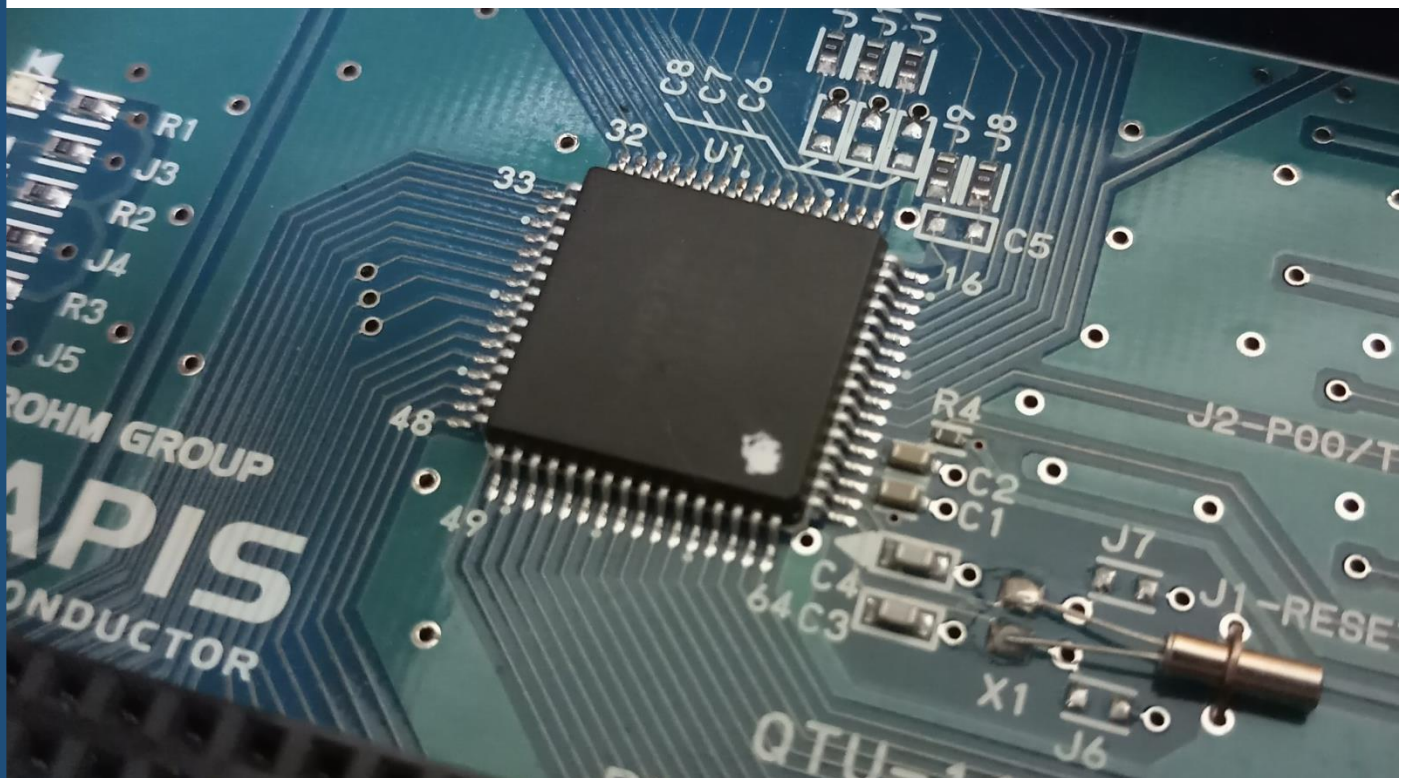


LAPIS ML62Q1622

16-Bit Microcontroller

ML62Q1600 Group

คู่มือการใช้งานเบื้องต้น



จัดทำโดย

บริษัท อิเล็คทรอนิกส์ ซอร์ซ จำกัด

LAPIS ML62Q1622
16-Bit Microcontroller
ML62Q1600 Group
คู่มือการใช้งานเบื้องต้น

บทนำ

บอร์ด LAPIS ได้ถูกพัฒนาขึ้นเพื่อเป็นตัวอย่างการใช้งานไมโครคอนโทรลเลอร์ของ LAPIS Semiconductor เบอร์ ML62Q1622 เพื่อเป็นพื้นฐานสำหรับ วิศวกร ช่างเทคนิค นักวิจัย นักศึกษาและนักอิเล็กทรอนิกส์สมัครเล่น ที่มีความสนใจที่จะเริ่มต้นการพัฒนาและเรียนรู้การใช้งาน LAPIS ML62Q1622 โดยอุปกรณ์ที่ใช้ในบอร์ด LAPIS เป็นอุปกรณ์พื้นฐานที่สามารถหาได้ทั่วไปในท้องตลาด บนบอร์ด LAPIS นอกจากจะมี LAPIS ML62Q1622 เป็นหัวใจหลักแล้วยังมี LCD รุ่น GTE3027SA ที่เป็น LCD แบบ 6 Digits ซึ่งเป็นส่วนสำคัญที่จะช่วยให้ผู้ใช้มีความเข้าใจหลักการทำงานระหว่าง LAPIS เบอร์ ML62Q1622 กับจอ LCD

คู่มือการใช้งานแบ่งเป็น 4 บท โดยบทแรกจะแสดงส่วนประกอบทางฮาร์ดแวร์ของบอร์ด Lapis การปรับตั้งหน้าที่ของแต่ละส่วน การเชื่อมต่อระหว่างบอร์ด LAPISกับคอมพิวเตอร์และข้อควรระวังในการใช้งาน ส่วนที่สองอธิบายเกี่ยวกับการติดตั้งและใช้งานโปรแกรมบนคอมพิวเตอร์ร่วมกับบอร์ด LAPIS บทที่ 3 เป็นการทดลองบอร์ด LAPIS เช่นการแสดงผลบนหน้าจอ LCD การรับค่าจากคีย์สวิตช์ เป็นต้น บทที่ 4 เป็นการอธิบายรายละเอียด Register เบื้องต้น

บทที่ 1 Hardware

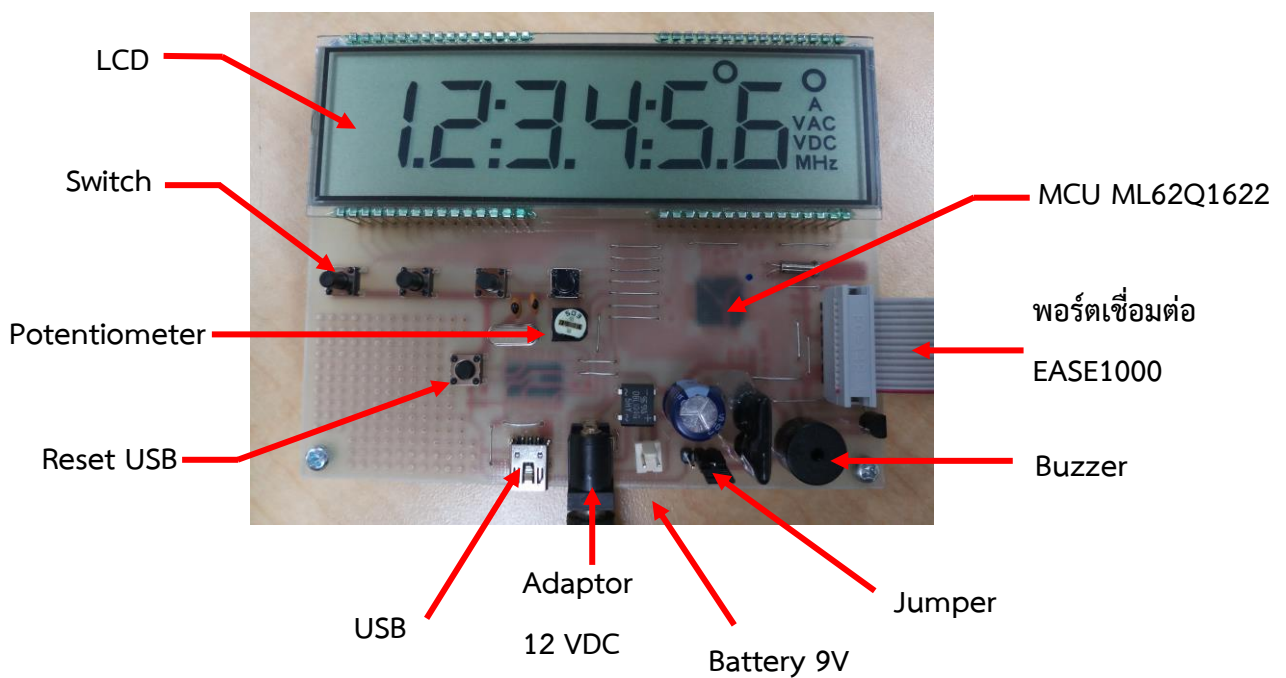
1) บอร์ด LAPIS

บอร์ด LAPIS นี้ได้นำเสนอตัวอย่างการนำไมโครคอนโทรลเลอร์ LAPIS ML62Q1622 เพื่อการใช้งานจริง บอร์ดประกอบด้วยวงจรพื้นฐานอย่างเช่น LCD Display วงจรรับคีย์จำนวน 4 คีย์ วงจรขยายเสียง เป็นต้น ซึ่งวงจรเหล่านี้มีประโยชน์อย่างยิ่งสำหรับการใช้ในการพัฒนาและทดสอบโปรแกรมพื้นฐาน

1.1) อุปกรณ์บนบอร์ด

บอร์ดจะประกอบไปด้วยอุปกรณ์ดังนี้

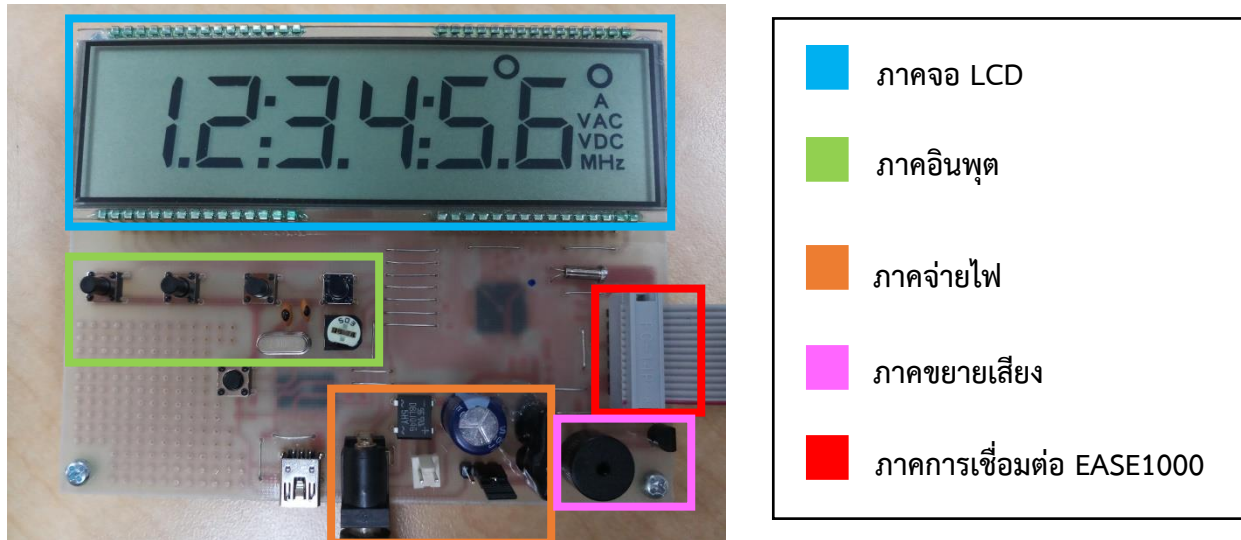
- 1) MCU LAPIS เบอร์ ML62Q1622
- 2) LCD รุ่น GTE3027SA เป็น LCD แบบ 6 Digits
- 3) คีย์สวิตช์ 4 คีย์
- 4) ลำโพง Buzzer
- 5) Potentiometer
- 6) Reset USB
- 7) USB
- 8) Adaptor 12 VDC
- 9) Battery 9V
- 10) Jumper
- 11) พอร์ตเชื่อมต่อ EASE1000



รูปที่ 1

1.2) องค์ประกอบของบอร์ด

- 1) ภาควินพุต
- 2) ภาควจอ LCD
- 3) ภาควการเชื่อมต่อ EASE1000
- 4) ภาควจ่ายไฟ
- 5) ภาควขยายเสียง



รูปที่ 2

2) คุณสมบัติของ MCU ML62Q1622

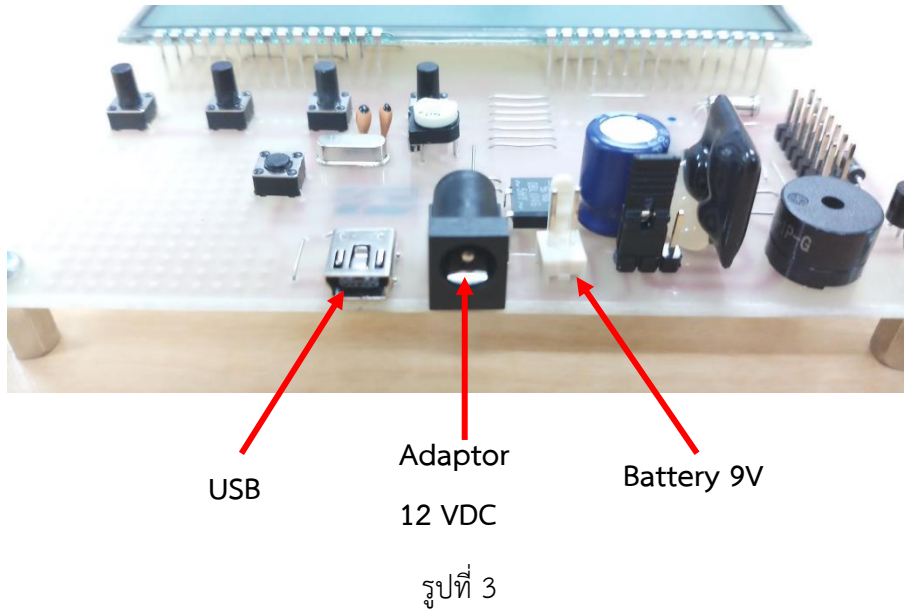
MCU ML62Q1622 เป็นไมโครคอนโทรลเลอร์ชนิด 16-Bit ของบริษัท LAPIS Semiconductor โดยคุณสมบัติของ MCU มีดังนี้

