บรรณนุกรม

[1] นคร ภักดีชาติ และคณะ. คู่มือทดลองไมโครคอนโทรลเลอร์ 32 บิตตระกูล ARM7 เบื้องต้น ฉบับ LPC2148. บริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด.

[2] บริษัทอีทีที จำกัด. คู่มือการใช้งานบอร์ด CP-JR ARM7 USB-LPC2148 EXP [ออนไลน์]: http://www.ett.co.th/product/ARM/images/CP_JR_ARM7_LPC2138/MAN_CP_JR_ARM7_LPC2138.pdf

[3] บริษัทอีทีที จำกัด. คู่มือการใช้งานบอร์ด ET-MINI [ออนไลน์]: http://www.ett.co.th/product/InterfaceBoard/P-ET-A-00230/MAN_MINI_I_15_Board.pdf

[4] รังสรรค์ ทองทา และ ประโยชน์ คำสวัสดิ์. 2545 ความไวของตัวตรวจจับแบบวงกลมที่เนื่องมาจากยานพาหนะขนาดเล็ก. รายงานฉบับสมบูรณ์ มหาวิทยาลัยเทคโนโลยีสุรนารี

[5] วีรชัย อาจหาญ. (2552). โครงการสาธิตการประยุกต์ใช้งานระบบควบคุมสัญญาณไฟจราจรอัจฉริยะเพื่อลดการใช้พลังงานในภาคขนส่ง. สำนักงานนโยบายและแผนพลังงาน กระรวงพลังงาน

[6] โครงการวิจัยและพัฒนาระบบ Intelligent Transportation system 2550. ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ

[7] ชญานนท์ มินเสน และคณะ, การพัฒนาเครื่องควบคุมสัญญาณไฟจราจรแบบตรวจจับยานพานะบนท้องถนนเพื่อตัดสินใจในการกำหนดสัญญาณไฟจราจรที่เหมาะสม (The development of traffic signal controller using the vehicle detector to set up proper traffic signals.) , มหาวิทยาลัยเทคโนโลยีมหานคร และมหาลัยวิทยาเชียงใหม่

[8] กิตติพงศ์ ศิลปะวงศ์เจริญ และคณะ, เครื่องตรวจจับและนับรถยนต์ (Car detector and counter) คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น

[9] James L.Pline, Traffic Engineering Handbook, Fifth Edition, Institute of Transportation Engineerings 1099 14th Street, NW Suite 300 West Washington, DC 20005-3438 USA.

[10] Sumit Ghosh, Tony S.Lee, Intelligent Transportation System. Smart and Green Infrastructure Design. Second Edition.

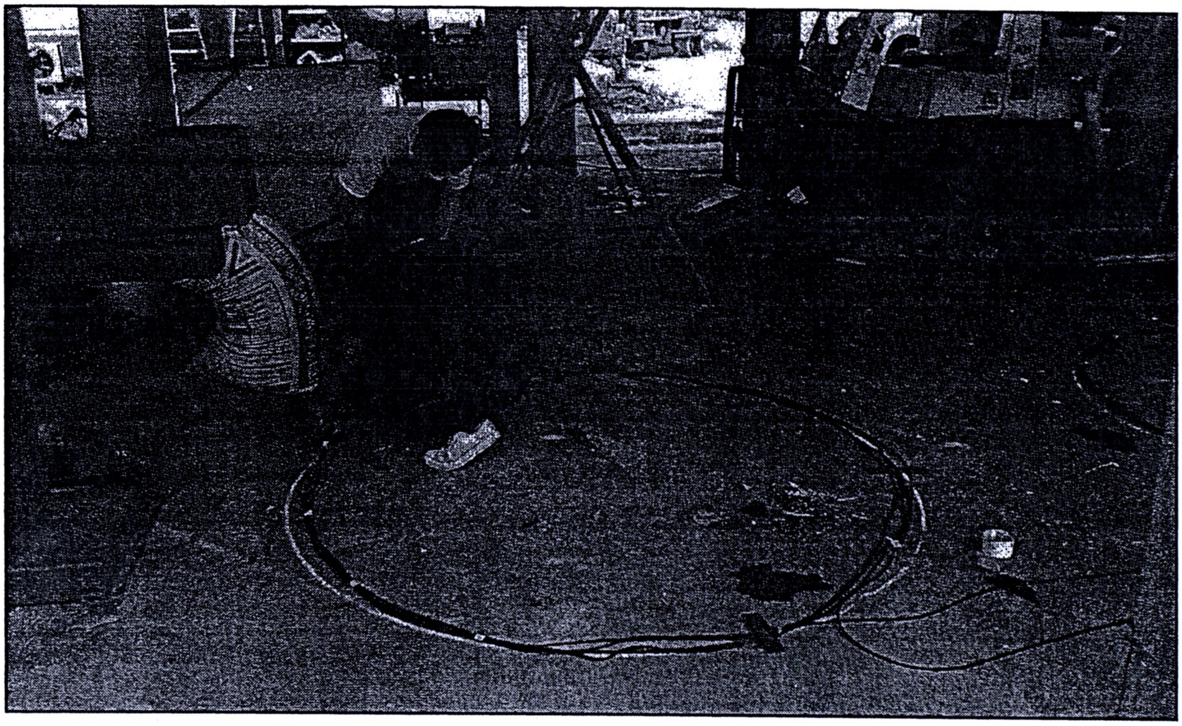
[11] Eric Norum. 2002. Vehicle Detector, Lab (Online). Available [On line]: http://www.engr.usask.ca/classes/EP/424/lab/EP414LabManual/node5.html.

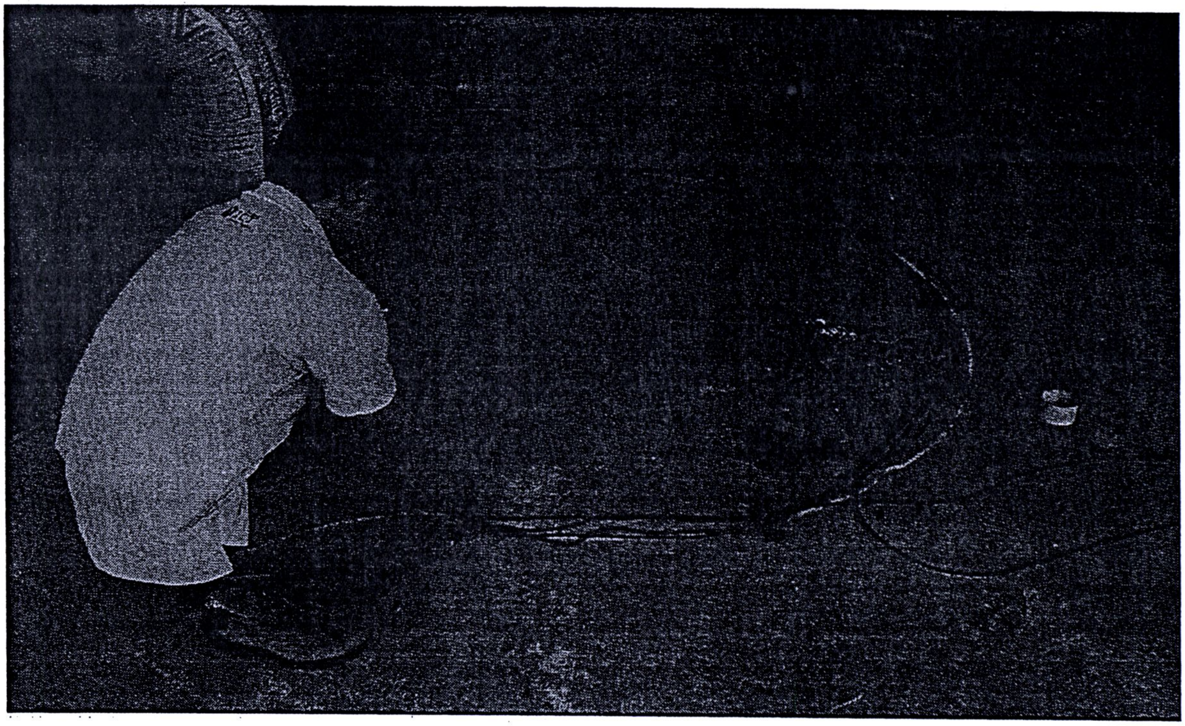
[12] Paul Consentino and Barry Grossman. 1996. Development of Fiber Optic Dynamic Weight-in-Motion Systems. Report Submitted to Transportation Statistic Office, Florida Department of Transportation, and Tallahassee, Florida.

[13] U.S. Department of Transportation. 1990. Traffic Detector Handbook. 2^{nd} Edition. McLean, Virginia

ภาคผนวก

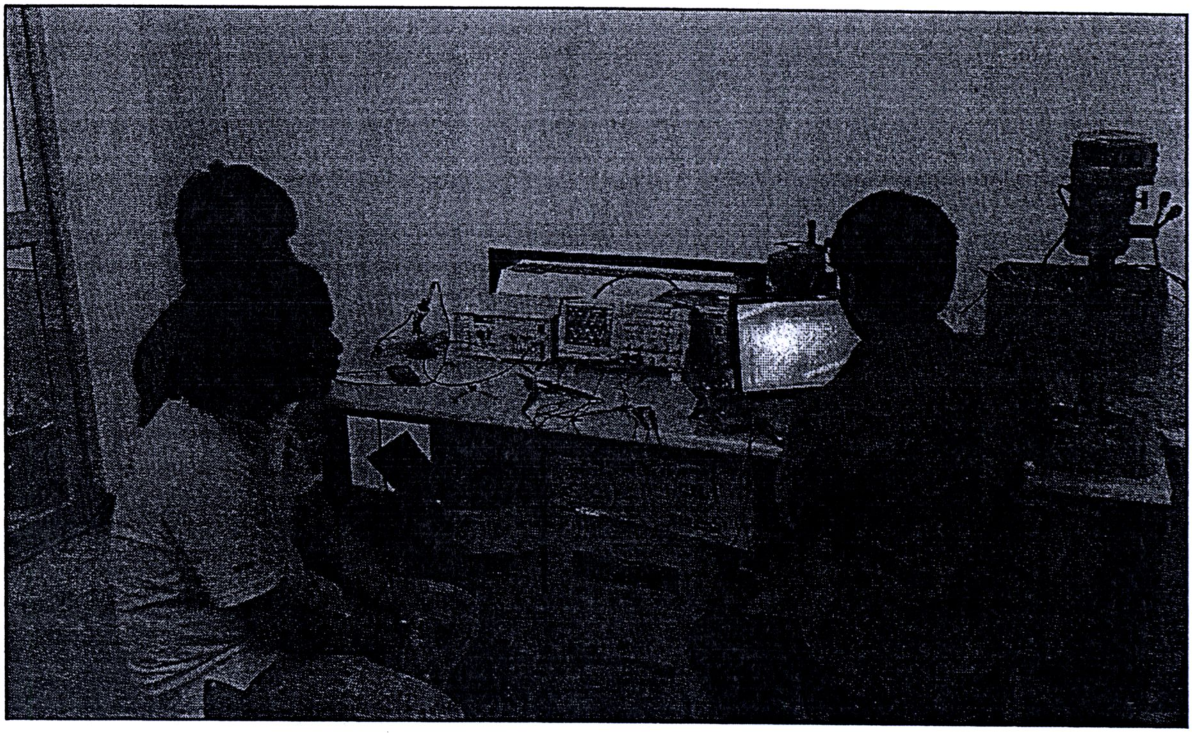
การจำลองลูปเหนี่ยวนำ (Loop Detector) ก่อนที่จะติดตั้งบนท้องถนน

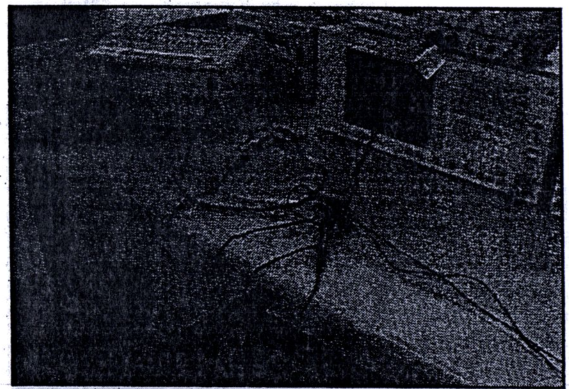


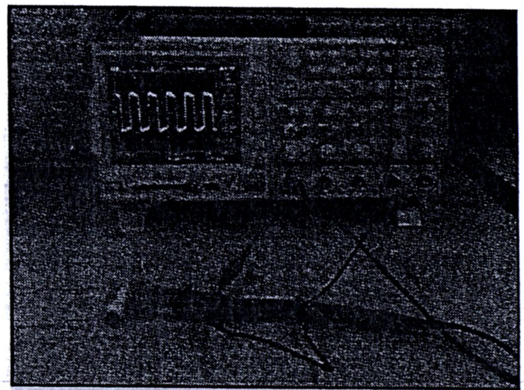


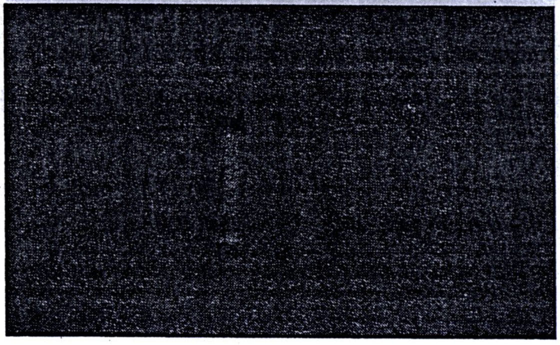
การทดสอบด้วยยานพาหนะขนาดเล็กก่อนที่จะติดตั้งบนท้องถนน

