gpio.c
- Edit stm8s.h file to select the device you are working on (#define STM8S208, in this case).

นั่นคือให้เราทำการเพิ่มไฟล์ stm8s_adc2.c และไฟล์ stm8s_gpio.c ซึ่งอยู่ในโฟลเดอร์

FWLib\examples\ADC2\ADC2_ContinuousConversion\library\src\ เข้ามารวมไว้ในโปรเจกต์ด้วย วิธีการคือให้เลื่อนเมาส์ไปที่กรอบซ้ายมือของรูปที่ 52. คลิกขวาที่โฟลเดอร์ Library ภายใต้โปรเจกต์ที่ต้องการ (ในรูปที่ 52. คือ stm8s208) จากนั้นเลือก Add Files To Folder... แล้วถอยออกจากโฟลเดอร์ปัจจุบัน 3 ระดับ แล้วเข้าไปที่โฟลเดอร์ library\src\ กดปุ่ม Ctrl ค้างไว้ เลือก stm8s_adc2.c และ stm8s_gpio.c จากนั้นกดปุ่ม Open (ในตัวอย่าง Peripherals ของ ST ทั้งหมดจะเลือก stm8s208 เป็น Default เพราะเป็นตัวที่ใหญ่ที่สุดในตระกูลนี้ ถ้าเราต้องการใช้เบอร์อื่น เช่นเบอร์ stm8s105 ให้เลือก stm8s105 เป็น default project และต้อง uncomment directive #define STM8S105 ในไฟล์ stm8s.h ด้วย)



รูปที่ 51. ลองเข้าไปในตัวอย่าง ADC2_ContinuousConversion ตามโฟลเดอร์ในภาพ จากนั้นเลือกไฟล์ project.stw



รูปที่ 52. ใน Firmware Library ของ ST version 1.1.0 จะเตรียม MCU เบอร์ต่างๆให้เลือกเป็นโปรเจกต์ ตามกรอบซ้ายมือในรูป