Operating Voltage	Max CPU Clock Freq. (On-chip)	CPU	Pin count	ROM	RAM
1.8 – 5.55 V	24MHz	16bit	64	64 KB	4 KB

Datat Flash	LCD driver (Max.dot)	Timer	PWM	Serial	ADC	DAC
2 KB	288	16bit x 12	16bit x 6	UART/SSIO x 2 I2C x 3	10bit x 12	8bit x 1

3) การเชื่อมต่อและการปรับตั้ง

ภาคแหล่งจ่ายไฟของบอร์ด LAPIS ได้รวมวงจรเรียงกระแส วงจรกรอง และวงจรรักษาระดับแรงดันไว้แล้ว ดังนั้นจึงสามารถป้อนแรงดันไฟตรง 12 โวลต์ได้ช่อง Adaptor 12 VDC หรือป้อนแรงดันไฟจากแบตเตอรี่ 9 V ได้ที่ช่อง Battery 9 V นอกจากนี้เราสามารถเลือกใช้ไฟจากช่อง USB ได้

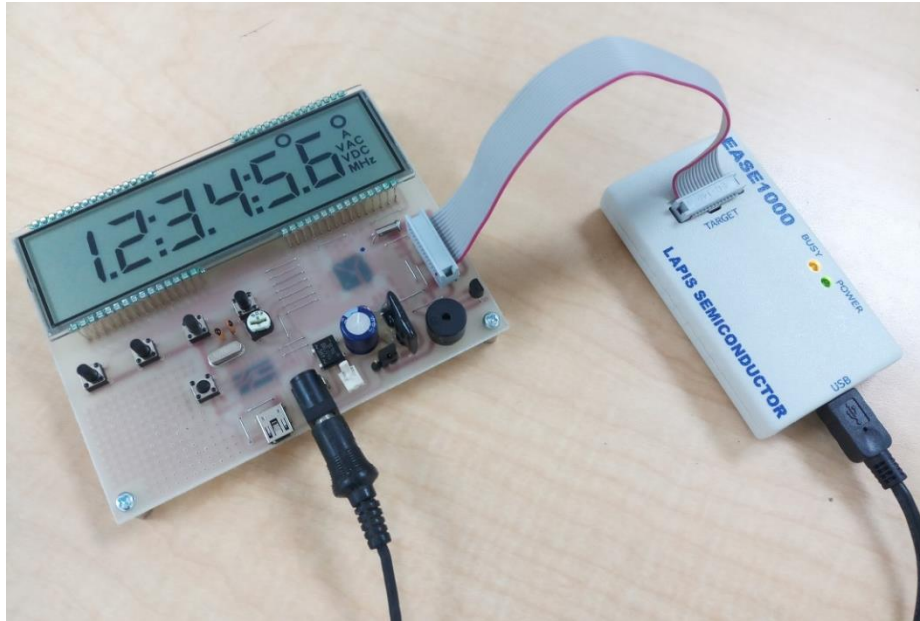


ซึ่งในการเลือกใช้แหล่งจ่ายไฟจาก USB นั้นสามารถทำได้โดย ขยับ Jumper จากภาพด้านล่างถ้าขยับ Jumper ไปทางซ้ายคือเลือกแหล่งจ่ายไฟจาก USB และถ้าขยับ Jumper ไปทางขวาคือใช้แหล่งจ่ายไฟจาก Battery หรือ Adaptor



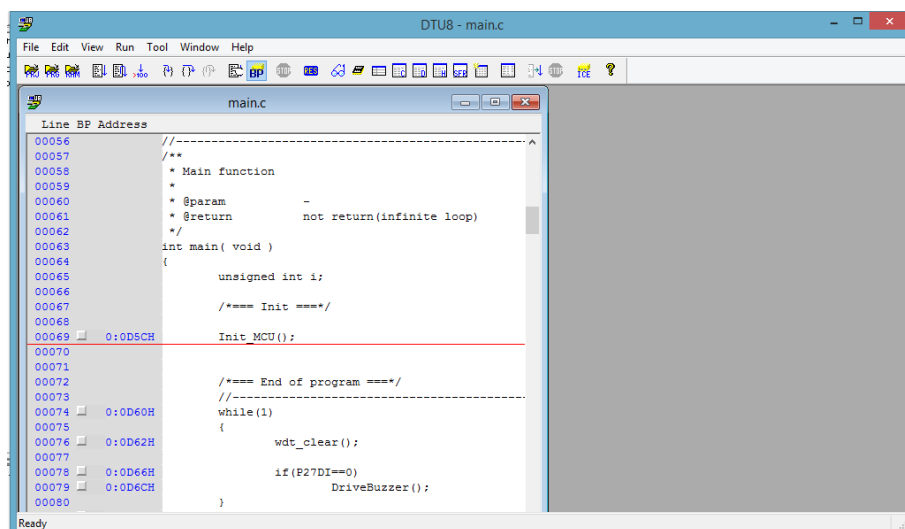
4) การเชื่อมต่อกับคอมพิวเตอร์

ในการเชื่อมต่อระหว่างบอร์ด LAPIS กับคอมพิวเตอร์สามารถทำได้โดยใช้ EASE1000 เป็นตัวกลาง ซึ่ง EASE1000 เป็นอุปกรณ์ประเภท On-Chip Emulator สำหรับไมโครคอนโทรลเลอร์ LAPIS EASE1000 ยังมีฟังก์ชันเขียน หรือเรียกว่า Flash writer ซึ่งจะเขียนลง LSI ด้วย Built-in flash memory ภาพด้านล่างเป็นการเชื่อมต่อระหว่างบอร์ด LAPIS กับคอมพิวเตอร์ โดยใช้ EASE1000 เป็นตัวกลางในการเชื่อมต่อ



รูปที่ 5

โดยในการเชื่อมต่อระหว่างบอร์ด LAPIS กับคอมพิวเตอร์จะมีโปรแกรม DTU8 Debugger ที่ใช้ควบคุมฟังก์ชัน On-Chip Emulator ของ EASE1000 ลักษณะตัวโปรแกรมเป็นดังภาพด้านล่าง



รูปที่ 6

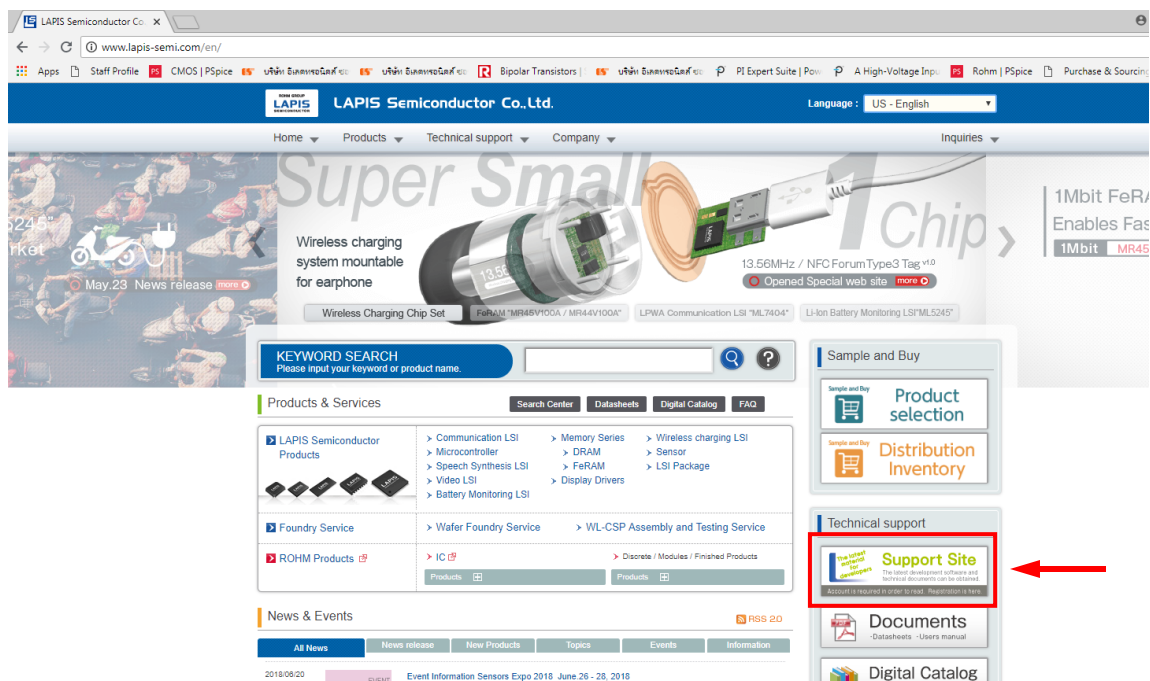
บทที่ 2 Software

ในการทดลองและใช้งานไมโครคอนโทรลเลอร์ LAPIS นั้นจะต้องใช้งานร่วมกับโปรแกรม U8 ซึ่งเป็นโปรแกรมที่ใช้พัฒนาโค้ดโปรแกรมที่จะใช้บนไมโครคอนโทรลเลอร์

1) การติดตั้งโปรแกรม U8

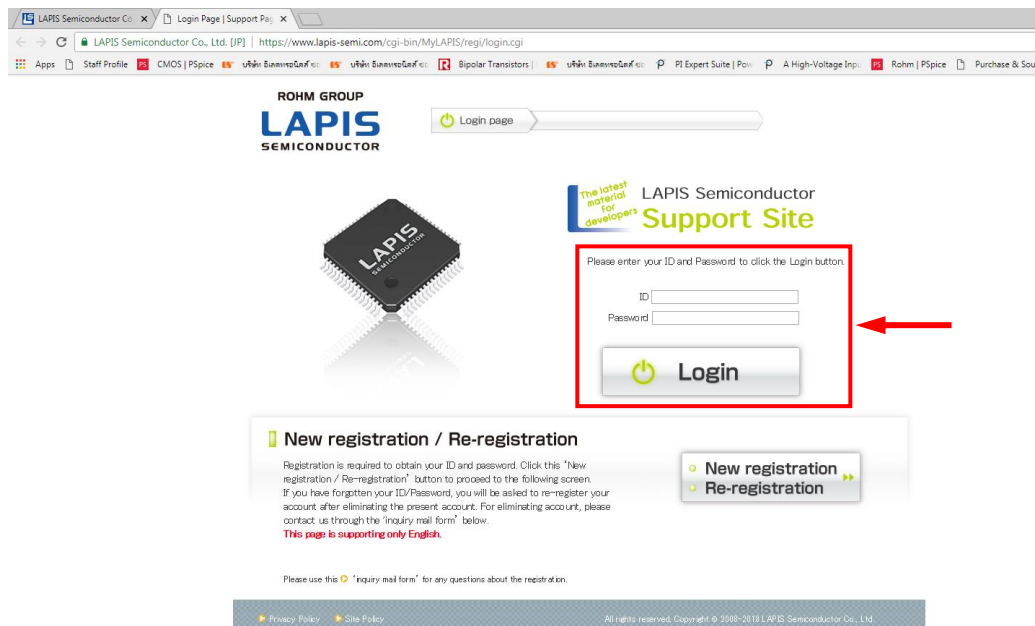
โปรแกรม U8 เป็นโปรแกรมที่ใช้สร้างและพัฒนาโปรเจกต์ที่ใช้กับ MCU LAPIS โดยการติดตั้งโปรแกรม U8 สามารถทำได้ง่าย ดังนี้

ขั้นตอนที่ 1 เข้าไปที่เว็บ <http://www.lapis-semi.com/en/> จากนั้นเลือก Support Site



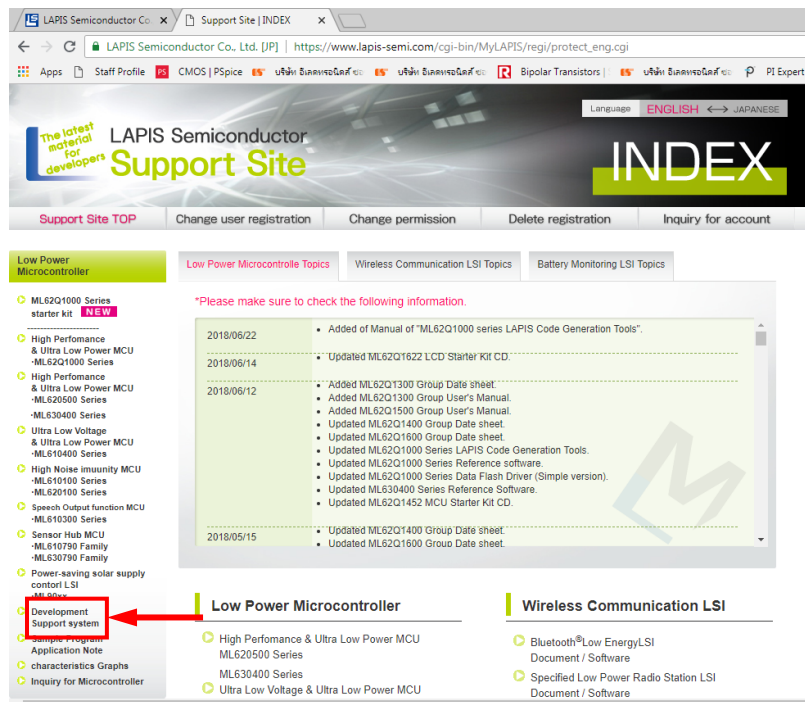
คู่มือการใช้ Microcontroller LAPIS ML62Q1622 เบื้องต้น

ขั้นตอนที่ 2 เมื่อกดแล้วจะปรากฏหน้าต่างขึ้นใหม่ ในหน้าต่างนี้จะมีให้กรอก ID และ Password .ให้สมัครเป็นสมาชิก จากนั้นให้ Login เข้าไป



รูปที่ 8

ขั้นตอนที่ 3 เมื่อ Login เข้าไปจะขึ้นหน้า Support Site ให้กดที่ Development Support System



รูปที่ 9

ขั้นตอนที่ 4 จากนั้นบนหน้า Development Support System จะมีที่ให้โหลดไฟล์มากมาย

The screenshot shows the Lapis Semiconductor Development Support System website. The page is titled "Low Power Microcontroller Development Support system". It features a sidebar with various product categories and a main content area with a table of software packages. A red box highlights the "U8/U16 Development Tools" row, and a red arrow points to the "Download" button for this package.