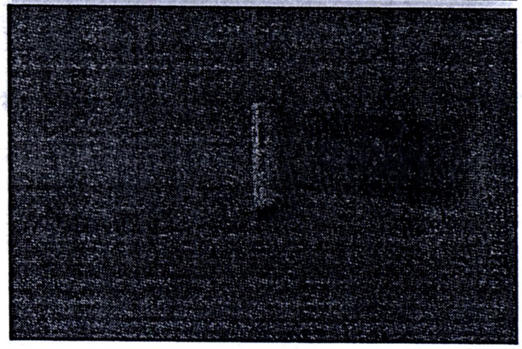
การทดสอบความถี่เรโซแนนซ์ของหม้อแปลงที่เชื่อมต่อกับลูปเหนี่ยวนำ





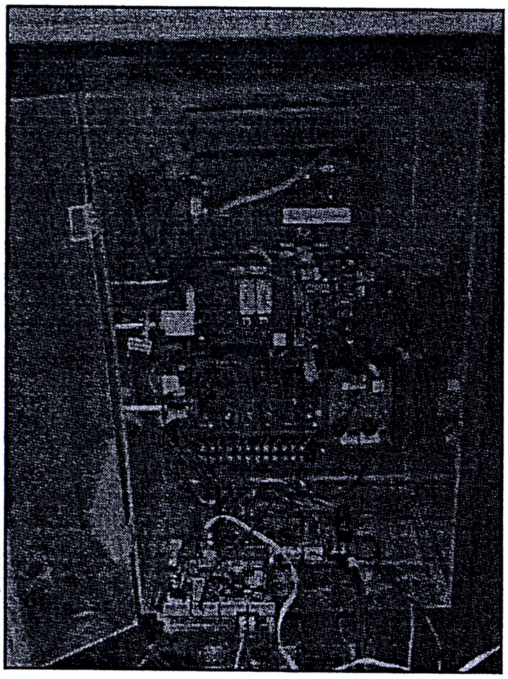


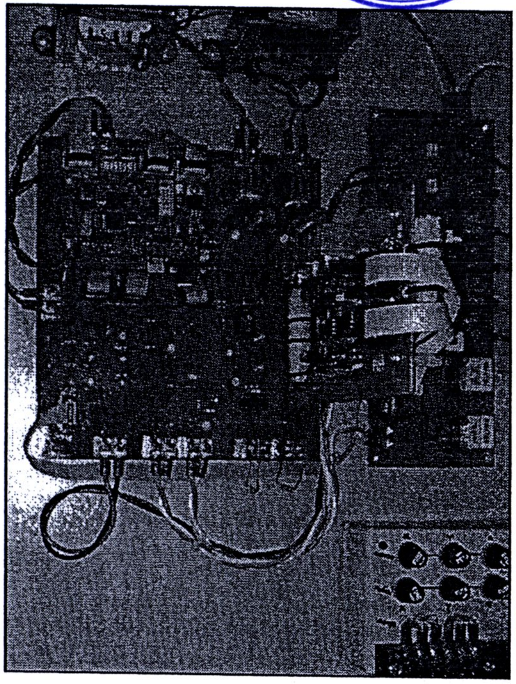




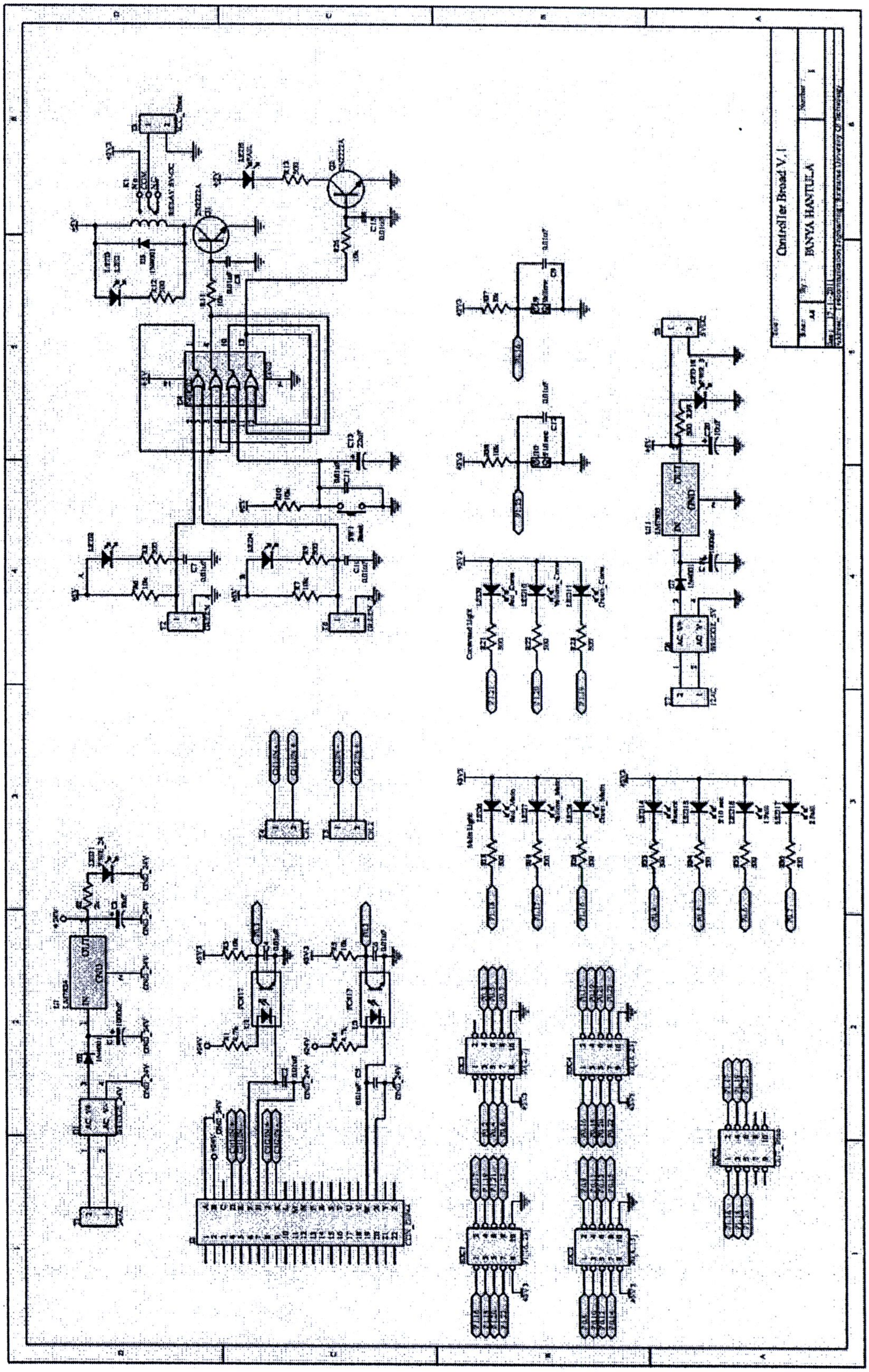
สำนักงานคณะกรรมการวิจัยแห่งชาติ
ห้องสมุดงานวิจัย

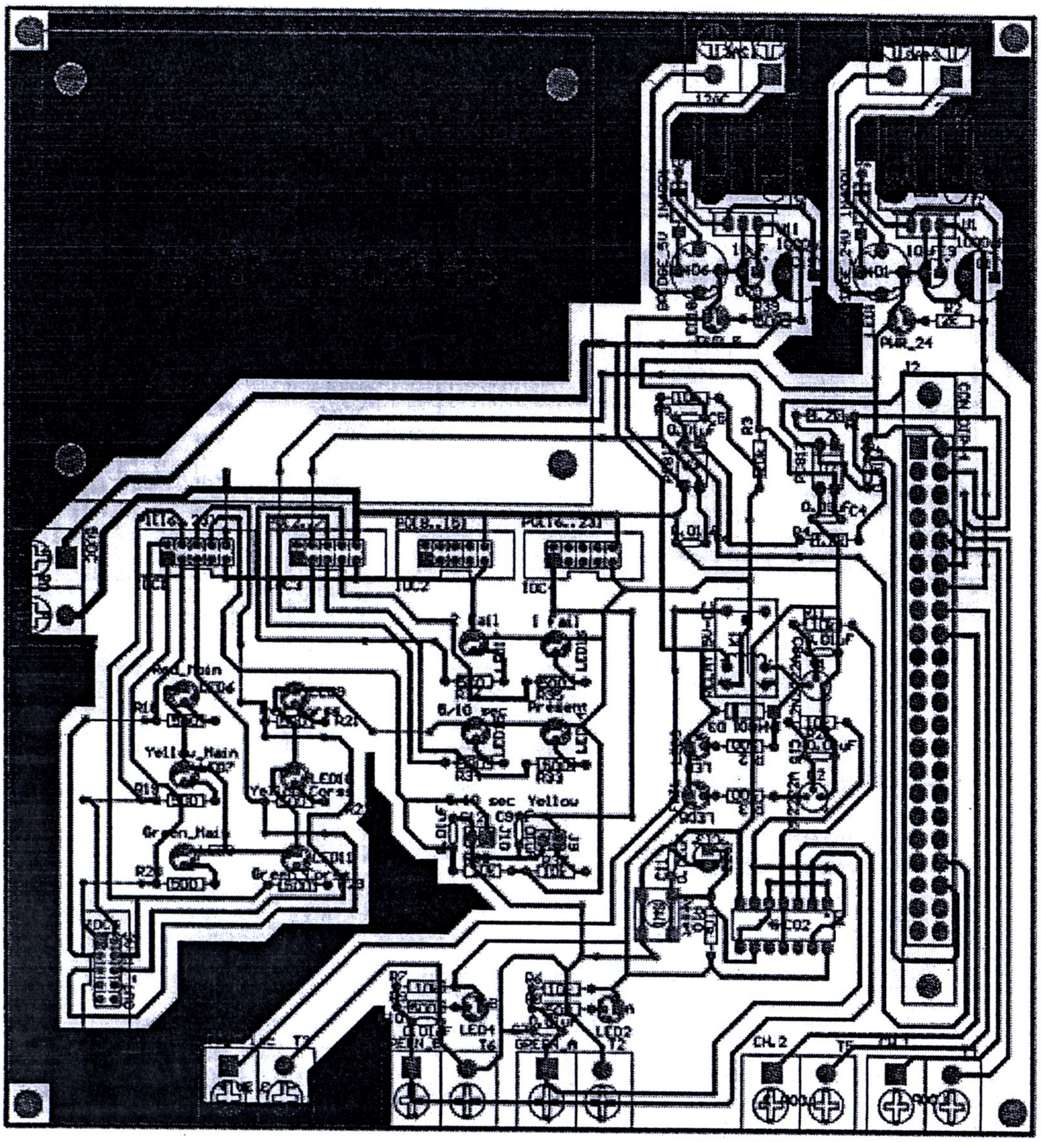
ทดสอบตัวตรวจจับยานพาหนะกับลูปเหนียวนำที่ติดตั้งอยู่บนท้องถนน