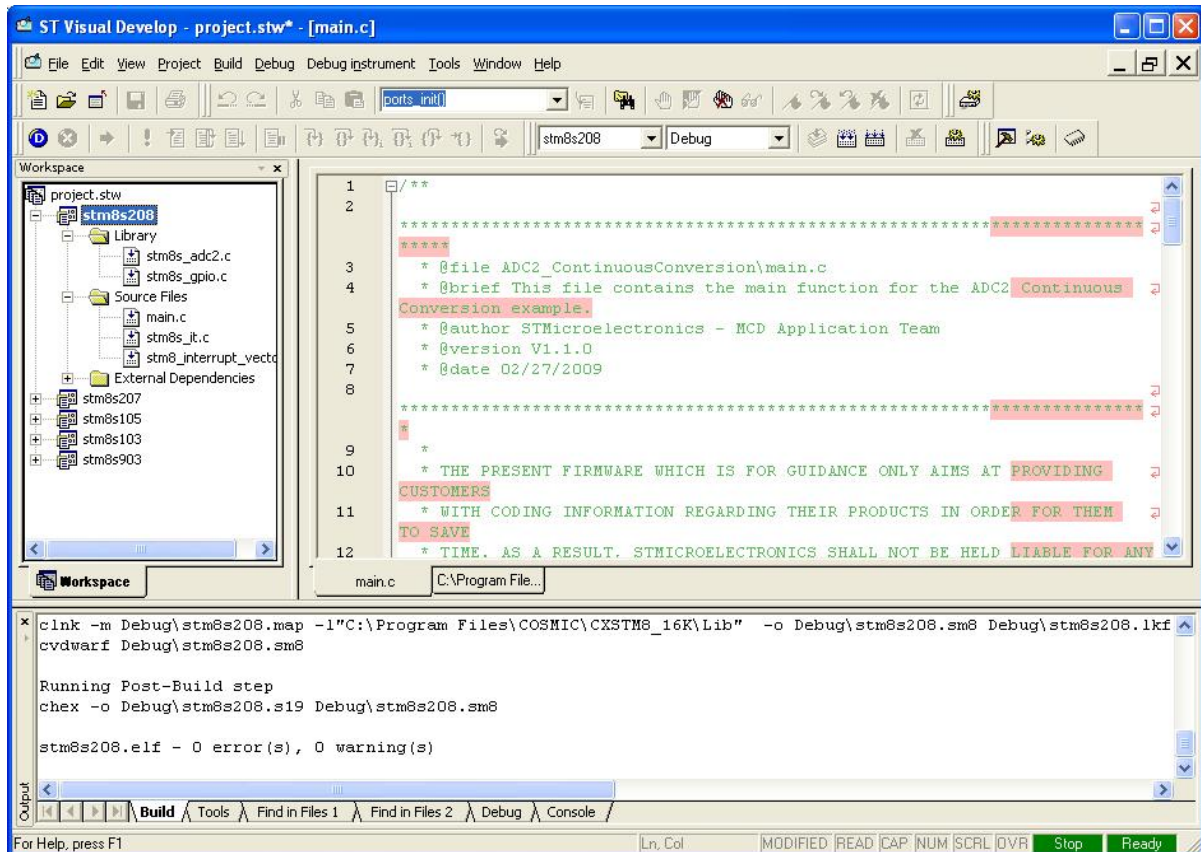


รูปที่ 53. ประกอบด้วยปุ่ม Compile, ปุ่ม Build และปุ่ม Rebuild All จากซ้ายไปขวา (ปรกติมักจะใช้ปุ่ม Rebuild All)

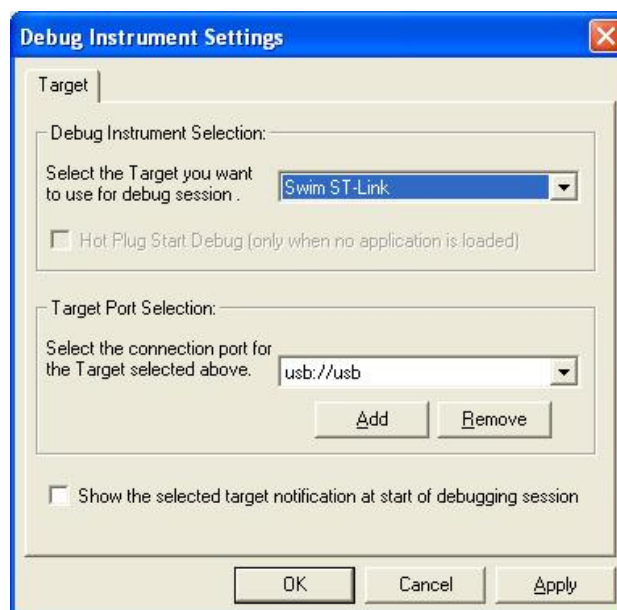
จากนั้นให้กดปุ่ม Rebuild All ตามรูปที่ 53. เพื่อ compile และ Link สร้างไฟล์.s19 เพื่อจะนำไปโปรแกรมลง MCU ต่อไป ถ้าทำทุกอย่างถูกต้อง ในกรอบล่างของหน้าต่างจะบอกว่า

0 error(s) , 0 warning(s)

ตามรูปที่ 54. (ถึงแม้จะมี warning บ้างก็ไม่เป็นความผิดปกติใดๆ สามารถใช้งานได้)



รูปที่ 54. ถ้าทุกอย่างถูกต้อง ในกรอบด้านล่างจะไม่แจ้ง error หรือ warning ใดๆ ตามรูป



รูปที่ 55. ให้เลือกเครื่องมือสำหรับการโปรแกรม/ดีบั๊ก จากเมนู Debug instrument > Target Setting... ในช่อง Debug Instrument Selection: ให้เลือก Swim ST-Link ตามรูป