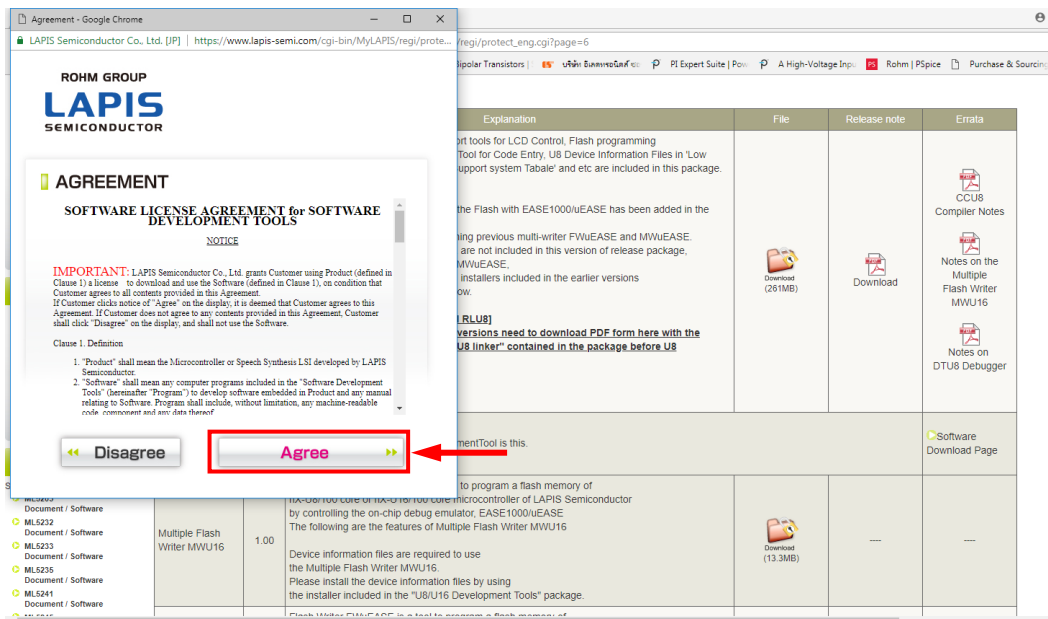
รูปที่ 10

ขั้นตอนที่ 5 เลือก U8/U16 Development Tools กดที่ File Download

The screenshot shows the Lapis Semiconductor Development Support System website, specifically the "U8/U16 Development Tools" row in the software package table. A red box highlights the "U8/U16 Development Tools" row, and a red arrow points to the "Download" button for this package.

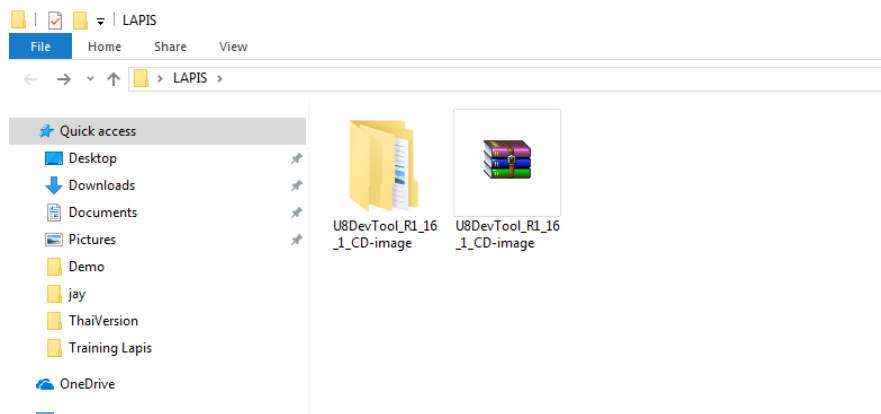
รูปที่ 11

ขั้นตอนที่ 6 เมื่อกดแล้วจะปรากฏหน้าต่างขึ้นใหม่ซึ่งมีเนื้อเกี่ยวกับข้อตกลงการใช้โปรแกรม เมื่ออ่านเสร็จแล้วให้กด Agree เพื่อทำการโหลดไฟล์โปรแกรม



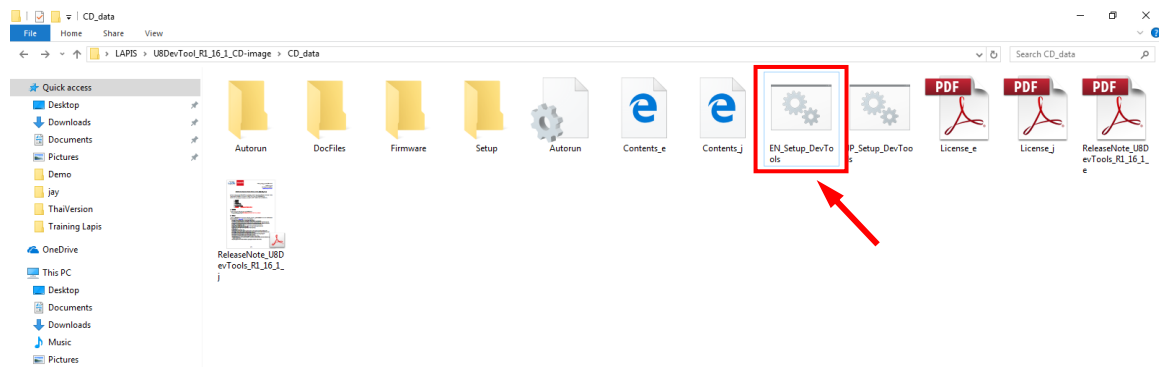
รูปที่ 12

ขั้นตอนที่ 7 พอโหลดเสร็จแล้วจะได้ไฟล์บีบ .rar มา ให้ทำการแตกไฟล์



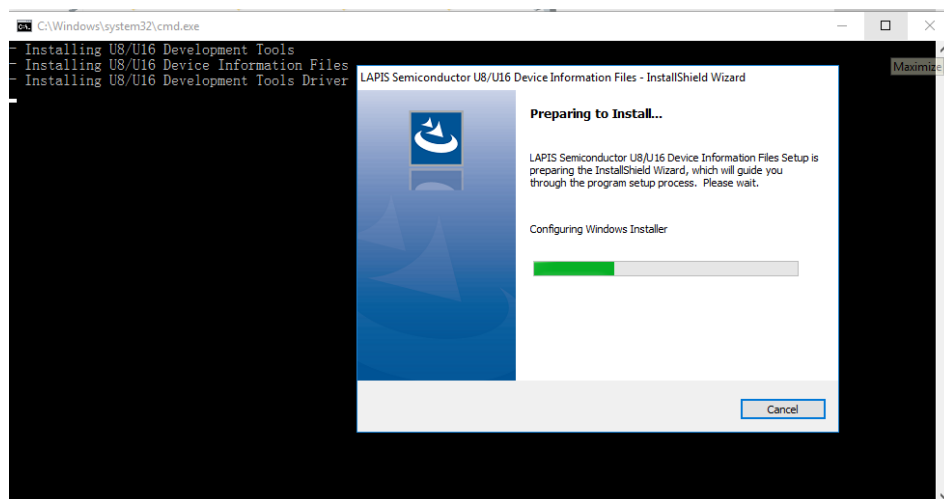
รูปที่ 13

ขั้นตอนที่ 8 เมื่อแตกไฟล์แล้วให้คลิกเข้าไปในโฟลเดอร์ จากนั้นดับเบิลคลิกที่ไฟล์ชื่อ EN_Setup_DevTools.BAT เพื่อทำการติดตั้งโปรแกรม



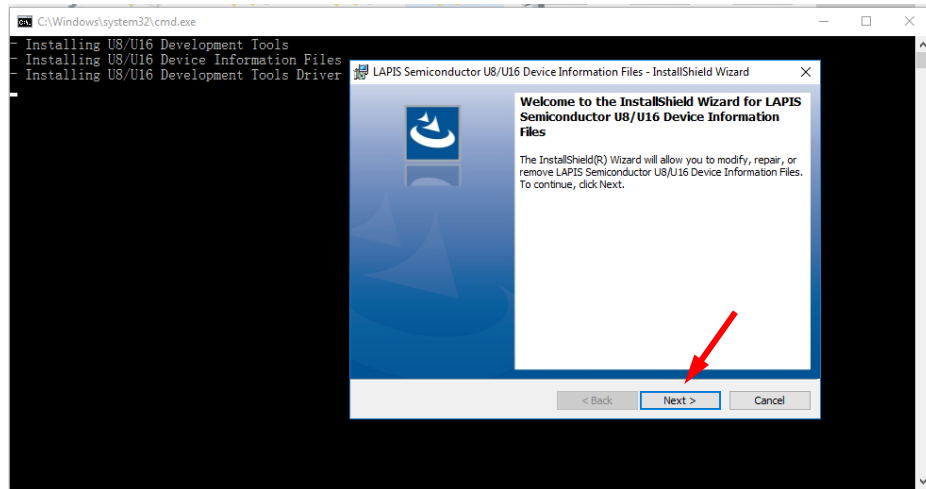
รูปที่ 14

ขั้นตอนที่ 9 เมื่อดับเบิลคลิกไปจะปรากฏหน้าต่างสำหรับลงโปรแกรม รอสักพักให้โปรแกรมเตรียมการติดตั้ง



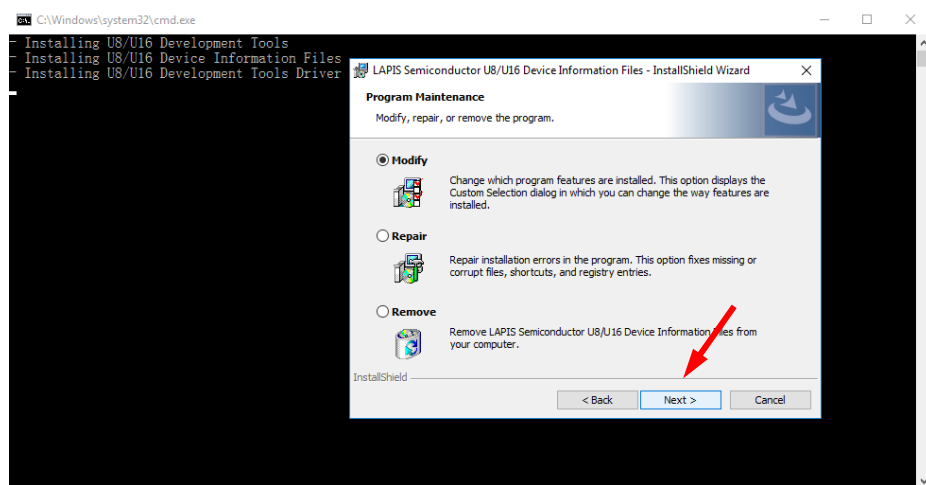
รูปที่ 15

ขั้นตอนที่ 10 โปรแกรมเตรียมการติดตั้งเสร็จแล้ว ให้กดปุ่ม Next



รูปที่ 16

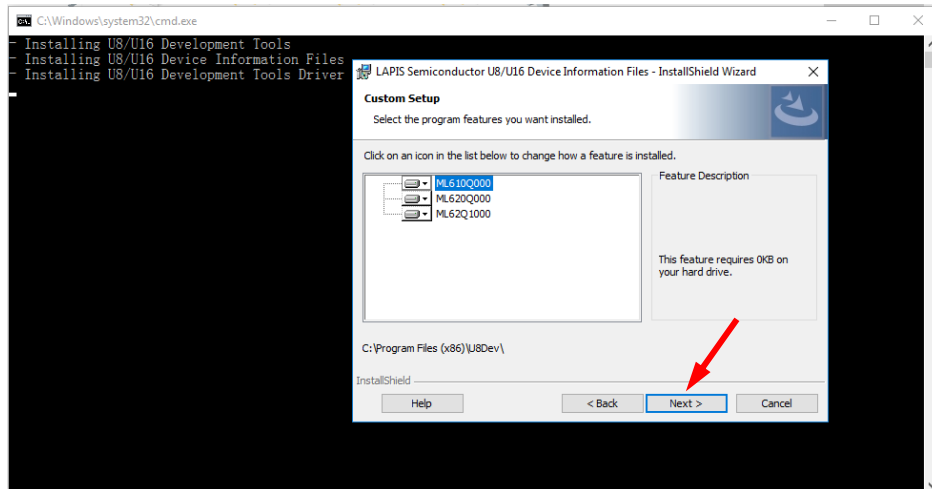
ขั้นตอนที่ 11 จากนั้นเลือก Modify แล้วกดปุ่ม Next