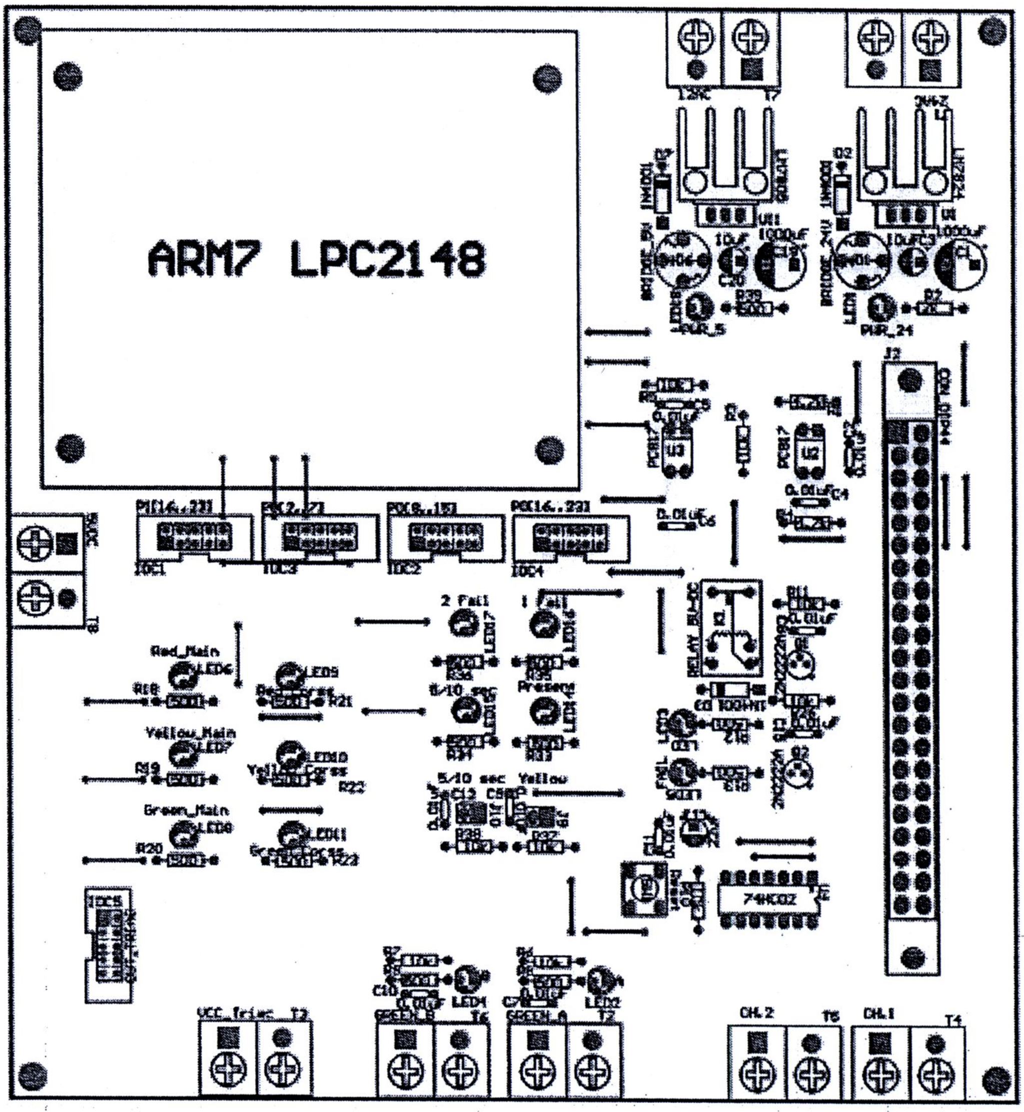


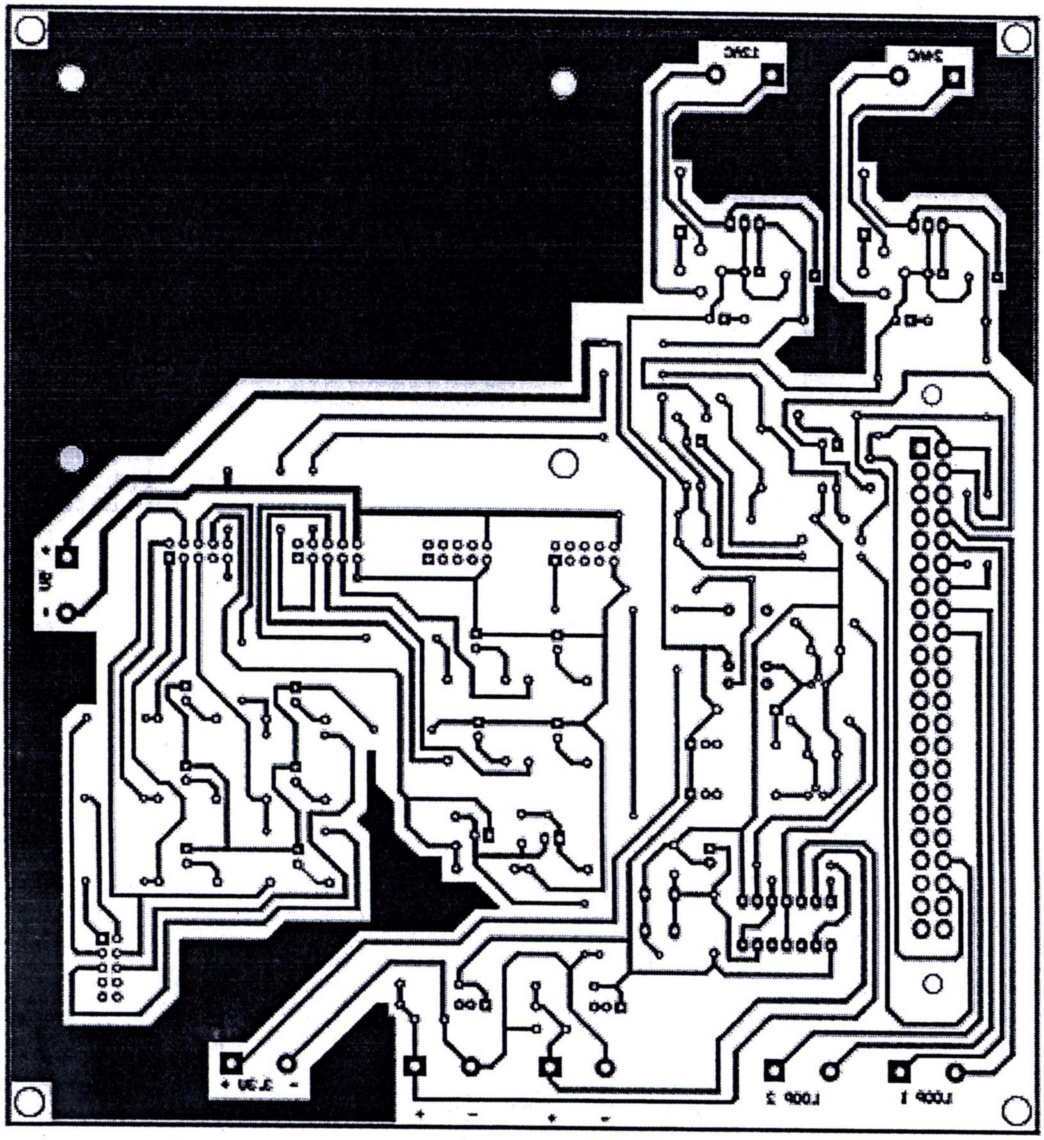


DATASHEET ระบบควบคุมสัญญาณไฟจราจร (Controller)