จากนั้นให้ไปที่เมนู Debug instrument > Target Setting... ในช่อง Debug Instrument Selection: ให้เลือก Swim ST-Link ตามรูปที่ 55. เพื่อเลือก ST-Link เป็นเครื่องมือสำหรับ Program/Debug เมื่อต้องการดีบั๊กให้กดปุ่มสีฟ้า ซ้ายมือสุดของรูปที่ 56. STVD จะทำการติดต่อกับ ST-Link แล้วโปรแกรมบริเวณ Flash memory พร้อมกับการเข้าโหมดดีบั๊ก (รายละเอียดเพิ่มเติมจะขอกล่าวถึงในบทความต่อไป)

ถ้าโปรแกรมไม่ได้ อาจเกิดจาก STVD ติดต่อกับ ST-Link ไม่ได้ หรือบอร์ดที่เราพัฒนาขึ้นมายังไม่ได้จ่ายไฟเลี้ยง



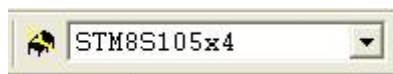
รูปที่ 56. เมื่อต้องการโปรแกรม Flash memory พร้อมกับการดีบั๊ก ให้กดปุ่มสีน้ำเงิน ซ้ายมือสุดตามรูป (ปุ่มที่เหลือใช้สำหรับการดีบั๊ก)

ST Visual Programmer (STVP)

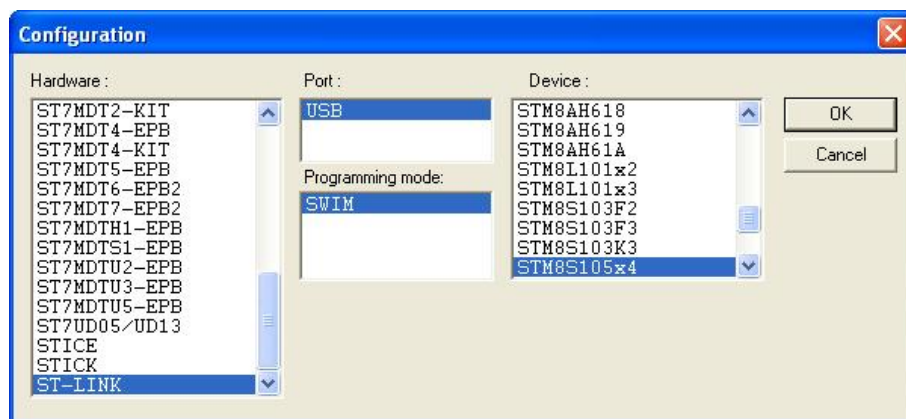
ต่อมาเป็นการใช้งานโปรแกรม STVP สำหรับ Burn Flash memory, EEPROM data และ Option Byte ให้ดับเบิลคลิกที่ไอคอนตามรูปที่ 57. จะได้หน้าต่างตามรูปที่ 60. ให้กดปุ่มด้านซ้ายของบอร์ด MCU ตามรูปที่ 58. เพื่อเลือกเครื่องมือพัฒนา และบอร์ดของ MCU ที่จะใช้ (ในที่นี้สมมุติว่าจะใช้ ST-Link และ MCU บอร์ด STM8S105K4T6) จากนั้นเลือกตามรูปที่ 59. กด OK