รูปที่ 17

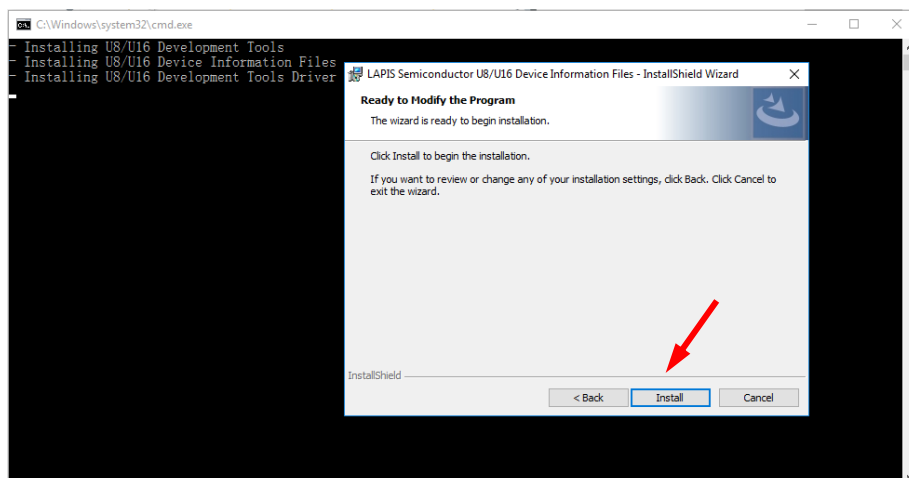
คู่มือการใช้ Microcontroller LAPIS ML62Q1622 เบื้องต้น

ขั้นตอนที่ 12 กดปุ่ม Next อีกครั้ง



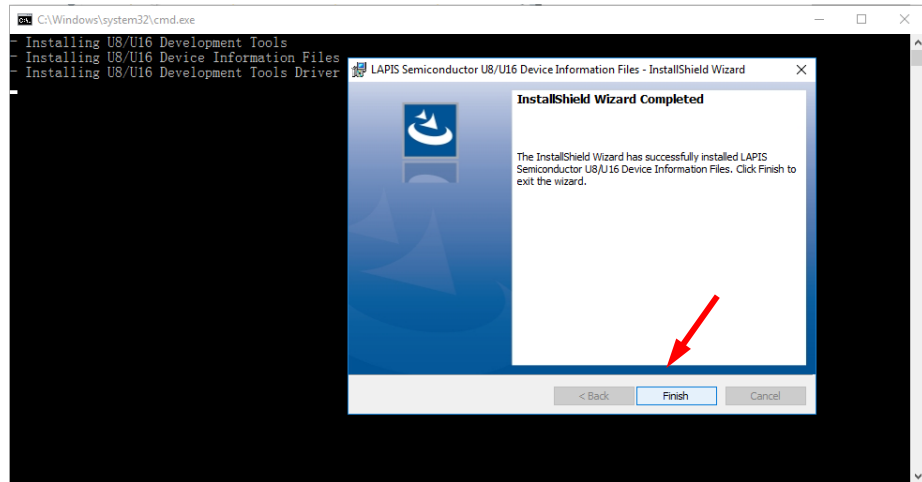
รูปที่ 18

ขั้นตอนที่ 13 กดปุ่ม Install เพื่อติดตั้งโปรแกรม



รูปที่ 19

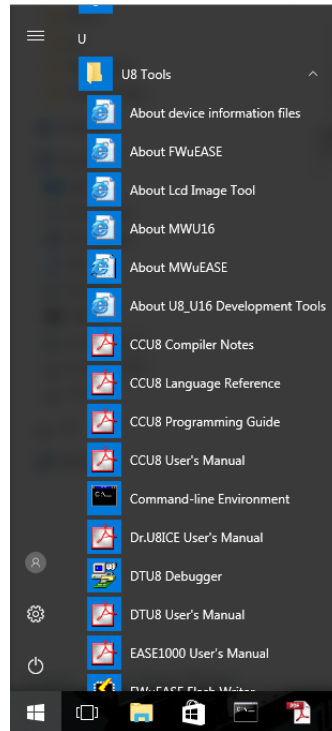
ขั้นตอนที่ 14 เมื่อติดตั้งเสร็จแล้วให้กดปุ่ม Finish



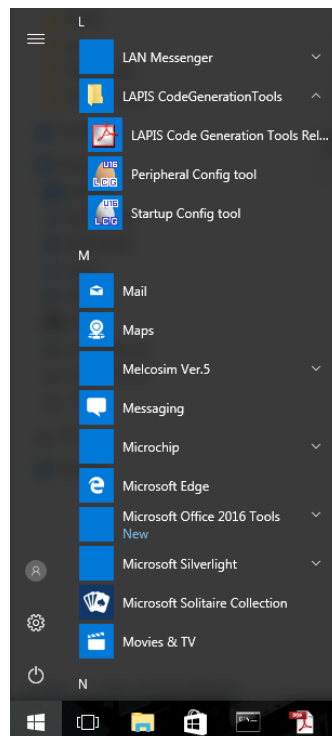
รูปที่ 20

ขั้นตอนที่ 15 เมื่อติดตั้งเสร็จแล้วจะมีโปรแกรมดังนี้

- LAPIS CodeGenerationTools
 - Peripheral Config tool
 - Startup Config tool
- U8 Tools
 - DTU8 Debugger
 - FWuEASE Flash Writer
 - IDEU8
 - LCD Image Assignment Tool
 - MWU16 Multiple Flash Writer
 - MWuEASE Multiple Flash Writer



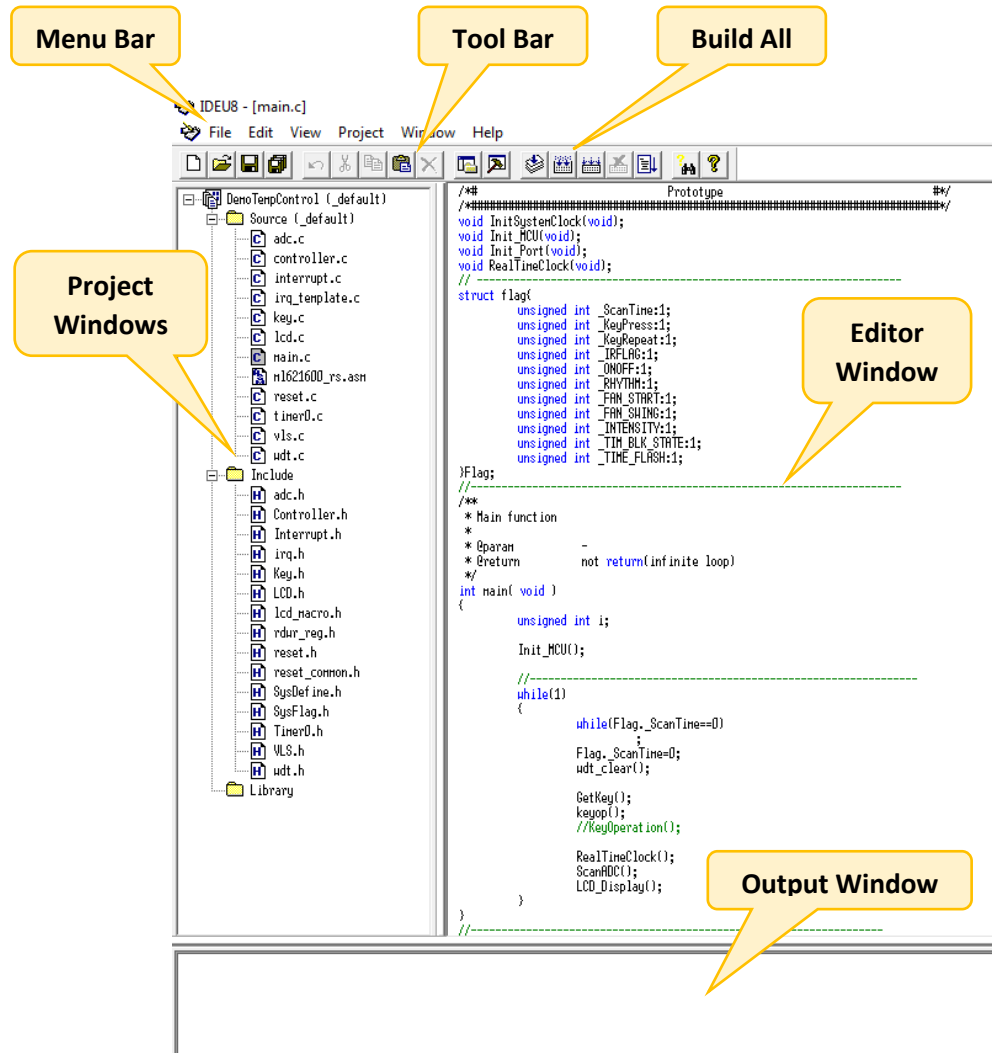
รูปที่ 21



รูปที่ 22

2) เริ่มต้นใช้งานโปรแกรม

2.1) ลักษณะหน้าต่างของโปรแกรม IDEU8



รูปที่ 24

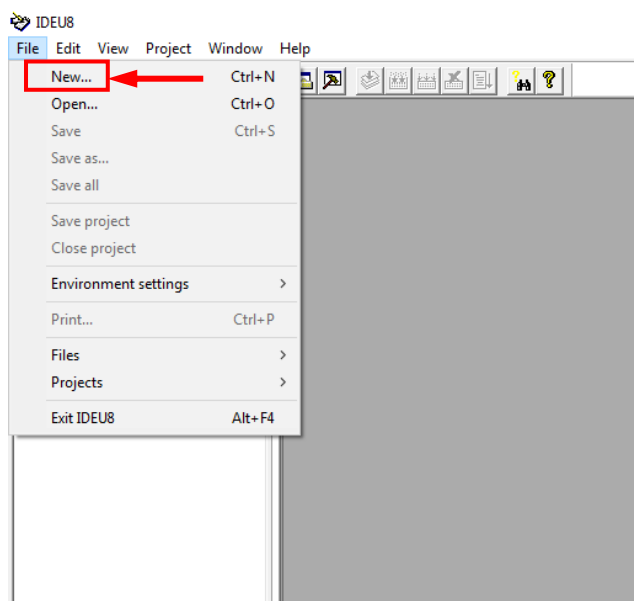
2.2) การเริ่มต้นสร้าง New Project

ขั้นตอนที่ 1 เข้าโปรแกรม IDEU8 เปิดขึ้นมาครั้งแรกจะมีลักษณะ ดังรูปด้านล่าง



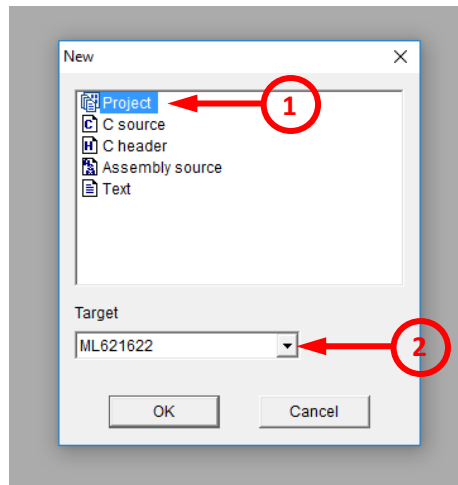
รูปที่ 25

ขั้นตอนที่ 2 กด File จากนั้นกด New...