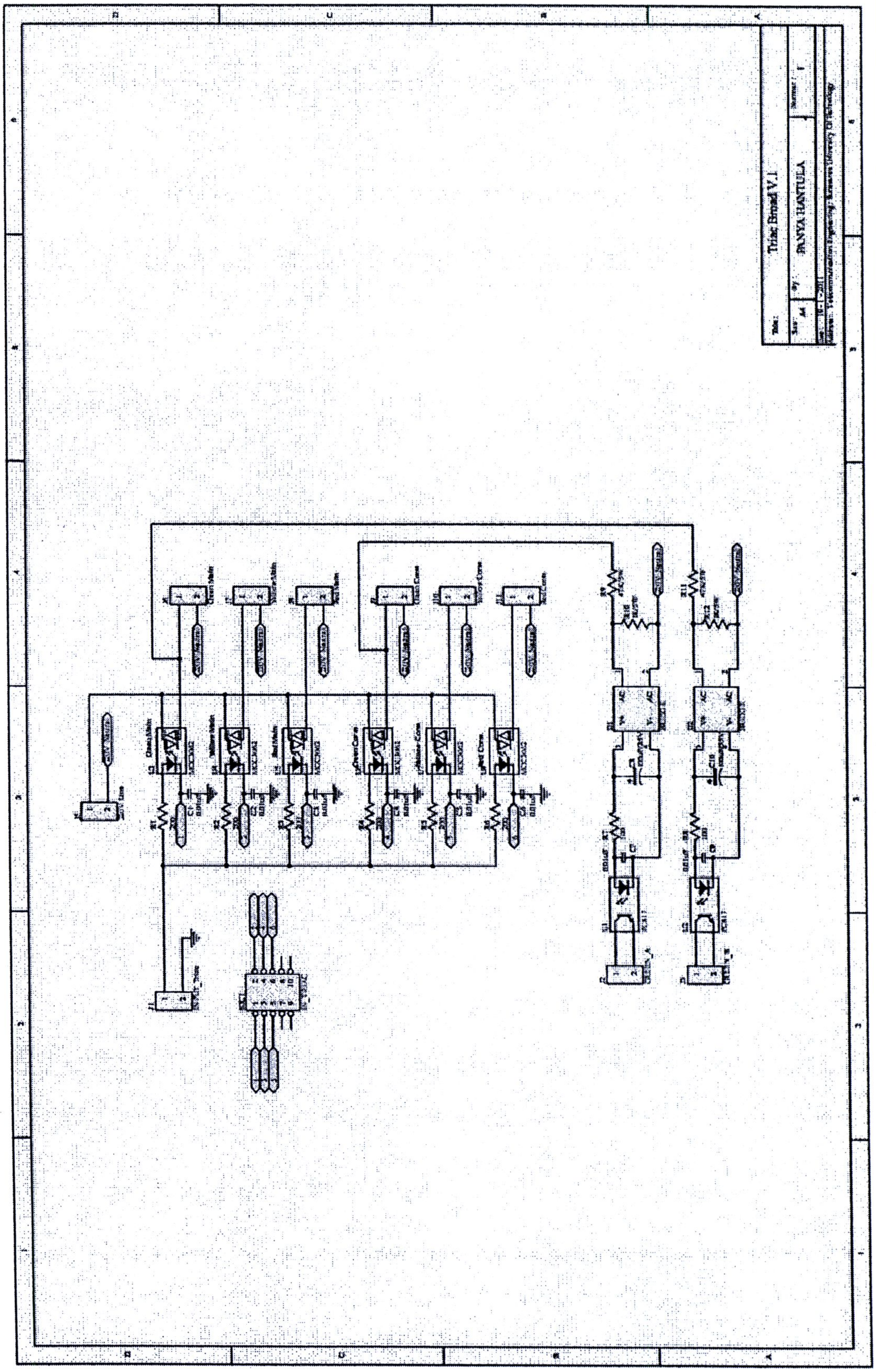


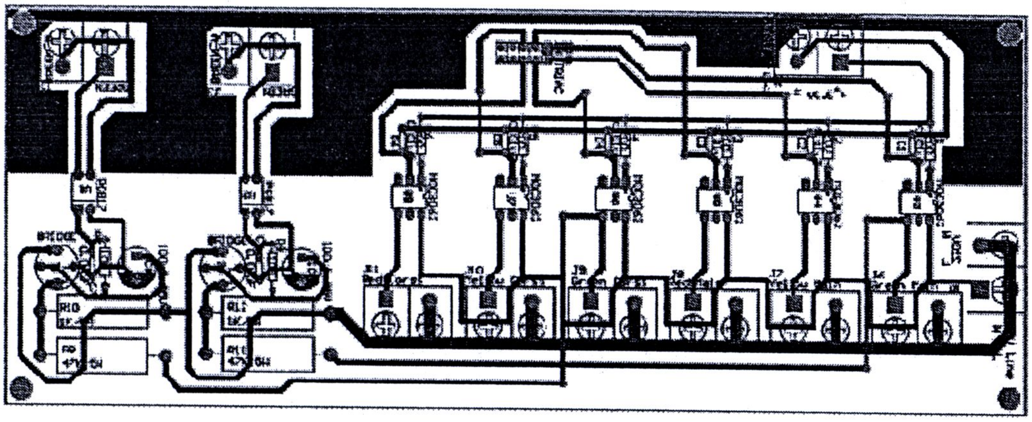

Red_Main
Yellow_Main
Green_Main
Present
PWR_24
LED4
LED2
CH.2

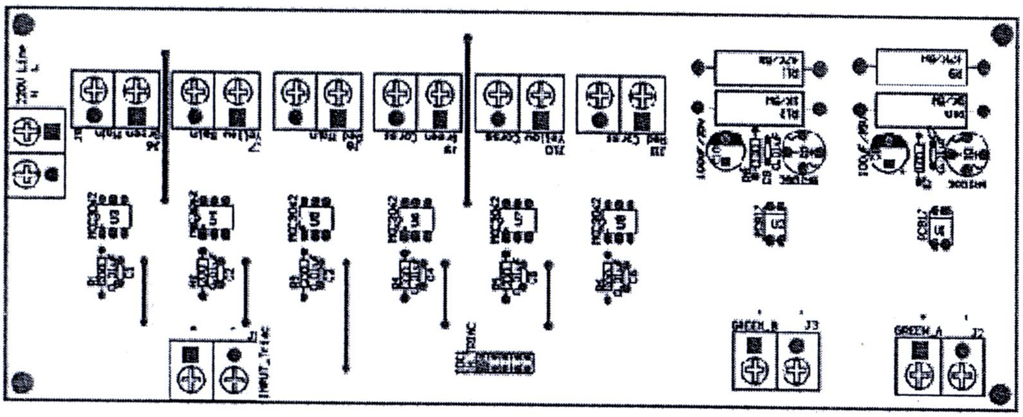

ARM7 LPC2148
74HC02
Red_Main
Yellow_Main
Green_Main
VCC_triac
CH.2
CH.1

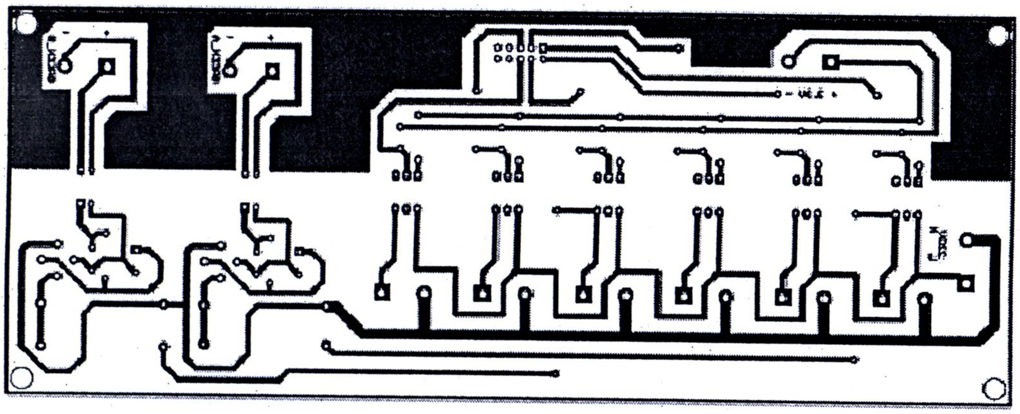


DATASHEET วงจรควบคุมการเปิดและปิดสัญญาณไฟจราจร

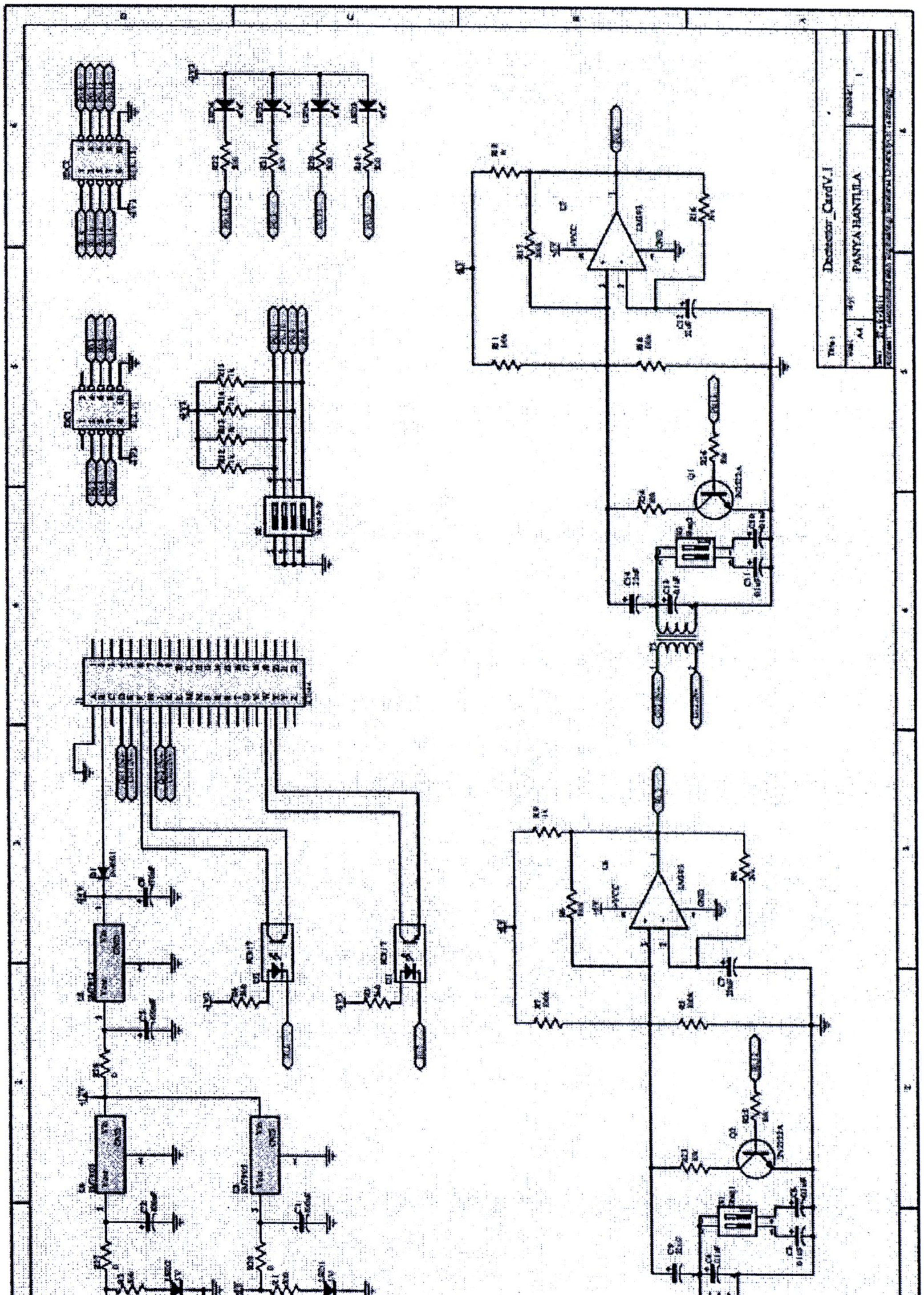


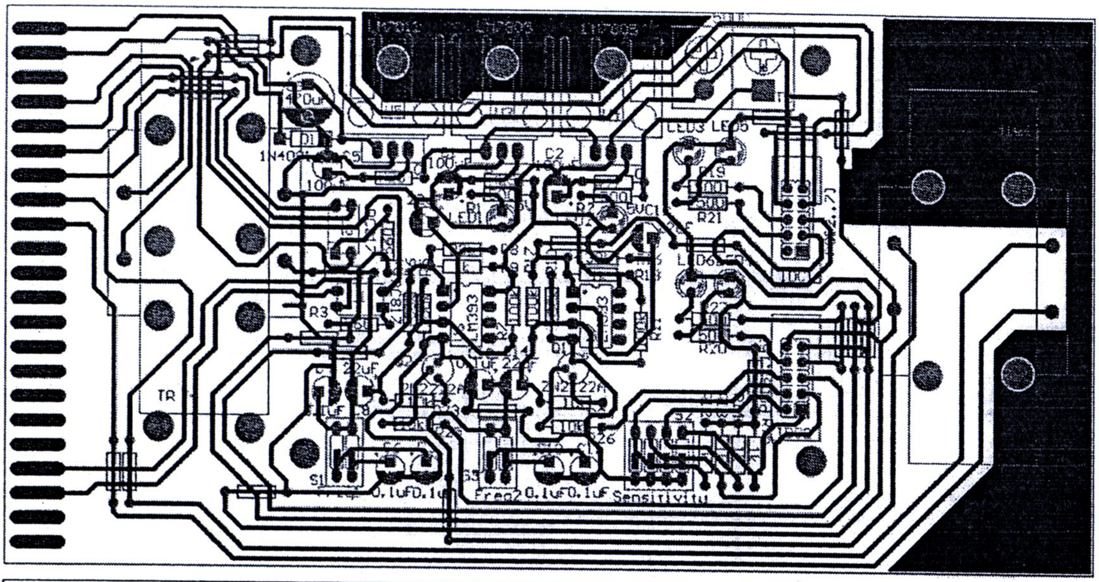


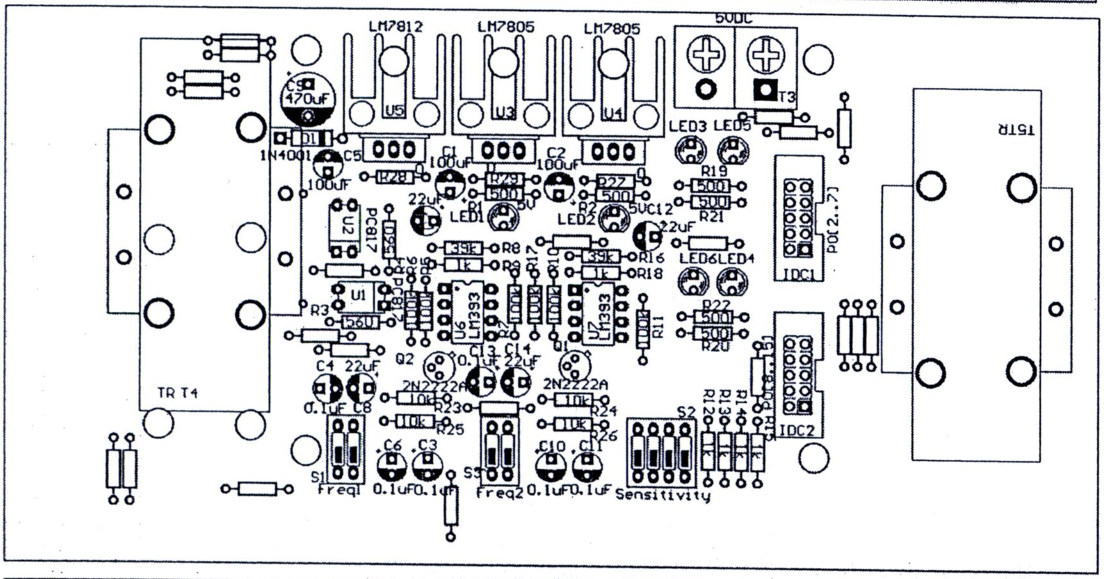

GREEN_B
J3
GREEN_A
J2

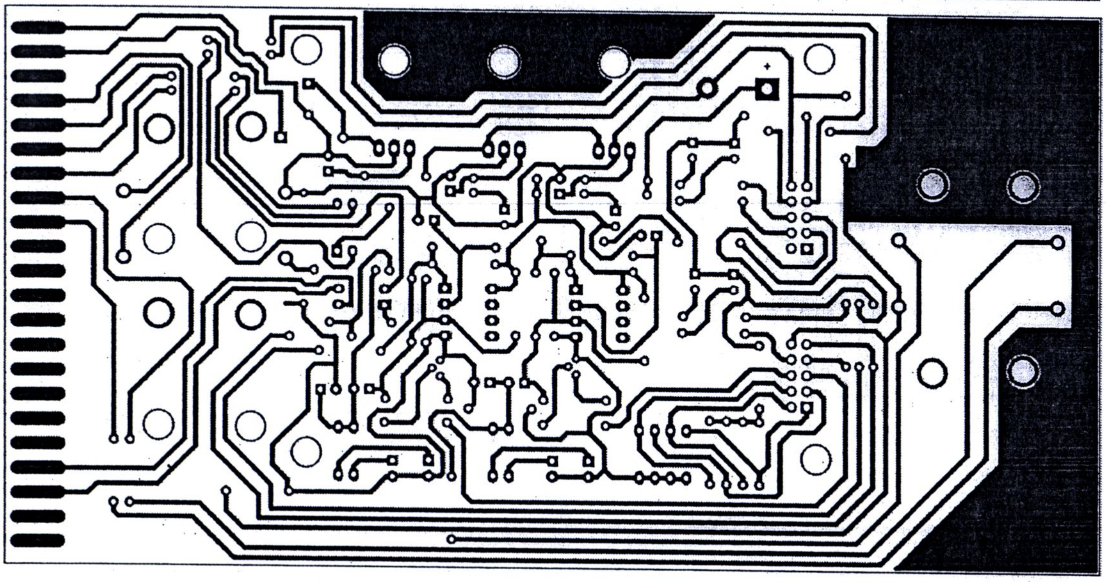


DATASHEET ตัวตรวจจับยานพาหนะ (Detector)