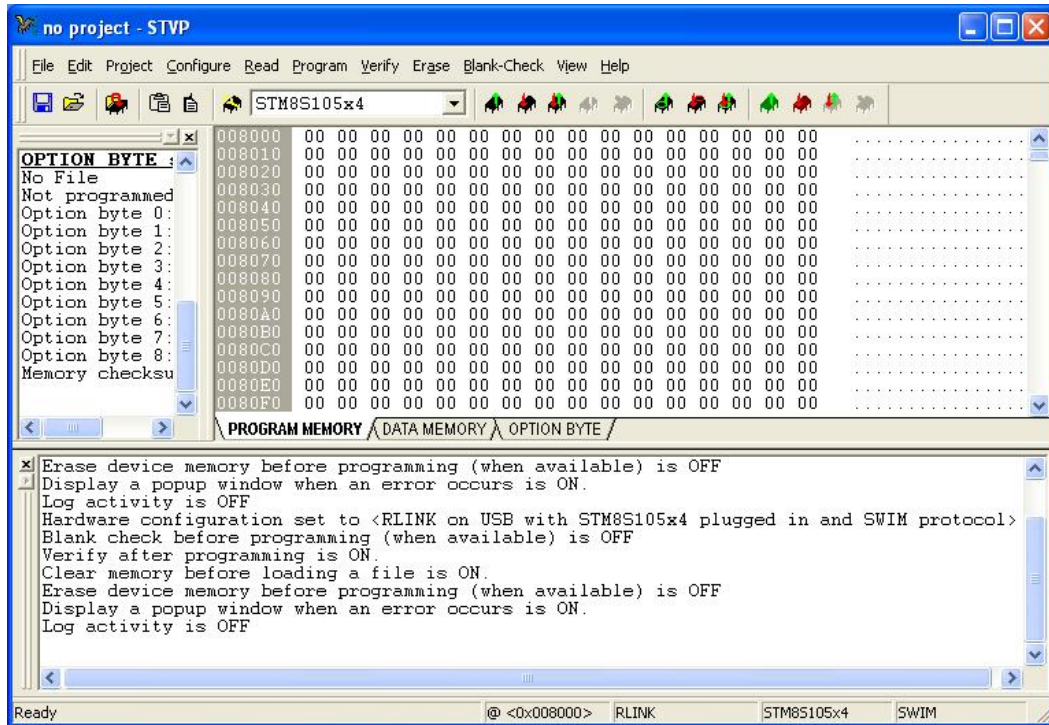
รูปที่ 57. แสดงโปรแกรม ST Visual Programmer สำหรับ Program STM8 MCU ผ่าน ST-Link โดยไม่ต้องดีบั๊ก



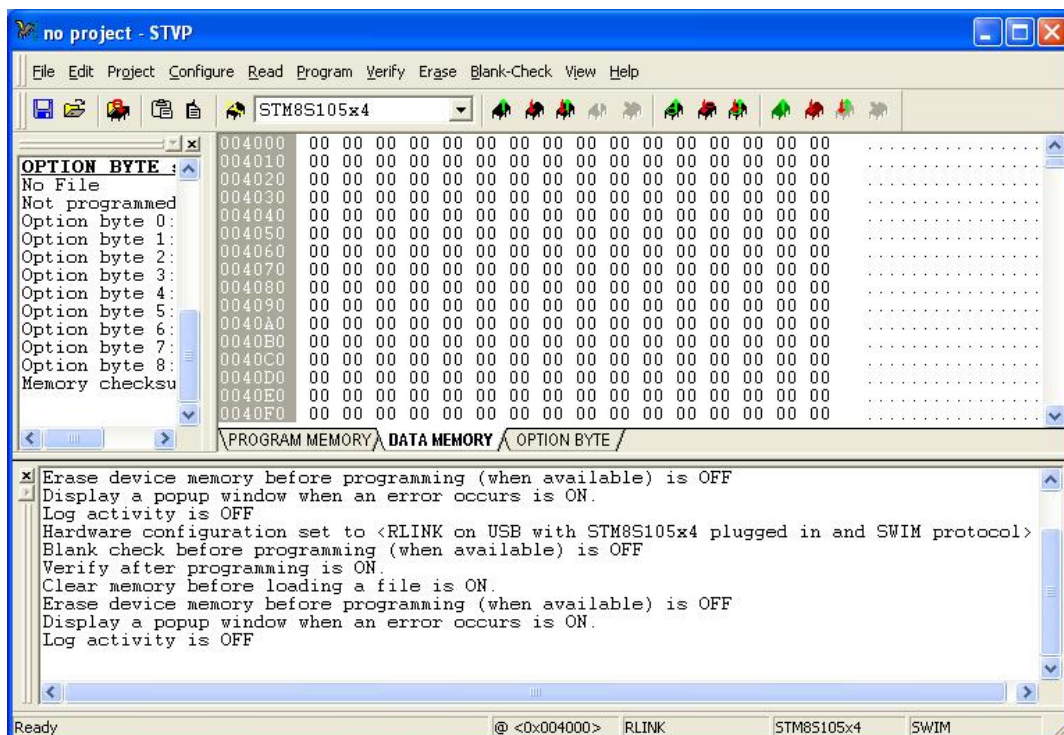
รูปที่ 58. กดปุ่มด้านซ้ายของบอร์ด MCU ตามรูป เพื่อเลือกเครื่องมือพัฒนา และบอร์ดของ MCU ที่จะใช้