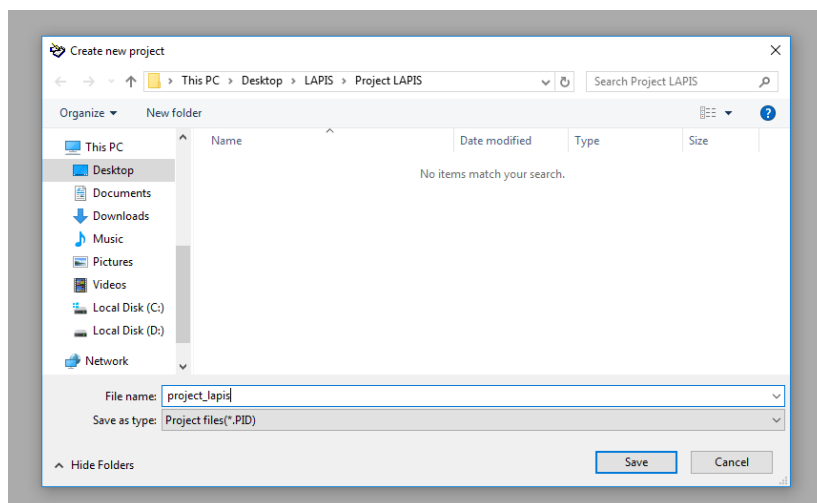
รูปที่ 26

ขั้นตอนที่ 3 เมื่อกดแล้วจะปรากฏหน้าต่างขึ้นมาใหม่ให้เลือก Project จากนั้นเลือกรุ่น Microcontroller ในช่อง Target (ณ ที่นี้ผู้เขียนเลือก ML621622) เสร็จแล้วกดปุ่ม OK



รูปที่ 27

ขั้นตอนที่ 4 จากนั้นทำการบันทึกไฟล์ Project (.PID) เสร็จแล้วกดปุ่ม Save



รูปที่ 28

ขั้นตอนที่ 5 เมื่อเสร็จเรียบร้อยแล้วในหน้าโปรแกรม IDEU8 จะปรากฏไฟล์ Project ขึ้นมาทางด้านซ้ายของโปรแกรม ดังรูปที่แสดงด้านล่าง



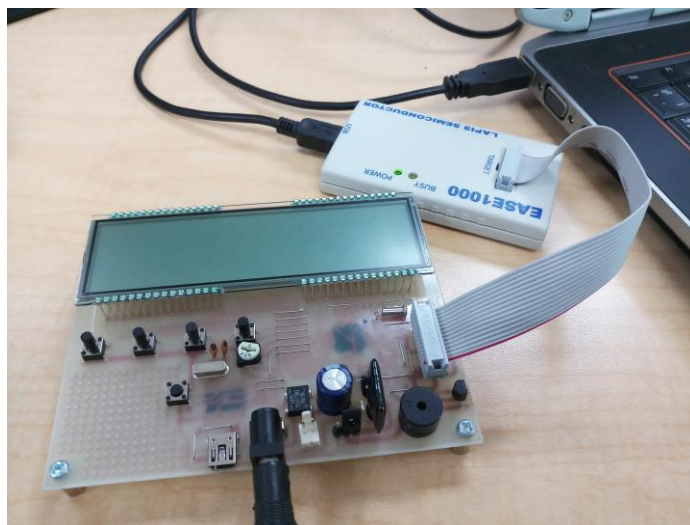
รูปที่ 29

บทที่ 3 Laboratory

1. การทดลองที่ 1 (Basic)

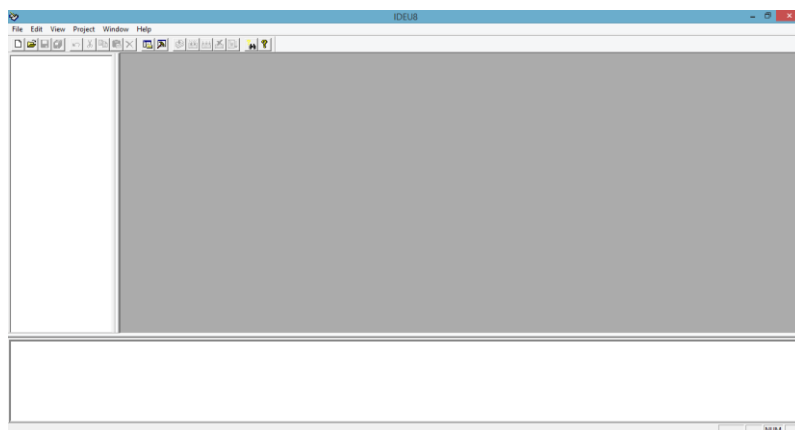
ในการทดลองแรกจะเป็นการทดลองง่ายๆ โดยไปทดลองเกี่ยวกับเสียง ถ้าเรากดคีย์สวิตช์ที่ 1 จะเกิดเสียงขึ้น โดยขั้นตอนมีดังนี้

ขั้นตอนที่ 1 นำสาย EASE1000 ฝั่ง TARGET ไปเสียบบอร์ด LAPIS แล้วนำสาย USB ของ EASE1000 ไปเสียบกับคอมพิวเตอร์ดังภาพ ในที่นี้บอร์ด LAPIS จะใช้แหล่งจ่ายไฟจาก Adaptor ดังนั้นตรง Jumper ต้องขยับไปทาง 9 VDC



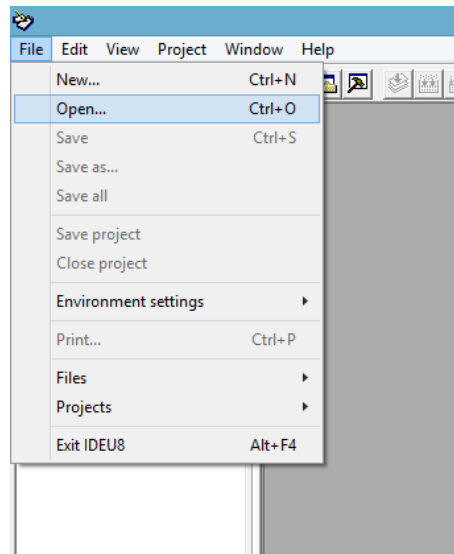
รูปที่ 30

ขั้นตอนที่ 2 เปิดโปรแกรม U8 IDE ขึ้น



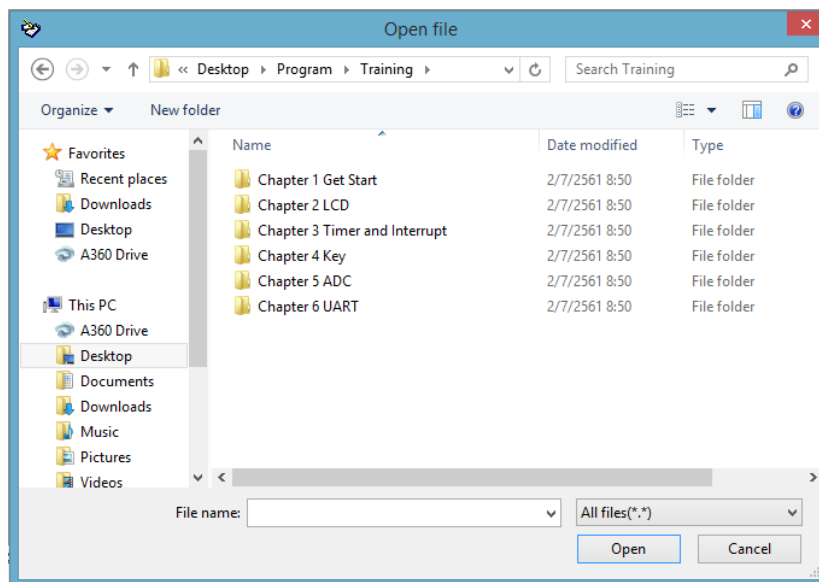
รูปที่ 31

ขั้นตอนที่ 3 จากนั้นให้ทำการเปิดไฟล์ Project ขึ้นมา โดยไปที่ File แล้วเลือก Open...



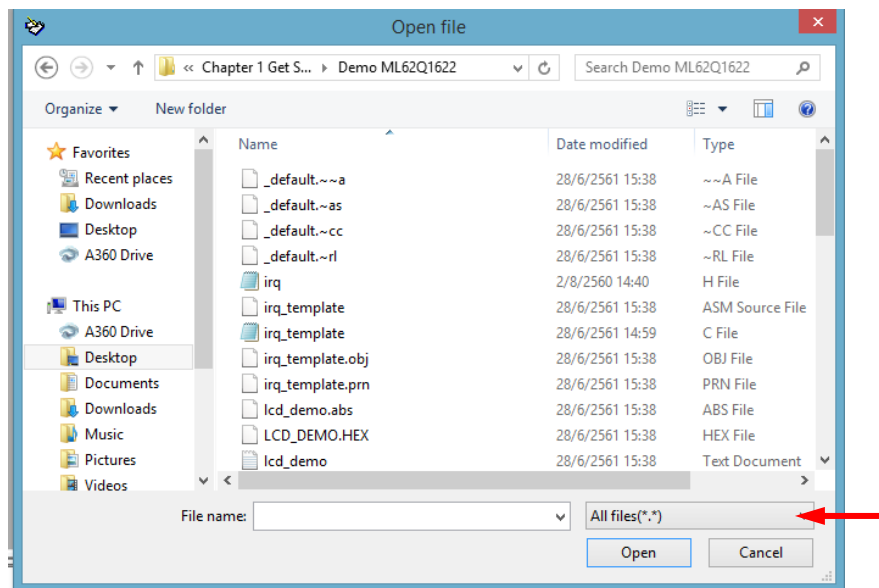
รูปที่ 32

ขั้นตอนที่ 4 จากนั้นเปิดโฟลเดอร์ชื่อ Training แล้วเข้าไปที่ Chapter 1 Get Start >> Demo ML62Q1622

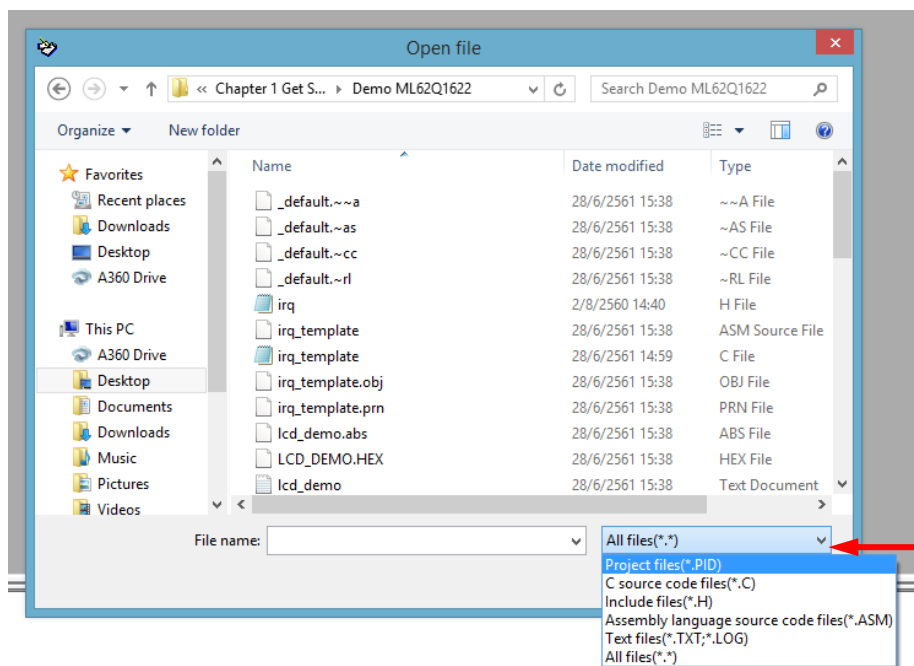


รูปที่ 33

ขั้นตอนที่ 5 เมื่อเข้ามาที่ Demo ML62Q1622 จะเห็นว่ามีไฟล์อยู่มากมาย ให้ไปที่ช่องเลือกประเภทไฟล์ที่ด้านล่างขวาของหน้าต่าง จากนั้นเลือกไฟล์ประเภท Project files(*.PID)

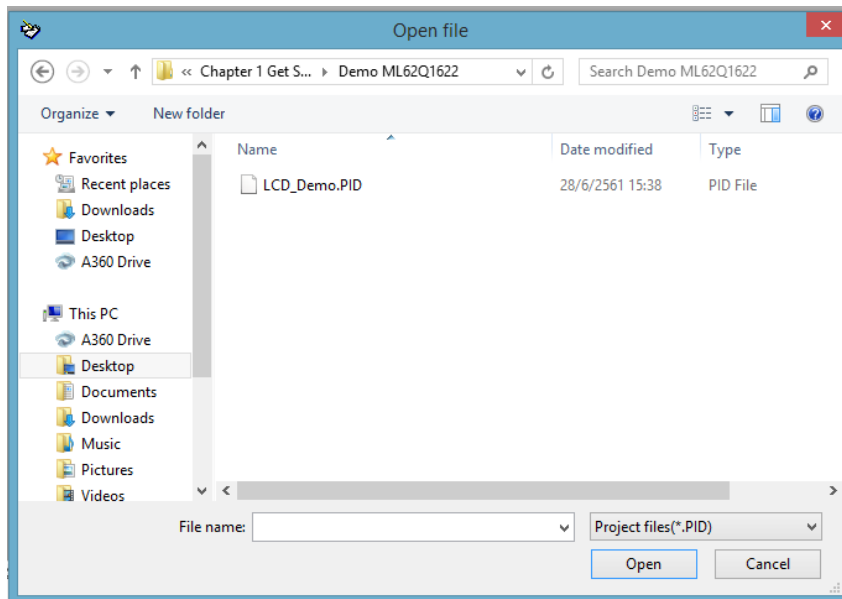


รูปที่ 34



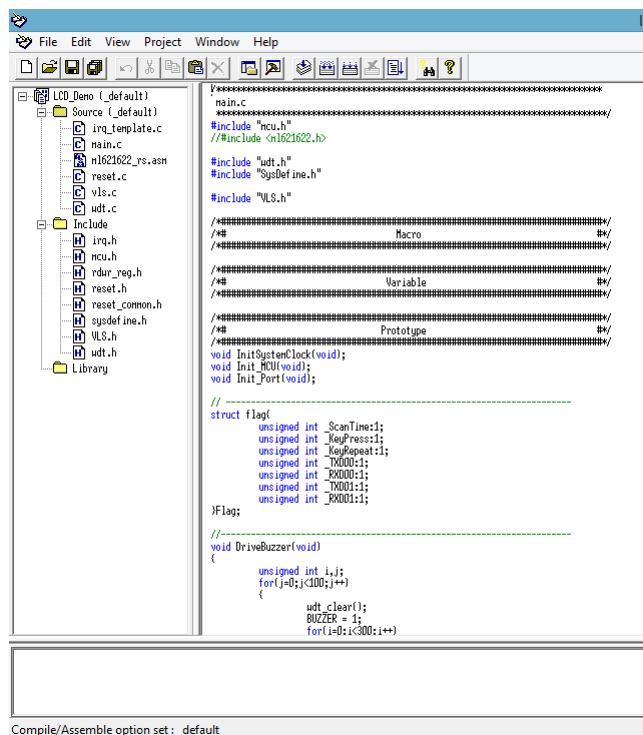
รูปที่ 35

ขั้นตอนที่ 6 เมื่อเลือกไฟล์ประเภท Project files(*.PID) แล้ว โปรแกรมจะตัดไฟล์ชนิดอื่นออกไปให้เหลือแต่ไฟล์ชนิด .PID จากนั้นเลือกไฟล์ชื่อ LCD_Demo.PID