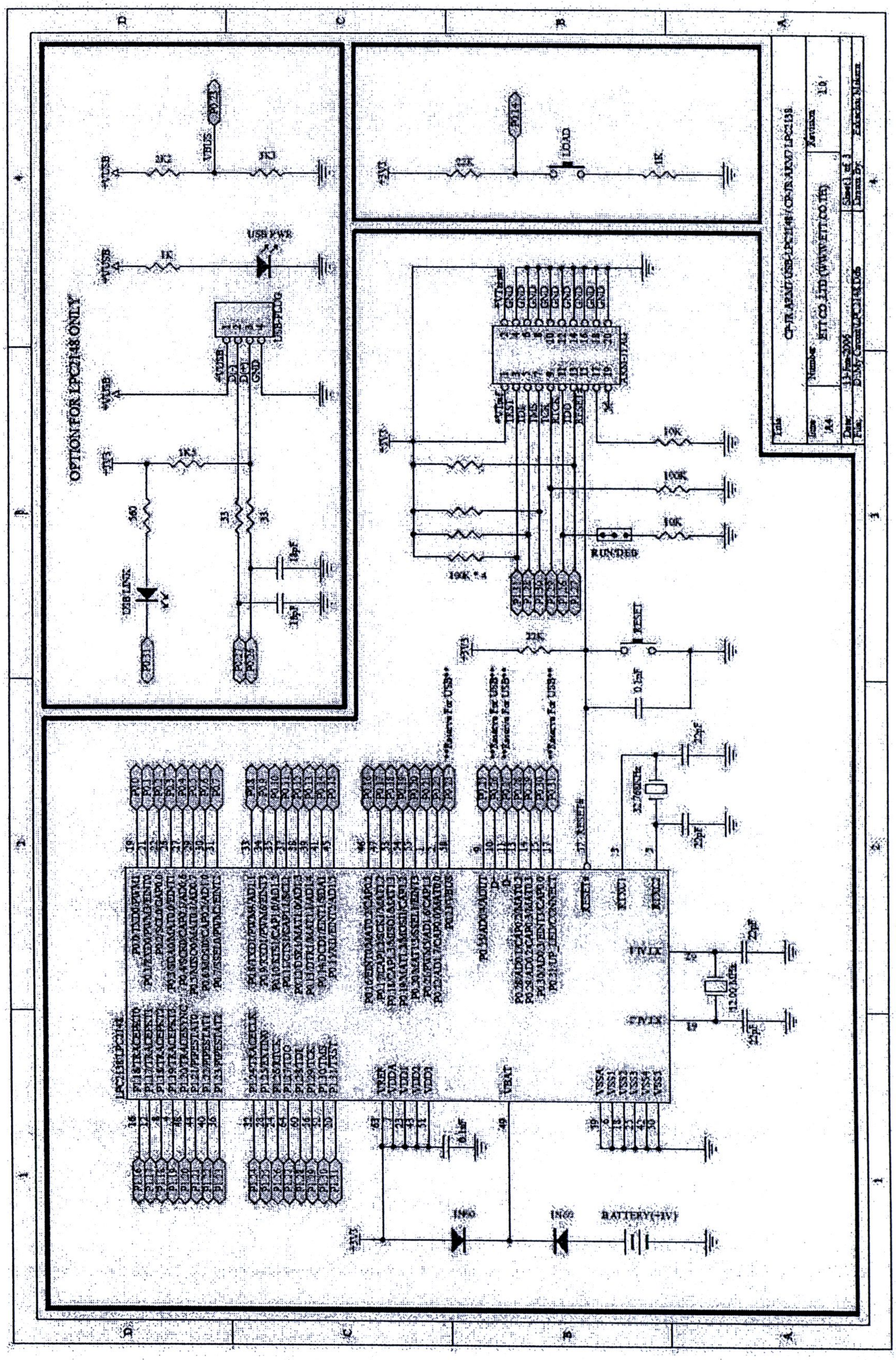


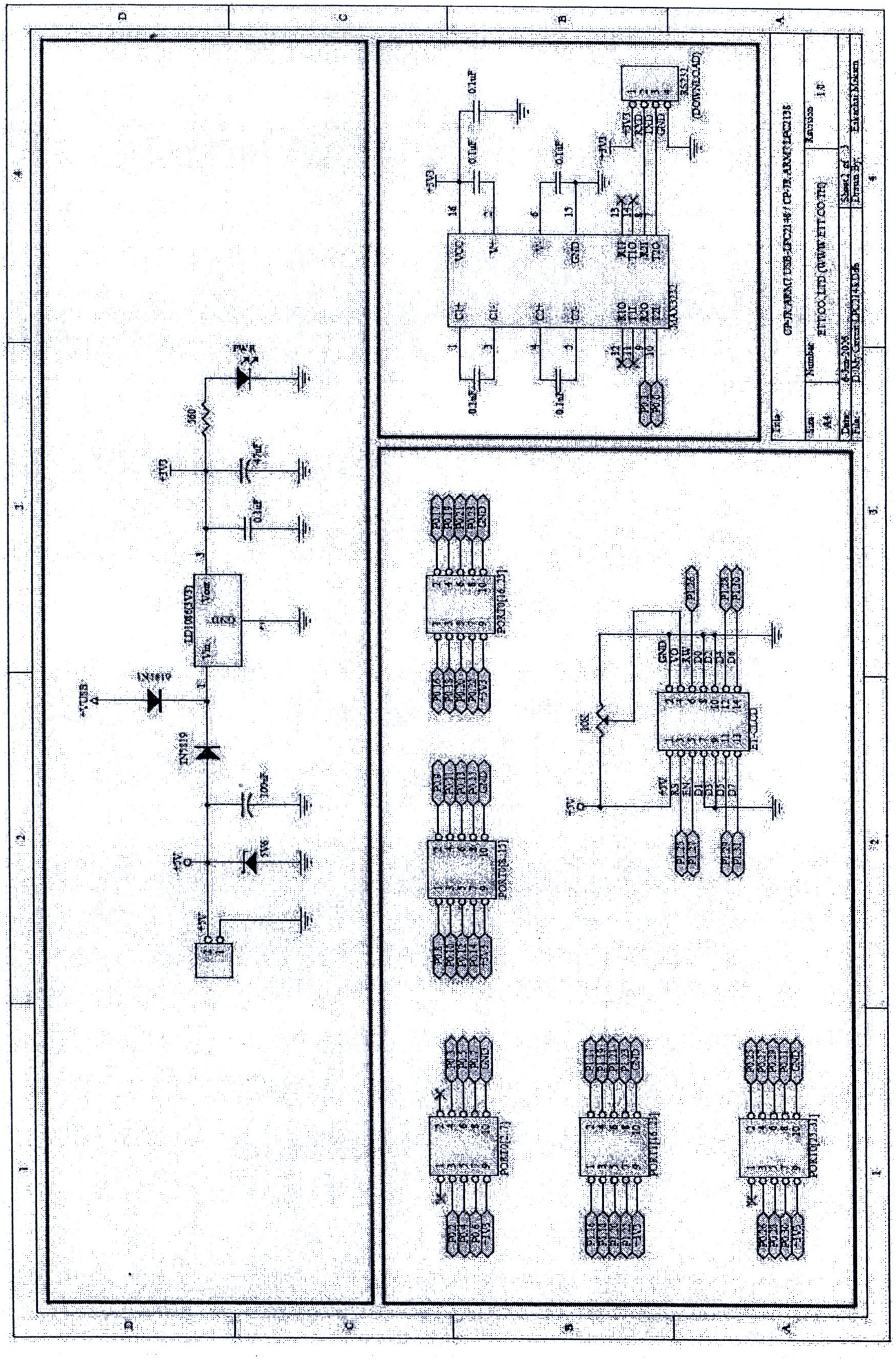



LM7812
LM7805
LM7805
5VDC
U5
U3
U4
T3
LED3 LED5
C9
470uF
D1
1N4001
C5
100uF
C1
100uF
C2
100uF
R28
R29
500
R27
500
R19
500
500
R21
22uF
LED1
5V
LED2
5VC12
22uF
PC817
U2
39k
1k
R8
39k
1k
R16
R18
LED6LED4
R22
500
500
R20
P0[2..7]
IDC1
U1
R3
56U
U6
LM393
U7
LM393
R11
Q2
C13 C14
0.1uF 22uF
Q1
2N2222A
2N2222A
C4 22uF
0.1uF C8
10k
R23
R25
10k
R24
10k
R26
S2
IDC2
S1
Freq1
C6 C3
0.1uF0.1uF
S3
Freq2
C10 C11
0.1uF0.1uF
Sensitivity
TR T4

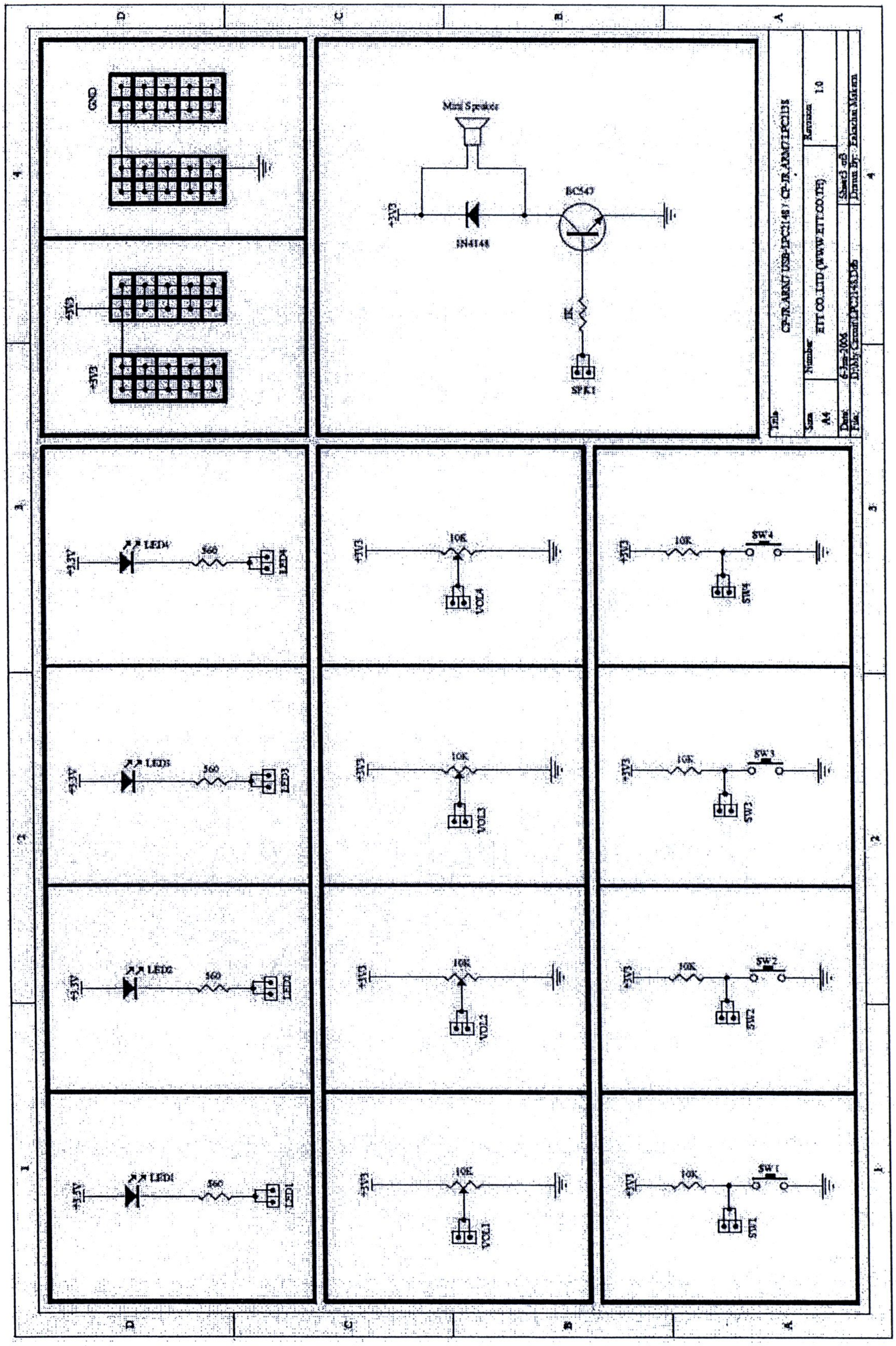


DATASHEET MICROCONTROLLER (CP_JR_ARM7_USB_LPC2148)