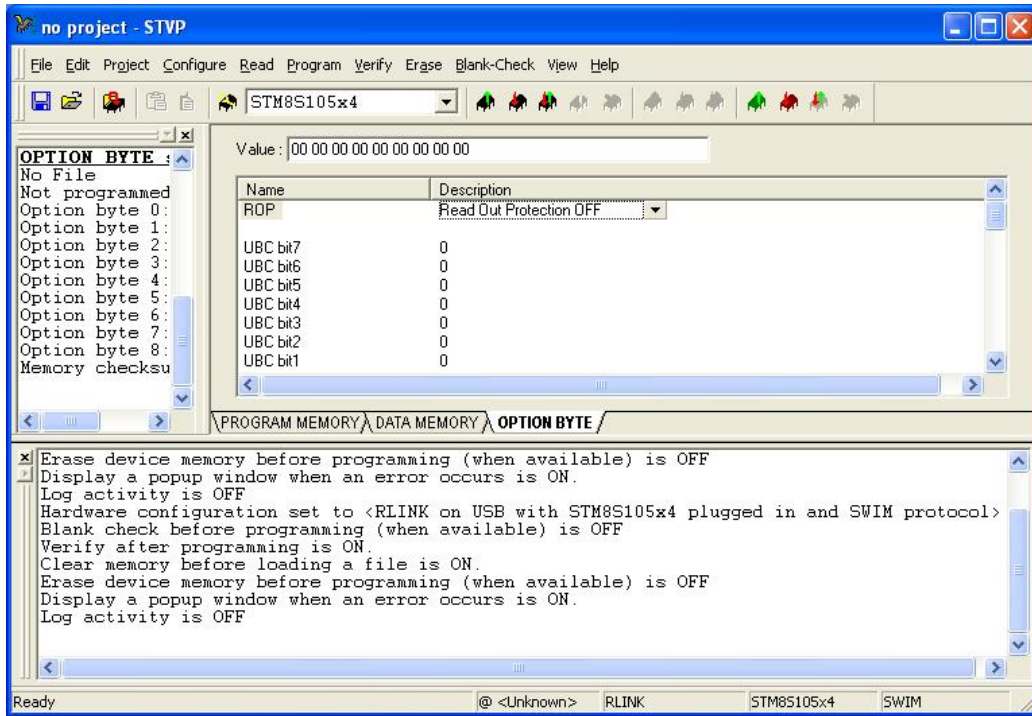
รูปที่ 59. ในกรอบ Hardware ให้เลือก ST-LINK ส่วนในกรอบ Device: ให้เลือกบอร์ด STM8 ที่ต้องการพัฒนา



รูปที่ 60. แท็บแรกของ PROGRAM MEMORY ซึ่งไฟล์ที่จะดาวน์โหลดจะอยู่ในรูปของ *.s19 เริ่มต้นที่แอดเดรส 0x8000



รูปที่ 61. แท็บที่สองของ DATA MEMORY (EEPROM AREA) ซึ่งไฟล์ที่จะดาวน์โหลดจะอยู่ในรูปของ *.s19 เช่นกัน แต่เริ่มต้นที่แอดเดรส 0x4000