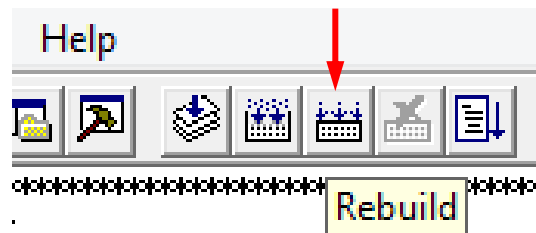
รูปที่ 36

ขั้นตอนที่ 7 เมื่อเปิดไฟล์ Project ขึ้นมาแล้วจะปรากฏโค้ดของโปรแกรม ดังภาพด้านล่าง



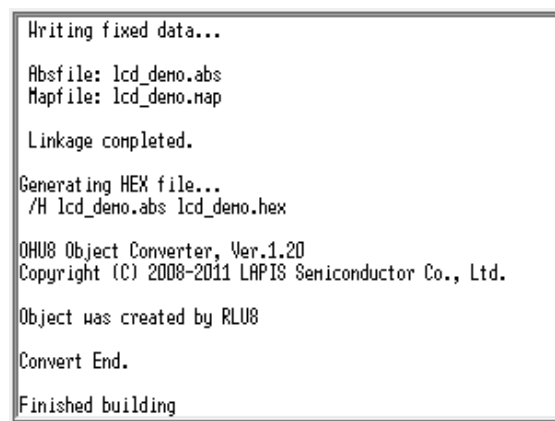
รูปที่ 37

ขั้นตอนที่ 8 จากนั้นกดปุ่ม Rebuild เพื่อ Rebuild โปรแกรมใหม่



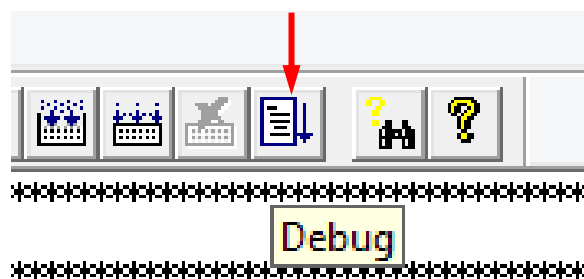
รูปที่ 38

ขั้นตอนที่ 9 เมื่อ Build โปรแกรมเสร็จแล้วจะขึ้นข้อความดังภาพตรงด้านล่างซ้ายของโปรแกรม



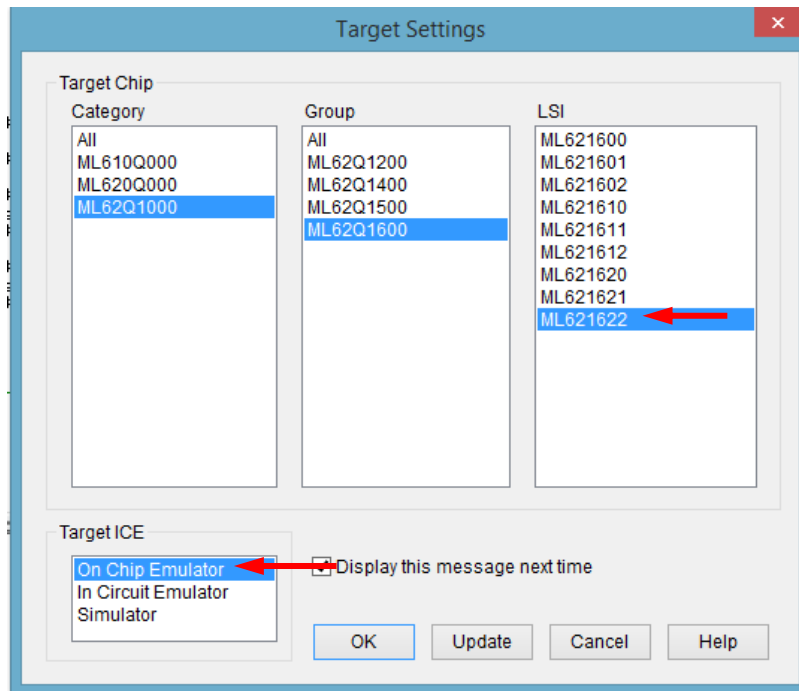
รูปที่ 39

ขั้นตอนที่ 10 ทำการ Debug โปรแกรมโดยกดปุ่ม Debug



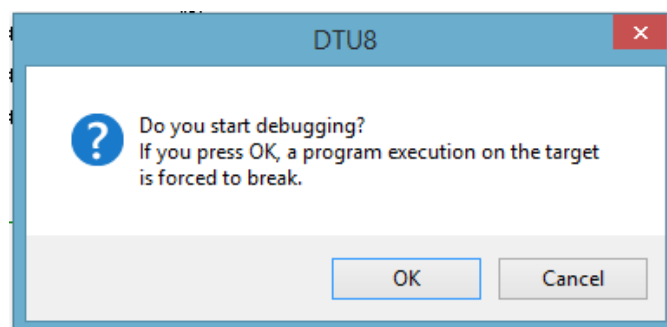
รูปที่ 40

ขั้นตอนที่ 11 เมื่อกดแล้วจะปรากฏหน้าต่างขึ้นใหม่ ซึ่งเป็นหน้าต่าง Target Settings เป็นหน้าต่างสำหรับการตั้งค่า รูนไมโครคอนโทรลเลอร์ LAPIS ที่ใช้งาน โดยบอร์ด LAPIS ที่ใช้ ณ ที่นี้จะใช้ไมโครคอนโทรลเลอร์รุ่น ML62Q1622 เมื่อรูนไมโครคอนโทรลเลอร์เรียบร้อยแล้ว ให้ไปที่ Target ICE เลือก On Chip Emulator เพราะตอนนี้เชื่อมต่อบอร์ด โดยใช้ EASE1000 จากนั้นกดปุ่ม OK



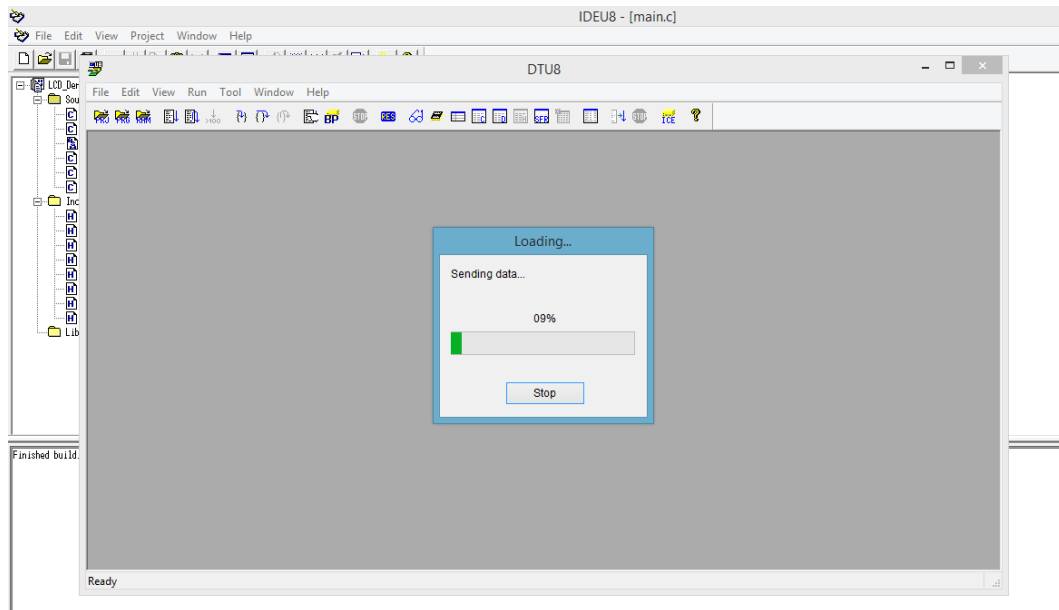
รูปที่ 41

ขั้นตอนที่ 12 โปรแกรมจะขึ้นถามว่า “คุณต้องการเริ่มการ Debug หรือไม่” ให้กดปุ่ม OK



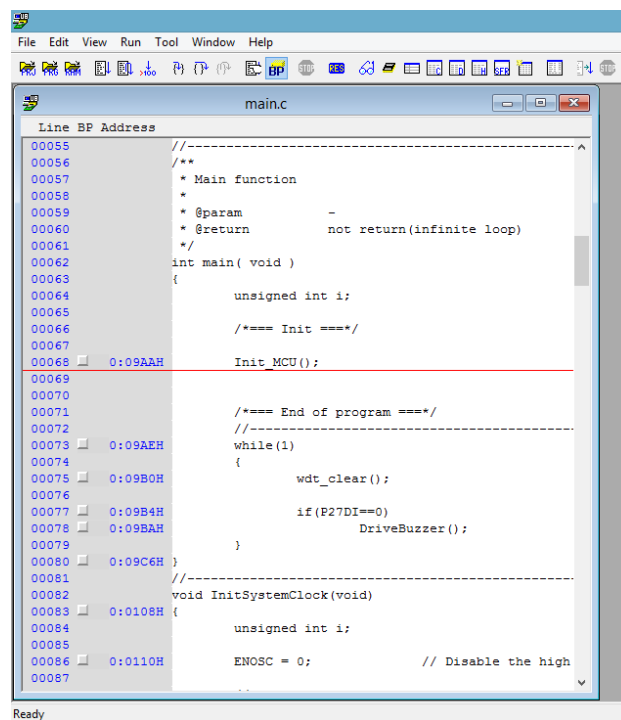
รูปที่ 42

ขั้นตอนที่ 13 จากนั้นจะขึ้นโปรแกรม DTU8 ซึ่งเป็นโปรแกรมที่ใช้ Debug ไมโครคอนโทรลเลอร์

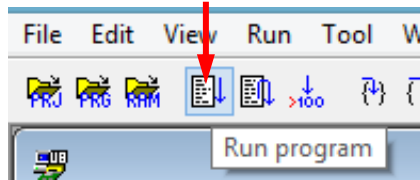


รูปที่ 43

ขั้นตอนที่ 14 ในโปรแกรม DTU8 จะมีหน้าต่างแสดงโค้ดที่กำลังประมวลผลอยู่ ให้กดปุ่ม Run program ดังภาพ เพื่อทำการเขียนโปรแกรมลงไมโครคอนโทรลเลอร์และทำการ Run โปรแกรม



รูปที่ 44

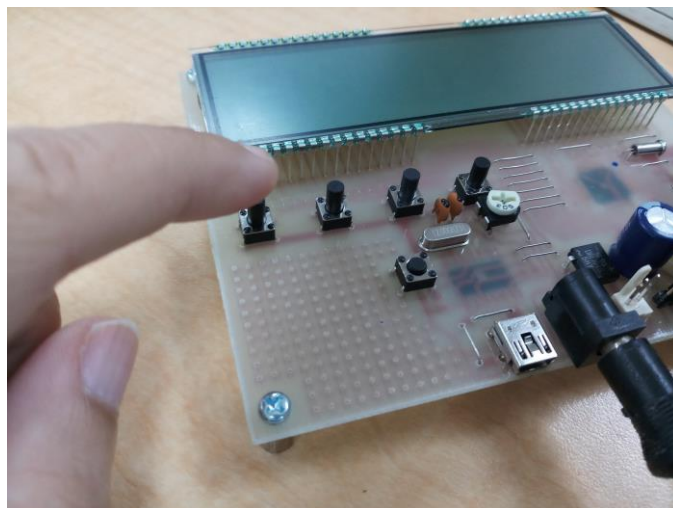


รูปที่ 45

ขั้นตอนที่ 15 เมื่อกดปุ่ม Run program ไฟสีส้ม (BUSY) ที่ EASE1000 จะติด จากนั้นให้ลองกดคีย์สวิตช์แรกจะเห็นว่า มีเสียงเกิดขึ้นที่ Buzzer