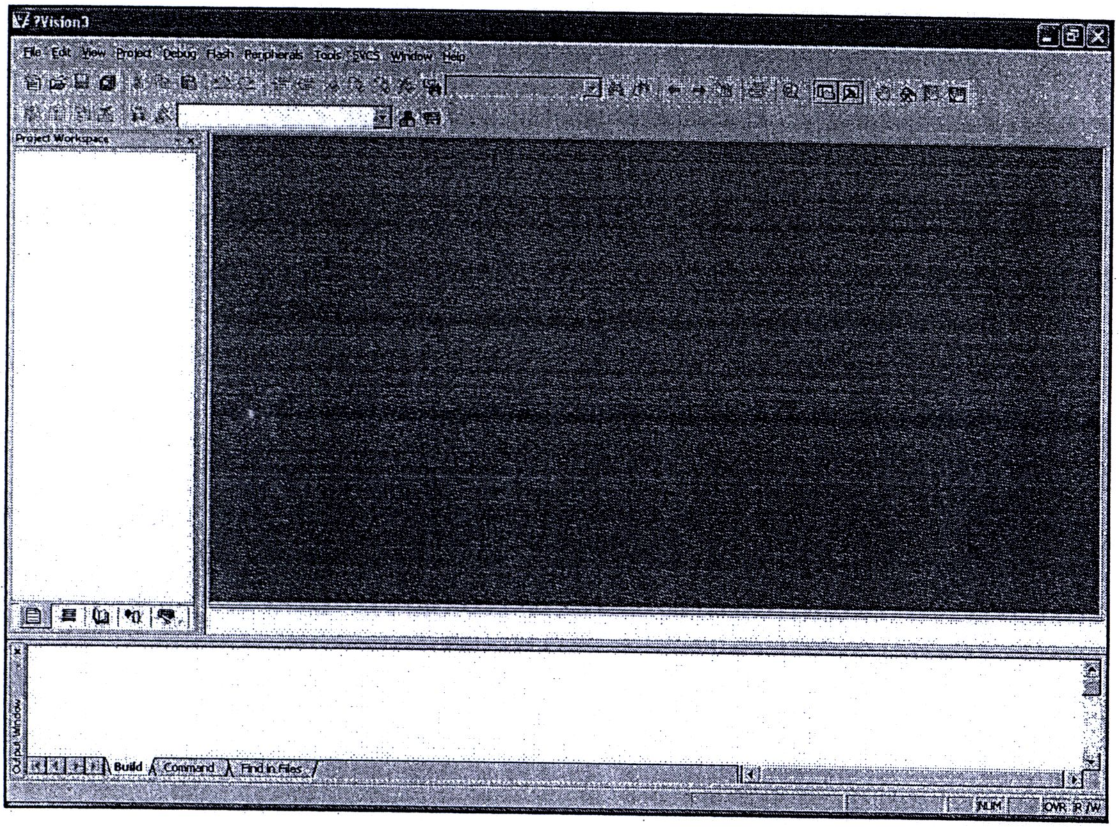
LD1086(3V3)
1N5819
PWR
MAX3232
RS232 (DOWNLOAD)


GND
+3V3
Mini Speaker
1N4148
BC547
1K
SPK1
LED1
LED2
LED3
LED4
560
+3.3V
10K
VOL1
VOL2
VOL3
VOL4
SW1
SW2
SW3
SW4
Title
CP-JR ARM7 USB-LPC2148 / CP-JR ARM7 LPC2138
Number
ETT CO.,LTD (WWW.ETT.CO.TH)
Revision
1.0
Size
A4

การใช้ Keil uVision3 ในการสร้าง Project File ของ Keil ARM

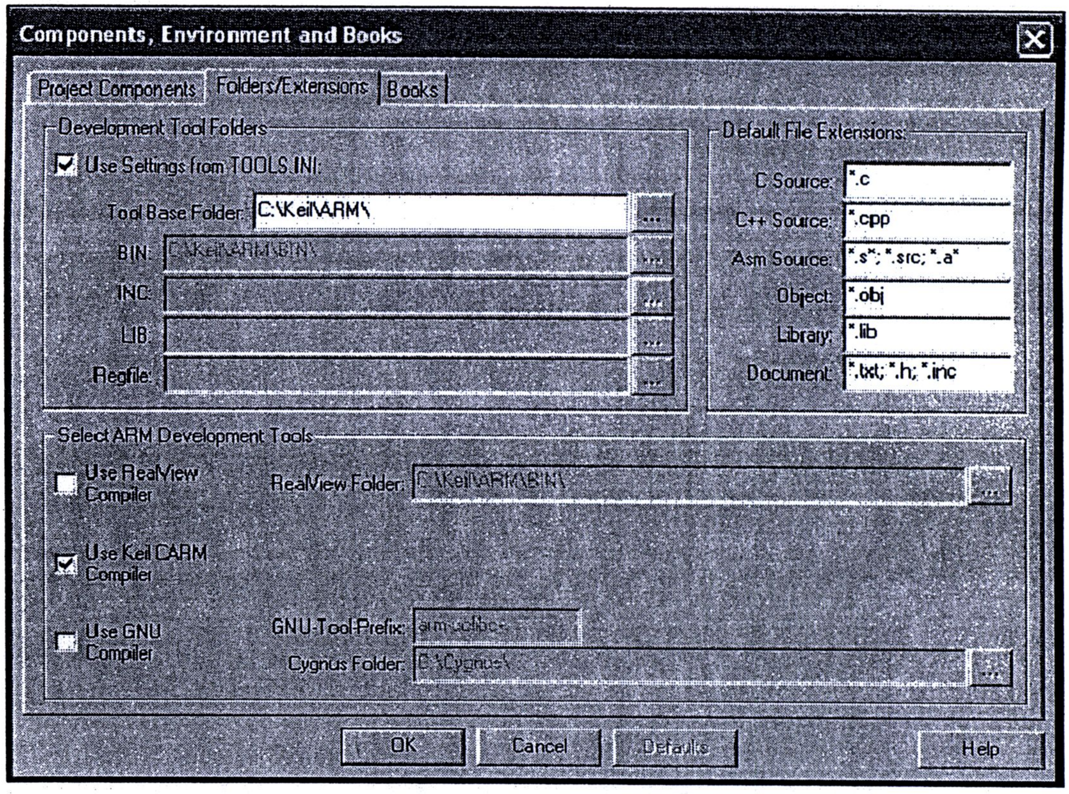
ในที่นี้จะขอแสดงแนวทางการเขียนโปรแกรมภาษาซีโดยใช้ Keil-CARMในการแปลคำสั่งภายใต้โปรแกรม Text Editor ของ Keil (Keil uVision3) โดยจะขออธิบายถึงเฉพาะวิธีการกำหนดค่า Option สำหรับเชื่อมโยงคำสั่งในการสั่งแปลโปรแกรมด้วย Keil-CARM ผ่านทาง Keil uVision3 เท่านั้น ส่วนรายละเอียดคำสั่งและการใช้งานฟังก์ชันต่างๆในการเขียนโปรแกรมของ Keil-CARM นั้นขอให้ผู้ใช้ศึกษาจากคู่มือคำสั่งของ Keil-CARM เอง โดยวิธีการกำหนดค่าตัวเลือกของ Keil uVision3 ให้ใช้งานกับ Keil-CARM นั้นมีขั้นตอนพอสังเขปดังนี้คือ

1. เปิดโปรแกรม Keil uVision3 ซึ่งเป็นโปรแกรม Text Editor ของ Keil-CARM ใช้สำหรับใช้ในการเขียนโปรแกรมที่เป็น Source Code ภาษาซี โดยจะมีลักษณะดังรูป

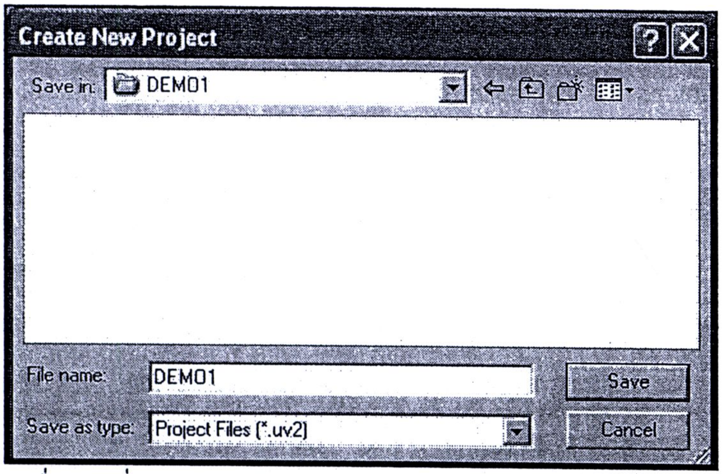


2. ทำการกำหนดค่าตัวเลือกในการแปลคำสั่งของ uVision3 ให้ใช้งานกับโปรแกรม Keil uVision3 และ Keil-CARM โดยให้เลือกคลิกเมาส์ที่เมนูคำสั่ง Project ➔ Components ,Environment, Books... จากนั้นให้เลือกค่าตัวเลือกสำหรับกำหนดการใช้งาน Complier จากหัวข้อ Select ARM Development Tools ซึ่งจะมีค่าตัวเลือกอยู่ 3 แบบ คือ Use Keil-CARM Tools ,Use GNU Tools และ Use ARM Tools โดยให้เลือกเป็น "Use Keil ARM Tools" จากนั้นให้ทำการกำหนดตำแหน่ง Folder

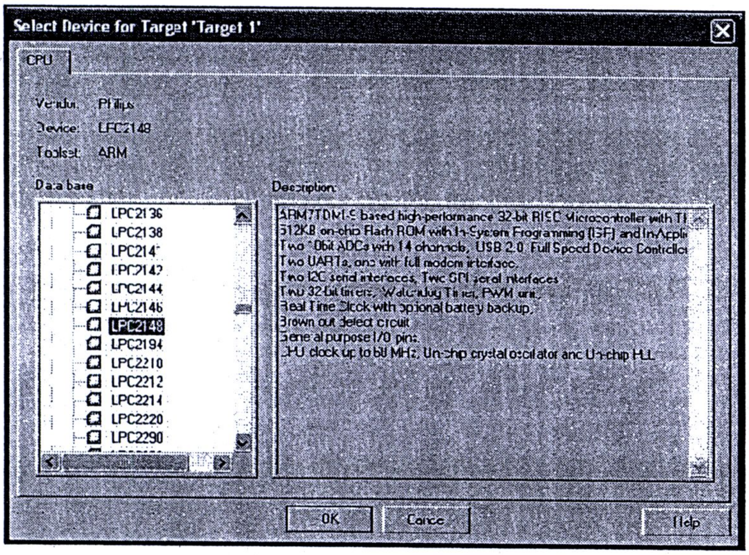
สำหรับเก็บค่าตัวเลือกการทำงานของโปรแกรม Keil ARM ซึ่งตามปรกติ แล้วจะเป็น “C:\Keil\ARM\” แต่ถ้าติดตั้ง Keil ไว้ที่อื่นก็ต้องปรับเปลี่ยนให้ถูกต้องตามความเป็นจริงด้วยดังรูป



3. ทำการสร้าง Project File ขึ้นมาใหม่ โดยเรียกเมนูคำสั่ง Project ➔ New Project จากนั้นให้เลือกกำหนดหรือสร้างตำแหน่ง Folder ที่จะบันทึก Project File พร้อมกับกำหนดชื่อ Project File ตามต้องการ เช่น ถ้าต้องการสร้าง Project File ชื่อ DEMO1 โดยเก็บไว้ใน Folder ชื่อ DEMO1 ก็สามารถกำหนดตำแหน่ง Folder และชื่อ Project Fileได้เอง โดยเมื่อกำหนดชื่อในช่อง File name แล้วให้เลือก Save เพื่อบันทึก Project File ไว้ ดังรูป

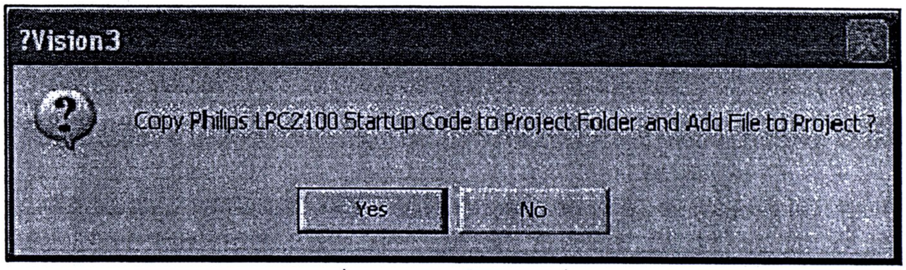


หลังจากกำหนดชื่อและสั่ง Save Project File แล้ว โปรแกรมจะรอให้ผู้ใช้ทำการกำหนดเบอร์ MCU ที่จะใช้งานใน Project ที่สั่ง Save นั้น ซึ่งในกรณีที่ใช้งานกับบอร์ด CP-JR ARM7 USB-LPC2148 นั้น ให้เลือกกำหนดเบอร์ของ MCU เป็น LPC2148 ของ Philips แล้วเลือก OK ดังรูป