รูปที่ 62. แท็บที่สามของ OPTION BYTE ซึ่งใช้จัดการป้องกันการเขียน, ป้องกันการอ่าน, ป้องกันการเขียนเป็นส่วนย่อยๆ, ใช้รีแมพขาทางเลือกต่างๆ, ใช้กำหนดการอนุญาต watch dog, ใช้ปรับละเอียด HSI และอนุญาตการใช้ฟังก์ชัน Boot Loader (พอยเทียบเคียงได้กับการเซ็ต configuration bits ของ Microchip)

ในหน้าจอของ STVP จะแบ่งออกเป็น 3 แท็บ ประกอบด้วย Program memory Tap, Data memory Tap และ Option Byte Tap ตามรูปที่ 60. ถึง 62. โดยสามารถโหลดโปรแกรมแยกส่วนกันมาลงก็ได้ หรือทำการโหลดทั้ง 3 แท็บเข้ามาพร้อมกันทีเดียว โดยการสร้างไฟล์ทั้ง 3 ในรูปของ program project (*.stp) ซึ่งขั้นตอนนี้ต้องเซตบน STVD ขั้นตอนสุดท้ายคือการ อ่านและเขียนลง MCU ผ่าน ST-Link ซึ่งรายละเอียดแสดงในรูปที่ 63. (รายละเอียดของ Option Byte ในแต่ละเบอร์จะแตกต่างกันไป ขอให้ท่านอ่านใน Datasheet แต่พอสรุปกว้างๆได้ว่า ประกอบด้วย ROP: Read Out Protect, UBC: User Block Code ใช้ป้องกันการเขียน Flash เป็นบล็อกละเลเยอร์, AFR: Alternative Function Register ใช้ในการ Remap ขา MCU, ส่วน Trim HSI OSC, ส่วนเกี่ยวกับ Watchdog, ส่วนเกี่ยวกับการใช้ HSE OSC และส่วนการอนุญาตให้ใช้ Boot Loader)



รูปที่ 63. ปุ่มที่ใช้ในการ อ่าน/เขียน แบ่งเป็น 3 กลุ่ม ที่ใช้บ่อยคือกลุ่มซ้ายมือ ใช้ในการ อ่าน/เขียน เฉพาะแท็บที่เลือก สีเขียวปุ่มแรกใช้ในการอ่าน สีแดงปุ่มที่สองใช้ในการเขียน ส่วนกลุ่มขวามือสุด ใช้ในการ อ่าน/เขียน ทุกแท็บที่ เช่นกันสีเขียวปุ่มแรกใช้ในการอ่าน สีแดงปุ่มที่สองใช้ในการเขียน

จะเห็นได้ว่า MCU ตัวใหม่นี้ มีความน่าสนใจ ทั้งประสิทธิภาพที่สูง (กว่า 20MIPs at 24MHz), Peripherals ต่างๆที่ครบครัน และราคาที่แสนจะประหยัด รวมทั้งสามารถประยุกต์ทำปุ่มกดแบบสัมผัสได้ในตัว ทำให้ Product คุ้มราคาและประหยัดเงินค่าปุ่มกดไปไม่น้อย เมื่อมีบอร์ดพัฒนารุ่นใหม่จาก ETT ออกมา ร่วมกับ Programmer/Debugger รุ่น ST-Link คงจะทำให้ MCU ตระกูลนี้ เป็นที่นิยมในตลาด 8 bit ได้ไม่ยาก

ในโอกาสต่อไป ผมจะกล่าวถึง ตัวอย่างการใช้งาน Peripherals ที่สำคัญแต่ละอย่าง รวมทั้งวิธีการลงโปรแกรมและการใช้งาน Capacitive Touch-Sensing Firmware Library ขั้นตอนการออกแบบลายวงจรสำหรับ Touch-Sensing (TS) วิธีการปรับแต่งพารามิเตอร์ของ TS รวมทั้งการเข้าโหมดดีบัก เพื่อทำการ tuning แบบ real time

พงษ์ศักดิ์ พร้อมวงศ์ ()

เทียนไชย นกครุฑ (tienchai.n@gmail.com)