รูปที่ 46



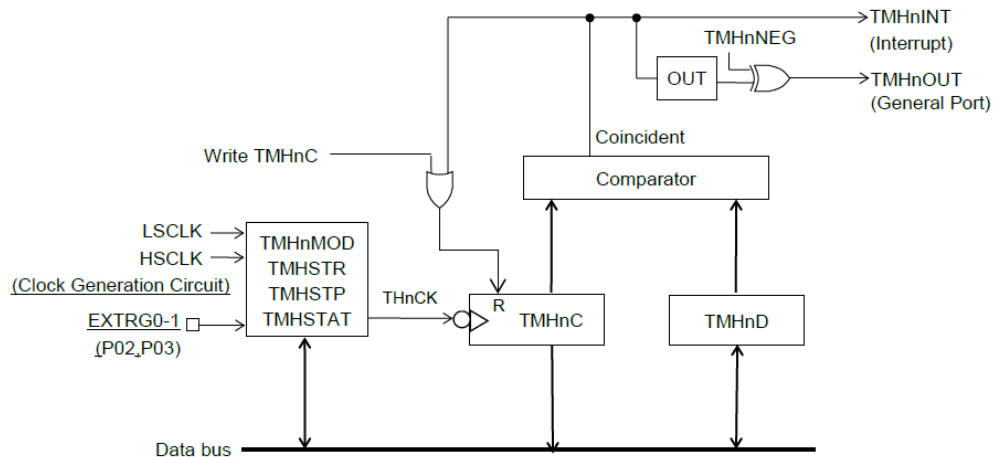
รูปที่ 47

2. การทดลองที่ 2 แสดงผลบนจอ LCD

บทที่ 4 ข้อมูล Register

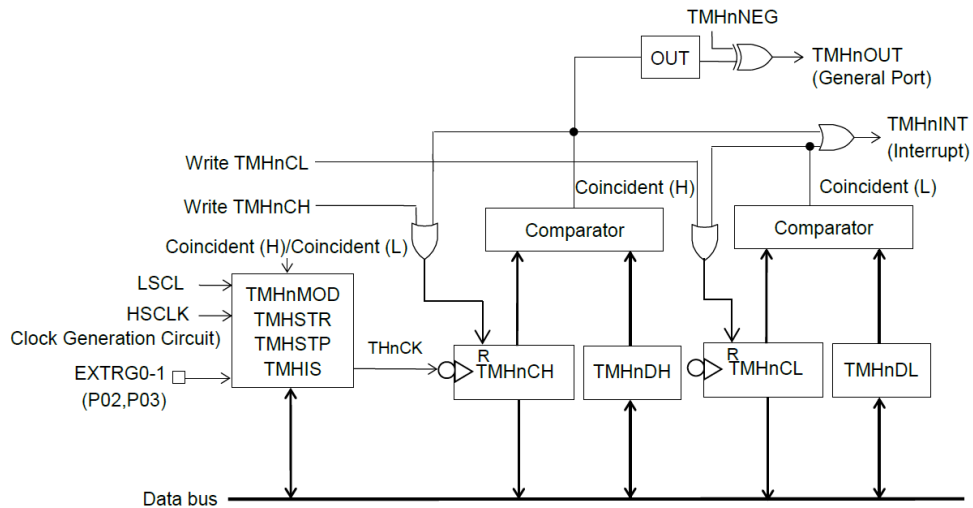
16-Bit Timer

ML62Q1000 มี 16 – Bit Timer ซึ่งการนับ Clock จะอ้างอิงกับ LSCLK (Low speed clock) หรือ HSCLK (High speed clock)



รูปที่

- TMHnMOD : 16-Bit timer mode register
- TMHSTR : 16-Bit timer start register
- TMHSTP : 16-Bit timer stop register
- TMHSTAT : 16-Bit timer status register
- TMHnD : 16-Bit timer n data register
- TMHnC : 16-Bit timer n counter register



รูปที่

- TMHnMOD : 16-Bit timer mode register
- TMHSTR : 16-Bit timer start register
- TMHSTP : 16-Bit timer stop register
- TMHIS : 16-Bit timer interrupt status register
- TMHnDL : 16-Bit timer n data register (lower side)
- TMHnDH : 16-Bit timer n data register (upper side)
- TMHnCL : 16-Bit timer n counter register (lower side)
- TMHnCH : 16-Bit timer n counter register (upper side)

1) 16-Bit Timer n Data Register (TMHnD : n = 0 ถึง 5)

Address : 0xF300, 0xF302, 0xF304, 0xF306, 0xF308, 0xF30A

Access : R/W

Access size : 8/16 bits

Initial value : 0xFFFF

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	TMHnD															
Byte symbol	TMHnDH								TMHnDL							
Bit symbol	THnD15	THnD14	THnD13	THnD12	THnD11	THnD10	THnD9	THnD8	THnD7	THnD6	THnD5	THnD4	THnD3	THnD2	THnD1	THnD0
Access type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

รูปที่

TMHnD (n = 0 ถึง 5) คือ specific function register (SFR) ที่ใช้ตั้งค่าเพื่อเปรียบเทียบกับ 16-bit timer n counter register (TMHnC)

ในโหมด 8-bit timer TMHnDL (n = 0 ถึง 5) จะถูกเปรียบเทียบกับ TMHnCL (n = 0 ถึง 5) และ TMHnDH (n = 0 ถึง 5) จะถูกเปรียบเทียบกับ TMHnCH (n = 0 ถึง 5)

Note: TMHnD จะตั้งค่าก็ต่อเมื่อ 16-Bit timer ถูกหยุด (THnSTAT/THnHSTAT bit ของ TMHSTAT register เป็น 0) เมื่อ

2) 16-Bit timer n counter register (TMHnC n = 0 ถึง 5)

Address : 0xF310, 0xF312, 0xF314, 0xF316, 0xF318, 0xF31A

Access : R/W

Access size : 8/16 bits

Initial value : 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	TMHnC															
Byte symbol	TMHnCH								TMHnCL							
Bit symbol	THnC15	THnC14	THnC13	THnC12	THnC11	THnC10	THnC9	THnC8	THnC7	THnC6	THnC5	THnC4	THnC3	THnC2	THnC1	THnC0
Access type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

รูปที่

TMHnC (n = 0 ถึง 5) คือ specific function register (SFR) เป็น Function ที่เป็น 16-Bit binary counter

3) 16-Bit Timer n Mode Register (TMHnMOD n = 0 ถึง 5)

Address : 0xF320, 0xF322, 0xF324, 0xF326, 0xF328, 0xF32A

Access : R/W

Access size : 8/16 bits

Initial value : 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	TMHnMOD															
Byte symbol	TMHnMODH								TMHnMODL							
Bit symbol	THnNEG	THnOST	THn8BM	.	THnDIV2	THnDIV1	THnDIV0	THnEXS	THnEX	.	THnCS
Access type	R	R	R	R	R	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	R	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

รูปที่

TMHnMOD (n = 0 ถึง 5) คือ Specific Function Register (SFR) ใช้ควบคุม 16-bit timer 8 และจะเขียน TMHnMOD อีกครั้งเมื่อ 16-bit timer 8 ถูกหยุด

รายละเอียดแต่ละบิต

THnCS (Bit 0)

THnCS คือ บิตที่ใช้เลือก Timer Clock ของ 16- Bit timer n timer

THnCS	รายละเอียด
0	LSCLK (ค่าเริ่มต้น)
1	HSCLK

THnEX (Bit 2)

THnEX คือ บิตที่ใช้เลือก Count Clock ของ 16-Bit timer n timer

Note : ความแตกต่างระหว่าง Timer Clock และ Count Clock คือ Count Clock จะถูกใช้ในงาน Counting Operation และ Waveform Output Control แต่ Timer Clock จะถูกใช้ในงาน Sampling และตรวจเช็คขอบของสัญญาณ Trigger ภายนอก

THnEXS	รายละเอียด
0	เลือกใช้ Timer Clock ในบิต THnCS โดยสามารถกำหนดอัตราส่วน Dividing Ratio ได้ด้วยการกำหนดบิตที่ 0 ถึง 2 ของ THnDIV
1	เลือกใช้ขอบขาขึ้นของสัญญาณ Trigger ภายนอก ในบิต THnEXS ที่ตรวจจับขอบของสัญญาณด้วยด้วย Timer Clock ใน THnCS

THnEXS (Bit 3)

THnEXS คือ บิตที่ใช้เลือก Trigger ภายนอก

THnEXS	รายละเอียด
0	P02 (ค่าเริ่มต้น)
1	P03

THnDIV2 ถึง THnDIV0 (Bit 4 ถึง 6)

THnDIV2 ถึง THnDIV0 ใช้เลือกอัตราส่วน Dividing Ratio ของ 16-Timer n timer

THnDIV2	THnDIV1	THnDIV0	รายละเอียด
0	0	0	หาร 1 (ค่าเริ่มต้น)
0	0	1	หาร 1/2
0	1	0	หาร 1/4
0	1	1	หาร 1/8
1	0	0	หาร 1/16
1	0	1	หาร 1/32
1	1	0	หาร 1/64
1	1	1	หาร 1/128

THn8BM (Bit 8)

THn8BM คือ บิตที่ใช้เลือก Timer ระหว่าง 16-Bit Timer กับ 8-Bit Timer

THn8BM	รายละเอียด
0	16-Bit Timer Mode (ค่าเริ่มต้น)
1	8-Bit Timer Mode

THnOST (Bit 9)

THnOST คือ บิตที่ใช้เลือกโหมดการทำงานของ 16-Bit Timer n

THnOST	รายละเอียด
0	Repeat Timer Mode (ค่าเริ่มต้น)
1	One-shot Timer Mode

THnNEG (Bit 10)

THnNEG คือ บิตที่ใช้เลือก Output Polarity ของ timer out (TMHnOUT)

THnNEG	รายละเอียด
0	Logic บวก (ค่าเริ่มต้น)
1	Logic ลบ

Note : ตั้งค่า TMHnMOD เมื่อ Timer n หยุด (บิต THnSTAT/THnHSTAT ของ TMHSTAT register เป็น “0”)

4) 16-Bit Timer Start Register (TMHSTR)

Address : 0xF350

Access : W

Access size : 8/16 bits

Initial value : 0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	TMHSTR															
Byte symbol	TMHSTRH								TMHSTRL							
Bit symbol	.	.	TH5HRUN	TH4HRUN	TH3HRUN	TH2HRUN	TH1HRUN	TH0HRUN	.	.	TH5RUN	TH4RUN	TH3RUN	TH2RUN	TH1RUN	TH0RUN
Access type	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

รูปที่

TMHSTRL คือ specific function register (SFR) ที่ใช้ควบคุมการเริ่มต้นการนับของ 16-Bit timer n (0 ถึง 5)

TMHSTRH คือ specific function register (SFR) ที่ใช้ในโหมด 8 Bit timer

TMHSTR คือ write – only register ใช้ควบคุมขบวนการเริ่มต้นของ timer และถูกหยุดการทำงานโดย TMHSTP

ค่าที่กำหนด	รายละเอียด
0	คงสถานะปัจจุบัน (ค่าเริ่มต้น)
1	เริ่มนับ

Bit	ชื่อ Bit	
Bit 13	TH5HRUN	ในโหมด 8-bit timer Upper side timer start control ของ 16-bit timer 5
Bit 12	TH4HRUN	ในโหมด 8-bit timer Upper side timer start control ของ 16-bit timer 4
Bit 11	TH3HRUN	ในโหมด 8-bit timer Upper side timer start control ของ 16-bit timer 3
Bit 10	TH2HRUN	ในโหมด 8-bit timer Upper side timer start control ของ 16-bit timer 2
Bit 9	TH1HRUN	ในโหมด 8-bit timer Upper side timer start control ของ 16-bit

		timer 1
Bit 8	TH0HRUN	ในโหมด 8-bit timer Upper side timer start control ของ 16-bit timer 0
Bit 5	TH5RUN	ในโหมด 16-bit timer Timer start control ของ 16-bit timer 5 ในโหมด 8-bit timer Lower side timer start control ของ 16-bit timer 5
Bit 4	TH4RUN	ในโหมด 16-bit timer Timer start control ของ 16-bit timer 4 ในโหมด 8-bit timer Lower side timer start control ของ 16-bit timer 4
Bit 3	TH3RUN	ในโหมด 16-bit timer Timer start control ของ 16-bit timer 3 ในโหมด 8-bit timer Lower side timer start control ของ 16-bit timer 3
Bit 2	TH2RUN	ในโหมด 16-bit timer Timer start control ของ 16-bit timer 2 ในโหมด 8-bit timer Lower side timer start control ของ 16-bit timer 2
Bit 1	TH1RUN	ในโหมด 16-bit timer Timer start control ของ 16-bit timer 1 ในโหมด 8-bit timer Lower side timer start control ของ 16-bit timer 1
Bit 0	TH50RUN	ในโหมด 16-bit timer Timer start control ของ 16-bit timer 0 ในโหมด 8-bit timer Lower side timer start control ของ 16-bit timer 0

Note : ในโหมด 16-Bit timer บิตที่ 13 ถึง 8 (บิต THnHRUN (n = 0 ถึง 5)) จะไม่ถูกใช้งาน และตั้งค่า TMHSTR เมื่อ timer n หยุดทำงาน (บิต THnSTAT/THnHSTAT ของ TMHSTAT register เป็น “0”)

การทำงานของ 16-Bit Timer เบื้องต้น (Basic Operation of 16-Bit timer)

ใน 16-bit timer n counter (TMHnC n 0 ถึง 5) เมื่อเขียนค่า “1” ลงบิต THnRUN ของ TMHSTR register THnSTAT จะมีสถานะเป็น “1” ณ จุดขอบขาขึ้นของสัญญาณ Timer n clock (THnCK) ที่ถูกเลือกใน 16-Bit timer n mode register (TMHnMOD) และเริ่มนับ ณ จุดขอบขาขึ้นของลูกคลื่นลูกที่ 2 ของสัญญาณ timer n clock (THnCK) ในระหว่างการนับค่า Count value ของ TMHnC จะเกิดขึ้นพร้อมกับค่าที่อยู่ใน Timer n data register (TMHnD) สัญญาณของ Timer n interrupt (TMnINT) จะเกิดขึ้น ณ จุดขอบขาขึ้นของสัญญาณ THnCK และ TMHnC ถูก Reset เป็น “0x0000” ณ ตอนนั้น การนับจะมีความต่อเนื่องเมื่อ Timer อยู่ในโหมด Repeat Timer Mode และหยุดการนับเมื่อ timer อยู่ในโหมด one-shot timer mode