หลังจากเลือกกำหนดเบอร์ของ MCU เป็นที่เรียบร้อยแล้ว ในขั้นตอนนี้โปรแกรมจะรอให้ผู้ใช้ยืนยันว่าต้องการจะทำการ Copy ไฟล์ Startup ของ Keil เพื่อใช้งานกับ MCU ของ Philips มาใช้ใน Project ด้วยหรือไม่ โดยStartup ไฟล์จะเป็นส่วนของการกำหนดค่าเริ่มต้นการทำงานให้กับ MCU เช่น การกำหนดค่า Stack และการกำหนดค่าการทำงานให้กับ Phase-Lock-Loop ต่างๆ ก่อนที่จะเริ่มต้นทำงานตามโปรแกรมที่เราเขียนขึ้น ไม่เช่นนั้นแล้วโปรแกรมที่เราเขียนขึ้นมานั้นจะต้องเพิ่มคำสั่งในการเตรียมการทำงานส่วนเหล่านี้ให้ MCU เองทั้งหมดแต่เนื่องจากไฟล์ Startup ของ Keil-ARM นั้น เป็นไฟล์ภาษา แอสแซมบลี้(Assembly) ซึ่งกำหนดค่าการทำงานไว้กับชุดพัฒนาของ Keil เอง ดังนั้นข้อกำหนดและการกำหนดค่าบางอย่างจะมีความแตกต่างกันอยู่กับค่าที่ต้องการสำหรับบอร์ด “CP-JR ARM7 USB-LPC2148” ไม่สามารถใช้งานไฟล์ Startup ได้ทันที ต้องมีการแก้ไขค่าตัวเลือกใหม่ดังนั้นก่อนที่จะใช้โปรแกรม Keil-CARM ในการแปลคำสั่งให้นั้น ผู้ใช้จะต้องเข้าไปแก้ไขไฟล์ Startup ใหม่โดยต้องกำหนดรูปแบบให้ถูกต้องตรงกับความต้องการของบอร์ดด้วย

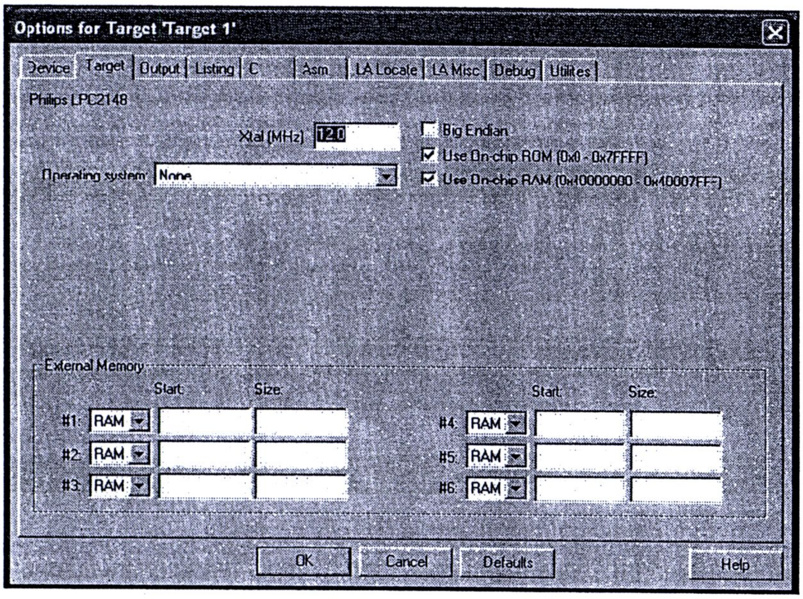
ดังนั้นในที่นี้ขอแนะนำให้เลือก "No" เพื่อไม่ให้ Keil uVision3 ทำการ Copy ไฟล์ Startup ของ Keil-CARM มาใช้ใน Project นี้ด้วย



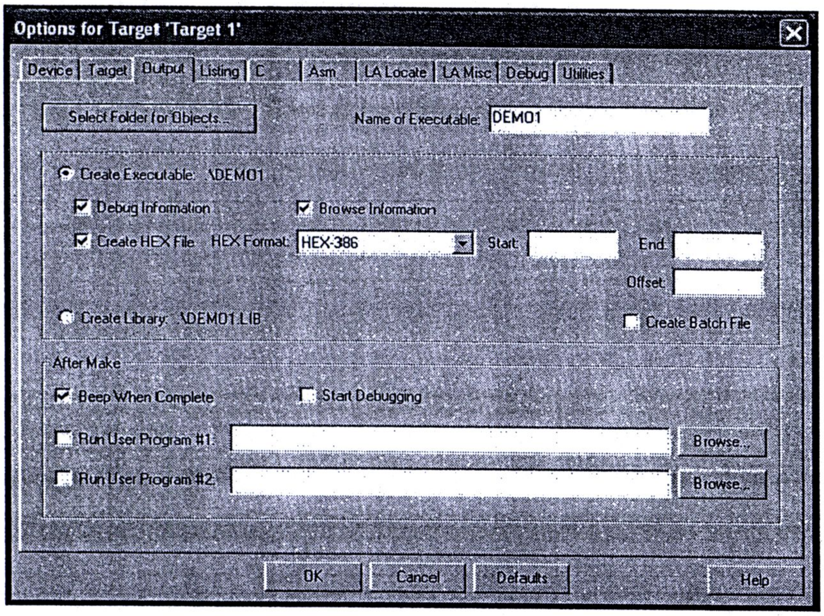
4. ให้ทำการ Copy File ชื่อ "Startup.s" ที่ทาง ผู้จัดทำจัดเตรียมไว้ใน CD-ROM ชื่อ "Startup.s" มาไว้ในตำแหน่ง Folder เดียวกันกับ Project File ที่สร้างขึ้นมาใหม่นี้โดยไฟล์ "Startup.s" จะเป็นไฟล์ซึ่งบรรจุคำสั่งภาษาแอสแซมบลี้ของ ARM7 สำหรับทำหน้าที่กำหนดค่าเริ่มต้นการทำงานที่จำเป็นให้กับ MCU ซึ่งได้แก่การ กำหนดค่า Stack ให้กับ MCU การ Initial Phase-Lock-Loopการกำหนดค่าให้กับ MAM Function และการกำหนดตำแหน่ง Vector ต่างๆของ MCU สำหรับใช้งานร่วมกับบอร์ด "CP-JR ARM7 USB-LPC2148" ซึ่งถ้าสั่ง Add ไฟล์ "Startup.s" จาก Keil หรือ Copy ไฟล์ดังกล่าวมาจากแหล่งอื่นๆ อาจมีการทำงานของโปรแกรมใน Startup ไม่เหมือนกัน ซึ่งจะส่งผลต่อการทำงานของโปรแกรมด้วย

5. ให้ทำการกำหนดค่า Option ของ Project File โดยเลือกเมนูคำสั่ง Project → Option for Target 'Target 1' จากนั้นเลือกที่ Tab ของ Target เพื่อกำหนดค่าของ MCU Target โดยให้กำหนดดังนี้

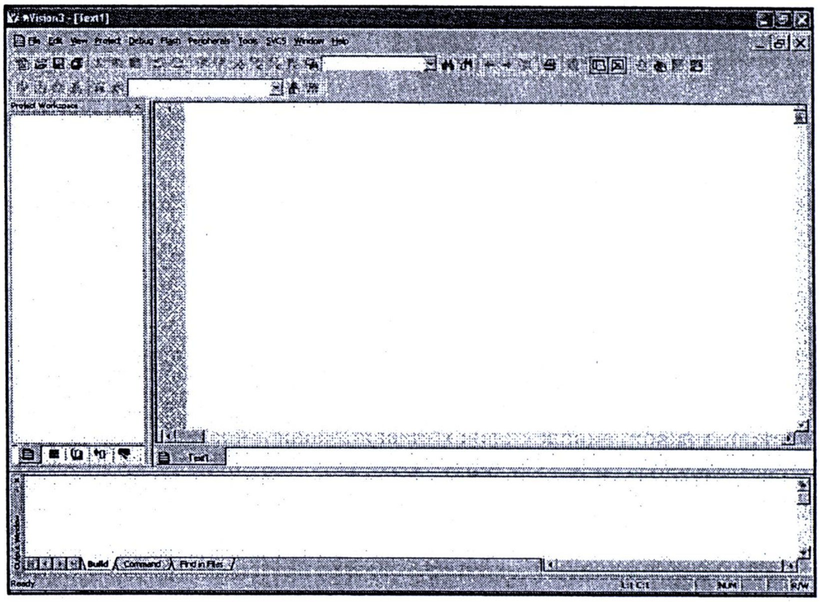
5.1 X-TAL ให้กำหนดเป็น 12 MHz พร้อมกับเลือกกำหนดให้ใช้หน่วยความจำที่มีอยู่ใน MCU เป็นเงื่อนไข ในการแปลโปรแกรมของ Keil-CARM ด้วย ดังรูป



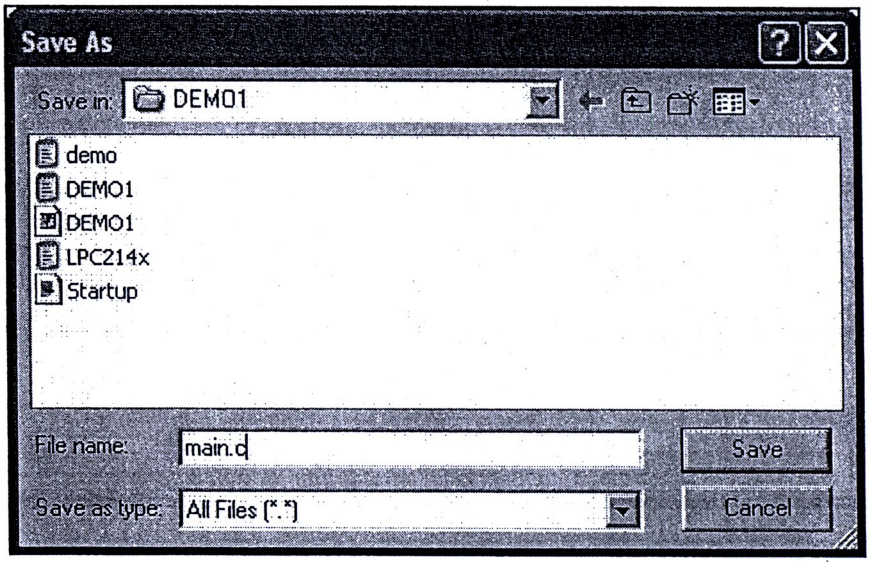
5.2 เลือกที่ Output ให้เลือกคลิกเมาส์ที่ค่าตัวเลือก Create HEX File พร้อมกับเลือกกำหนดรูปแบบของ Hex ให้เป็นแบบ HEX-386 แล้วเลือก OK ดังรูป



6. เริ่มต้นเขียน Source Code ภาษาซี โดยให้เลือกคลิกเมาส์ที่เมนูคำสั่ง File ➔ New... ซึ่งจะได้พื้นที่ในการเขียน Text File เกิดขึ้นมา โดยในครั้งแรกจะกำหนดชื่อตามค่า Default เป็น "Text1" ดังรูป



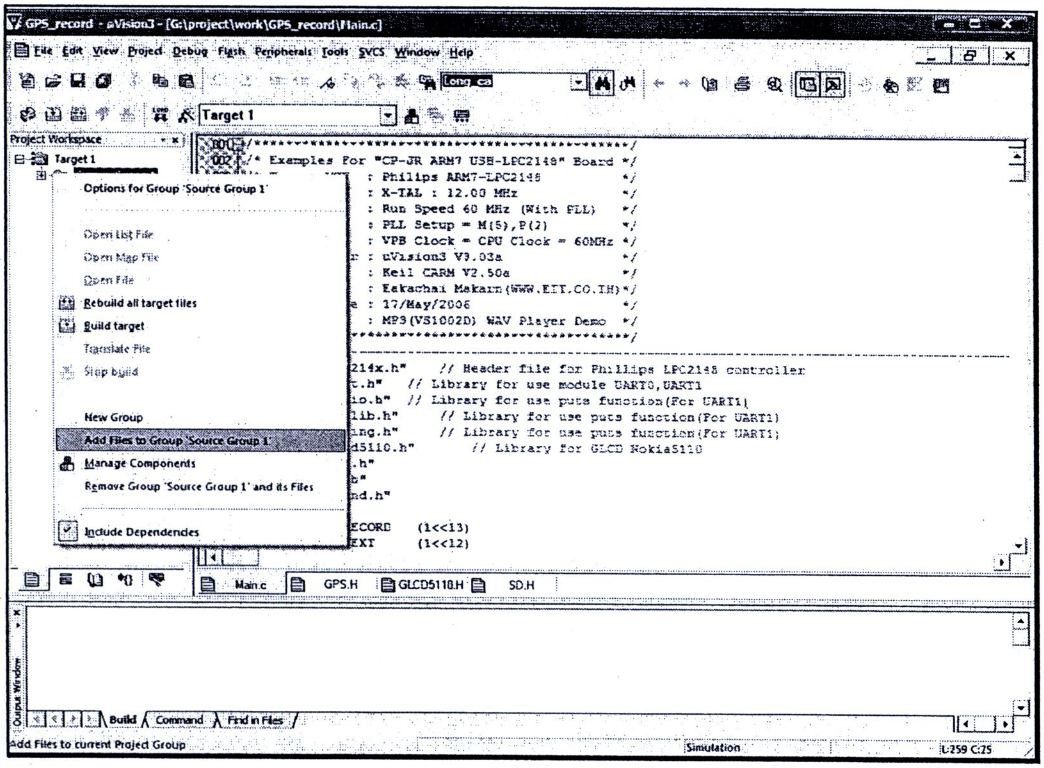
ในขั้นตอนนี้ให้ทำการพิมพ์ Source Code ภาษาซี ตามข้อกำหนดของ Keil-CARM ในพื้นที่เขียนโปรแกรมตามต้องต้องการ หลังจากพิมพ์คำสั่งภาษาซีเสร็จเรียบร้อยตามต้องการแล้ว ให้สั่ง Save ไฟล์ดังกล่าว โดยต้องกำหนดเป็นไฟล์ที่มีนามสกุลเป็น ".C" ในที่นี้ขอแนะนำให้สั่ง Save โดยใช้คำสั่ง File ➔ Save As... แล้วกำหนดชื่อและนามสกุลของไฟล์เป็น .main.c" ดังรูป