เมื่อบิต THnSTP ของ 16-Bit timer stop register (TMHSTP) ถูกตั้งค่าเป็น “1” TMHnC จะหยุด ณ ขอบขาขึ้นของลูกคลื่นลูกแรกของสัญญาณ THnCK และ THnSTAT เป็น “0” เมื่อบิต THnRUN ถูกตั้งค่าเป็น “0” อีกครั้ง TMHnC จะ Restart การนับจากค่าเวลาที่หยุด TMHnC จะกลับไปค่าเริ่มต้น “0x0000”

คาบของ Timer Interrupt มีสมการดังนี้

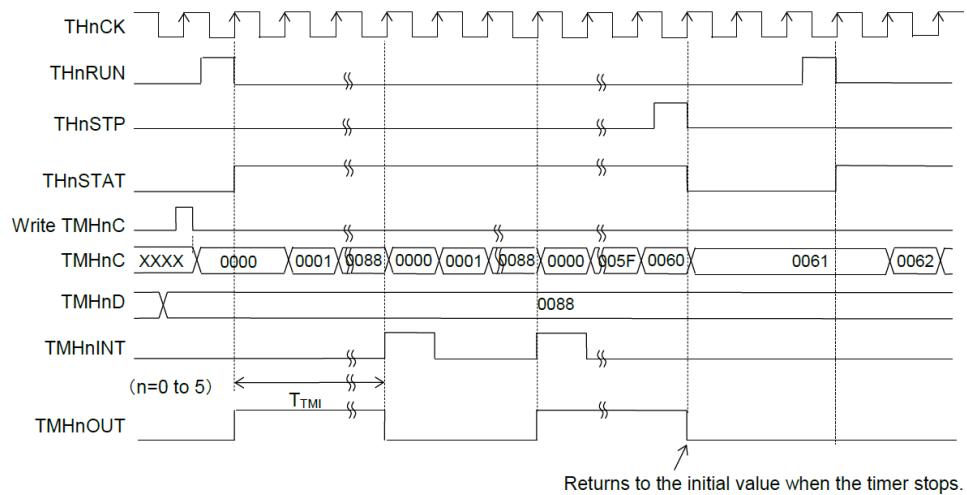
$$T_{TMI} = \frac{TMHnD+1}{THnCK (Hz)} \quad (n=0 \text{ to } 7)$$

TMHnD : 16-Bit timer n data register (TMHnD) ตั้งค่าในช่วง “0x0001” ถึง “0xFFFF”

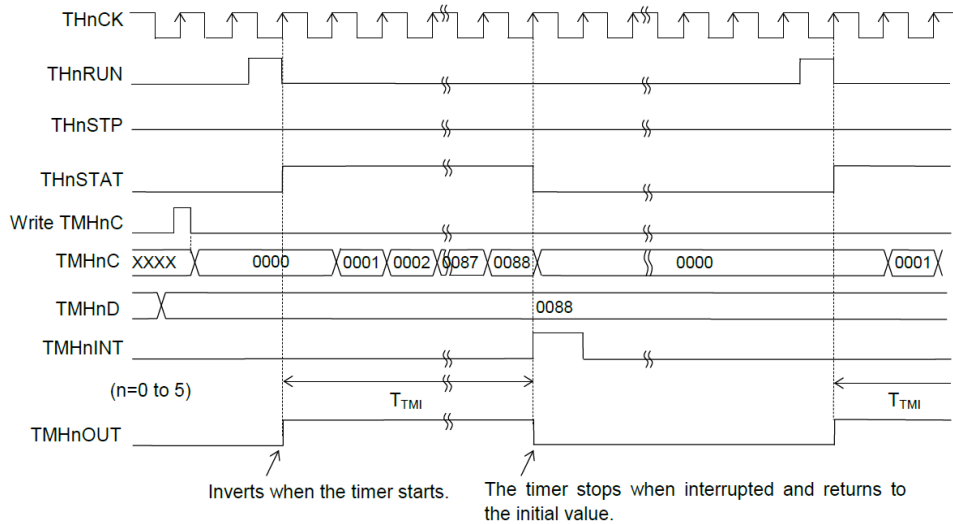
THnCK : Clock Frequency ตั้งค่าโดย 16-bit timer n mode register (TMHnMOD)

เมื่อบิต THnRUN ถูกตั้งค่าเป็น “1” จะเริ่มต้นการนับ

จากภาพด้านล่างแสดงถึงคลื่นสัญญาณในโหมด Repeat timer และในโหมด one-shot timer ในภาพถัดไป



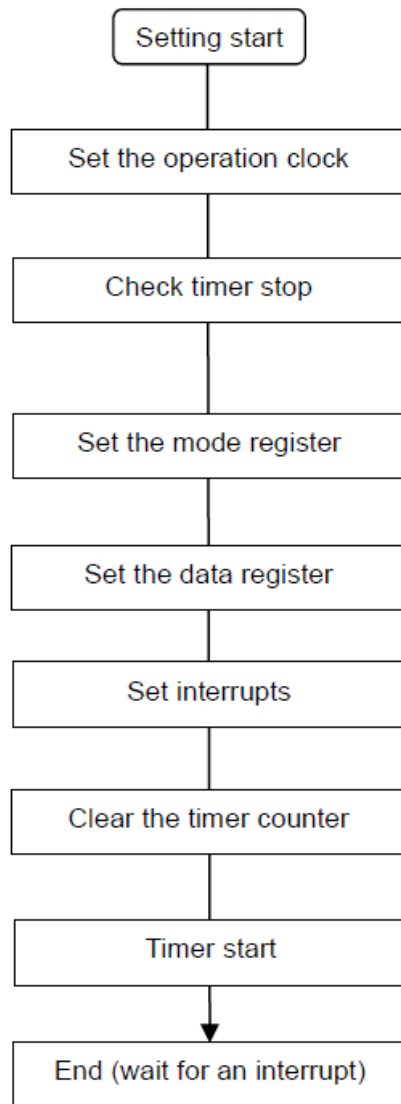
รูปที่



รูปที่

Example (Repeat Timer Mode)

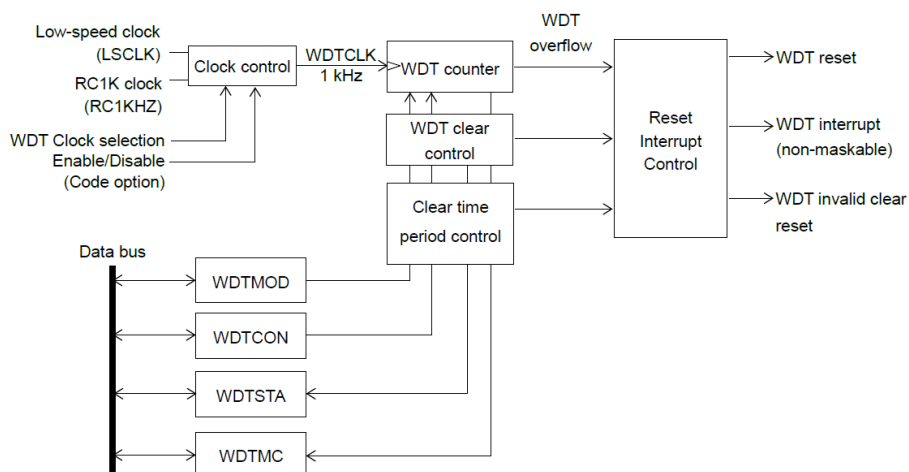
จากรูป แสดงถึงการตั้งค่าเพื่อสร้าง interrupt ทุกๆ 100 mS ในโหมด repeat timer โดยใช้ 16-bit timer 0



Watchdog Timer

ML62Q1000 มี watchdog timer (WDT) ที่ใช้ตรวจจับสถานะที่ไม่ได้กำหนดของ CPU และสร้าง non-maskable interrupt หรือระบบจะ Reset เพื่อกลับไปสู่สถานะปกติจากสถานะที่ไม่ได้กำหนด การเปิด - ปิดการใช้งาน WDT หรือ WDT clock สามารถทำได้ด้วย code

Note : Non-maskable interrupt คือ interrupt ที่ CPU ไม่สามารถปฏิเสธได้



รูปที่

WDTCON : Watchdog timer control register

WDTMOD : Watchdog timer mode register

WDTMC : Watchdog timer counter register

WDTSTA : Watchdog timer status register

1) Watchdog Timer Control Register (WDTCON)

Address : 0xF010

Access : R/W

Access size : 8 bits

Initial value : 0x00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	-															
Byte symbol	-								WDTCON							
Bit symbol	d7	d6	d5	d4	d3	d2	d1	WDP/d0
Access type	R	R	R	R	R	R	R	R	W	W	W	W	W	W	W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

รูปที่

WDTCON คือ specific function register (SFR) ที่ใช้ Clear ค่า WDT Counter

เมื่อ WDTCON ถูกอ่านค่า Internal pointer (WDP) จะเปลี่ยนเป็นอ่านในบิต 0

รายละเอียดแต่ละบิต

WDP (Bit 0)

WDP คือ บิตที่ใช้อ่านค่า internal pointer (WDP) WDP จะ Reset เป็น “0” เมื่อระบบ Reset หรือ WDT Counter Overflow

d7 ถึง d0 (Bit 7 ถึง 0)

d7 ถึง d0 คือ บิตที่ใช้เขียนข้อมูลสำหรับการ Clear ค่า WDT Counter ซึ่ง WDT Counter สามารถ Clear ได้โดยเขียนค่า “0x5A” ในขณะที่ internal pointer (WDP) เป็น “0” และเมื่อเขียน “0xA5” WDP จะเป็น “1” WDT จะไม่ Clear ถ้าการ Clear WDT Counter

2) Watchdog Timer Mode Register (WDTMOD)

Address : 0xF012

Access : R/W

Access size : 8 bits

Initial value : 0x06

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	-															
Byte symbol	-								WDTMOD							
Bit symbol	WOF1	WOF0	.	WDT2	WDT1	WDT0
Access type	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

รูปที่

WDTMOD คือ specific function register (SFR) ที่ใช้ตั้งค่าคาบเวลา overflow ของ WDT counter และทำให้ WDT สามารถ Clear คาบเวลาได้

รายละเอียดแต่ละบิต

WDT 2 ถึง 0 (Bit 2 ถึง 0)

WDT2, WDT1, WDT0 คือ บิตที่ใช้ตั้งค่าคาบเวลา overflow (T_{Wov}) ของ WDT Counter

WDT2	WDT1	WDT0	รายละเอียด
0	0	0	ประมาณ 7.8 mS
0	0	1	ประมาณ 15.6 mS
0	1	0	ประมาณ 31.3 mS
0	1	1	ประมาณ 62.5 mS
1	0	0	ประมาณ 125 mS
1	0	1	ประมาณ 500 mS
1	1	0	ประมาณ 2 S (ค่าเริ่มต้น)
1	1	1	ประมาณ 8 S

หมายเหตุ เมื่อความถี่ของ WDT counter clock คือ 1 kHz

WOVF1 ถึง 0 (Bit 5 ถึง 4)

WOVF1 ถึง 0 คือ บิตที่ใช้เลือกช่วงคาบเวลาของ WDT ที่สามารถ Clear ได้ ()

เมื่อ WDT มีคาบ overflow 62.5 mS หรือน้อยกว่าค่าคาบ overflow ของ WDT counter ที่ตั้งไว้ WDT นั้นจะทำการ clear ในช่วงจากคาบเวลาที่กำหนดไว้ไปจนถึง 100% ไม่สนใจข้อมูลที่ตั้งไว้ใน WOVF1 และ WOVF0

WOVF1	WOVF0	รายละเอียด
0	0	100% ของคาบเวลา overflow
0	1	75% ของคาบเวลา overflow
1	0	50% ของคาบเวลา overflow
1	1	

Note : เมื่อตั้งค่า clear คาบเวลา 50% หรือ 75% ของคาบเวลา overflow WDT ไม่สามารถเกิด interrupt ได้ และเมื่อเกิด overflow ครั้งแรก WDT จะ reset ค่า

3) Watchdog Status Register (WDTSTA)

Address : 0xF016

Access : R

Access size : 8 bits

Initial value : 0x01

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Word symbol	-															
Byte symbol	-								WDTSTA							
Bit symbol	WDTCLR2	WDTCLR1	WDTWIN
Access type	ร	ร	ร	ร	ร	ร	ร	ร	ร	ร	ร	ร	ร	ร	ร	ร
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

รูปที่

WDTSTA คือ specific function register (SFR) ใช้อ่านสถานะสำหรับการ clear ค่า WDT Counter รายละเอียดของแต่ละบิต

WDTWIN (Bit 0)

WDTWIN เป็นบิตที่แสดงว่า WDT Counter สามารถ Clear ค่าได้หรือไม่

WDTWIN	รายละเอียด
0	WDT Counter ไม่สามารถ clear ค่าได้
1	WDT Counter สามารถ clear ค่าได้

WDTCLR1 (Bit 1)

WDTCLR1 เป็นบิตที่ใช้อ่านสถานะการทำงาน WDT Counter

บิตนี้จะถูกตั้งค่าเป็น “1” เมื่อเขียนค่า “0x5A” และ “0xA5” ลง WDTCON register และ WDT