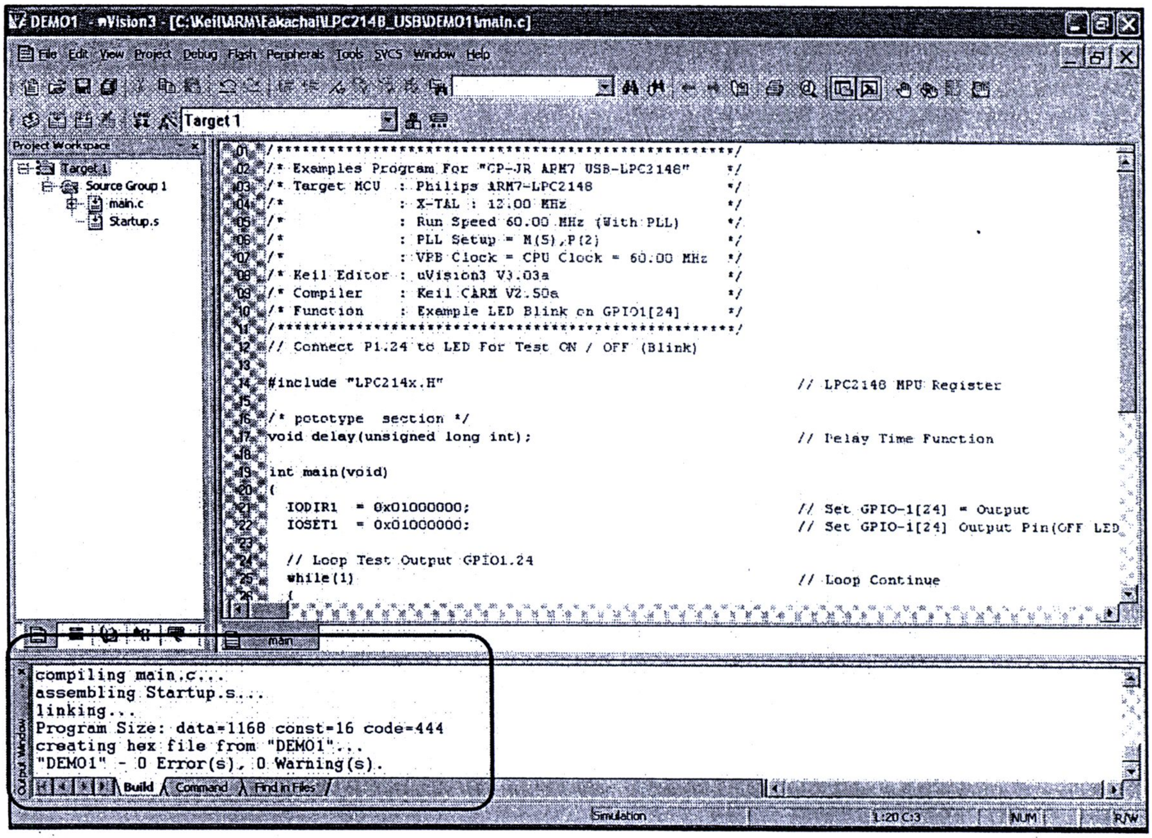
ซึ่งหลังจากที่สั่ง Save ไฟล์เป็น "main.c" แล้วจะเห็นว่าลักษณะสีของตัวอักขระต่างๆในโปรแกรมจะเกิดการเปลี่ยนแปลงไปตามหน้าที่ เช่น Comment, ตัวแปร และ คำสั่ง เป็นต้น ซึ่งส่วนนี้เป็นข้อดี

ของ Keil uVision3 ซึ่งสามารถแยกและแสดงตัวอักขระได้อย่างเป็นหมวดหมู่ ทำให้ง่ายต่อการอ่านโปรแกรมด้วย

7. ทำการสั่ง Add File ต่างๆเข้ากับ Project File โดยให้เลือกคลิกเมาส์ที่ด้านซ้ายของหน้าต่าง จากนั้นให้เลือกที่ Add Files to Group "Source Group 1 " แล้วเลือกที่ Add File ที่ต้องการจะเพิ่มเข้าไปใช้งานร่วมกับ Project Fileโดยในครั้งแรกให้เลือก Files of type เป็น "C Source files(*.c)" ซึ่งจะปรากฏชื่อไฟล์ต่างๆที่เป็น Source Code ภาษาซีให้เห็น โดยในที่นี้ให้เลือกคลิกเมาส์ที่ไอคอนของไฟล์ชื่อ "main.c" แล้วเลือก Add เพื่อสั่งเพิ่มไฟล์ชื่อ "Startup.s" เข้าไปรวมกับ Project Files ที่เราสร้างไว้

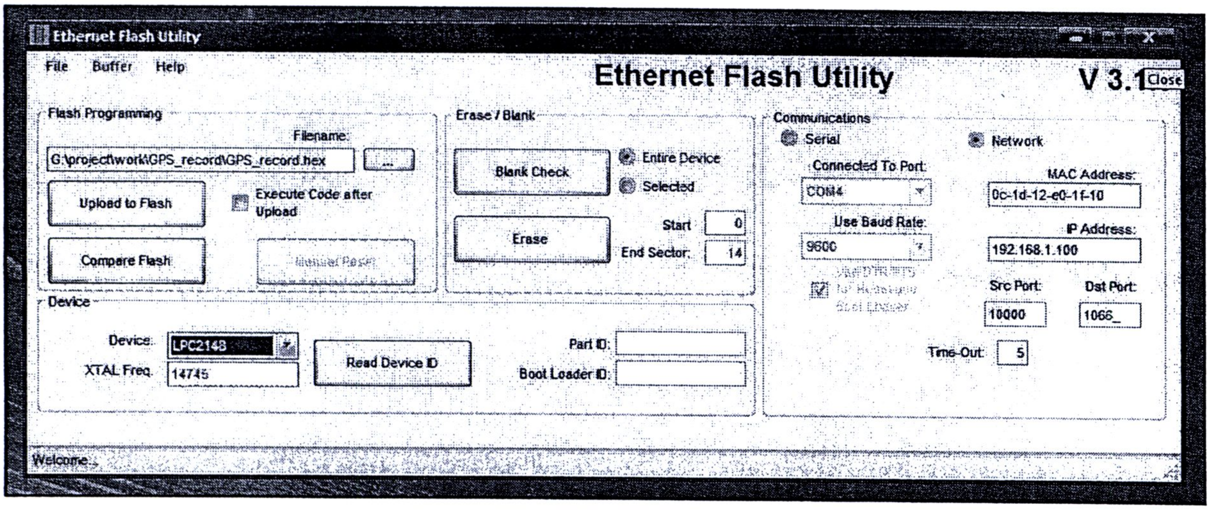


8. ให้ทำการสั่งแปลโปรแกรมที่เราเขียนขึ้นเรียบร้อยแล้ว โดยให้คลิกเมาส์ที่เมนูคำสั่ง Projects ➔ Rebuild all target files ซึ่งโปรแกรม Keil uVision3 จะทำการสั่งให้โปรแกรม Keil-CARM ทำการแปลคำสั่งให้ทันที ซึ่งหลังจากสั่งแปลโปรแกรมแล้วได้ผลถูกต้องและไม่เกิดข้อผิดพลาดใดๆขึ้น (0 Error และ 0 Warning) ก็ได้ Hex File ซึ่งมีชื่อเหมือนกันกับชื่อของ Project File ที่สร้างไว้ ซึ่งผู้ใช้สามารถนำ Hex File ดังกล่าวไปทำการ Download ให้กับ MCU ได้ทันที

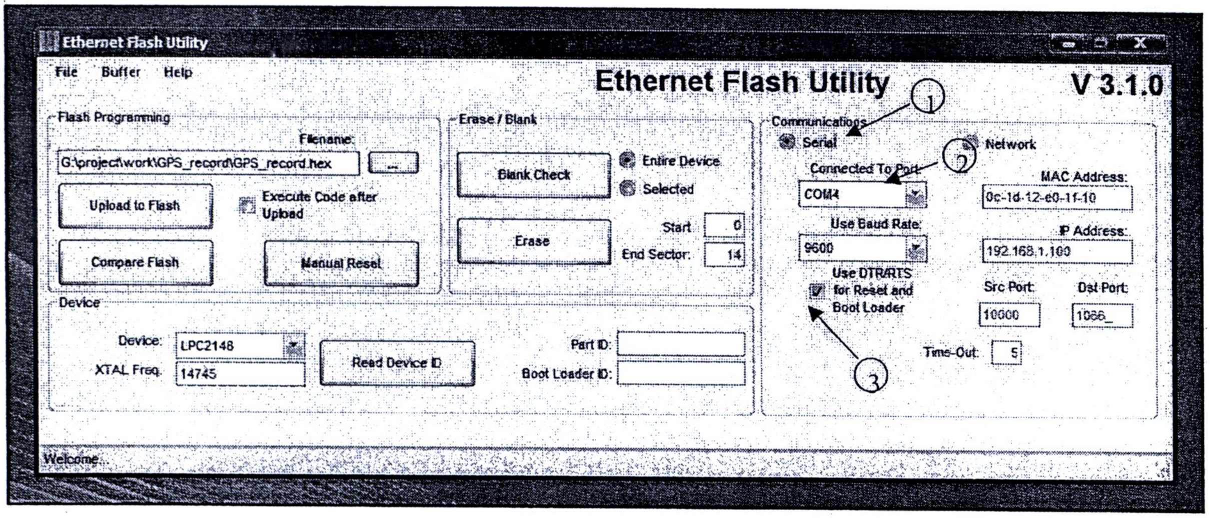

DEMO1 - µVision3 - [C:\Keil\ARM\Eakachai\LPC2148_USB\DEMO1\main.c]
File Edit View Project Debug Flash Peripherals Tools SVCS Window Help
Target 1
Project Workspace
Target 1
Source Group 1
main.c
Startup.s
/* Examples Program For "CP-JR ARM7 USB-LPC2148" */
/* Target MCU : Philips ARM7-LPC2148 */
/* : X-TAL : 12.00 MHz */
/* : Run Speed 60.00 MHz (With PLL) */
/* : PLL Setup = M(5),P(2) */
/* : VPB Clock = CPU Clock = 60.00 MHz */
/* Keil Editor : uVision3 V3.03a */
/* Compiler : Keil CARM V2.50a */
/* Function : Example LED Blink on GPIO1[24] */
// Connect P1.24 to LED For Test ON / OFF (Blink)
#include "LPC214x.H" // LPC2148 MPU Register
/* pototype section */
void delay(unsigned long int); // Delay Time Function
int main(void)
IODIR1 = 0x01000000; // Set GPIO-1[24] = Output
IOSET1 = 0x01000000; // Set GPIO-1[24] Output Pin(OFF LED
// Loop Test Output GPIO1.24
while(1) // Loop Continue
main
compiling main.c...
assembling Startup.s...
linking...
Program Size: data=1168 const=16 code=444
creating hex file from "DEMO1"...
"DEMO1" - 0 Error(s), 0 Warning(s).
Build
Command
Find in Files
Output Window
Simulation
L:20 C:3
NUM
R/W

การโหลดโปรแกรมทดสอบ

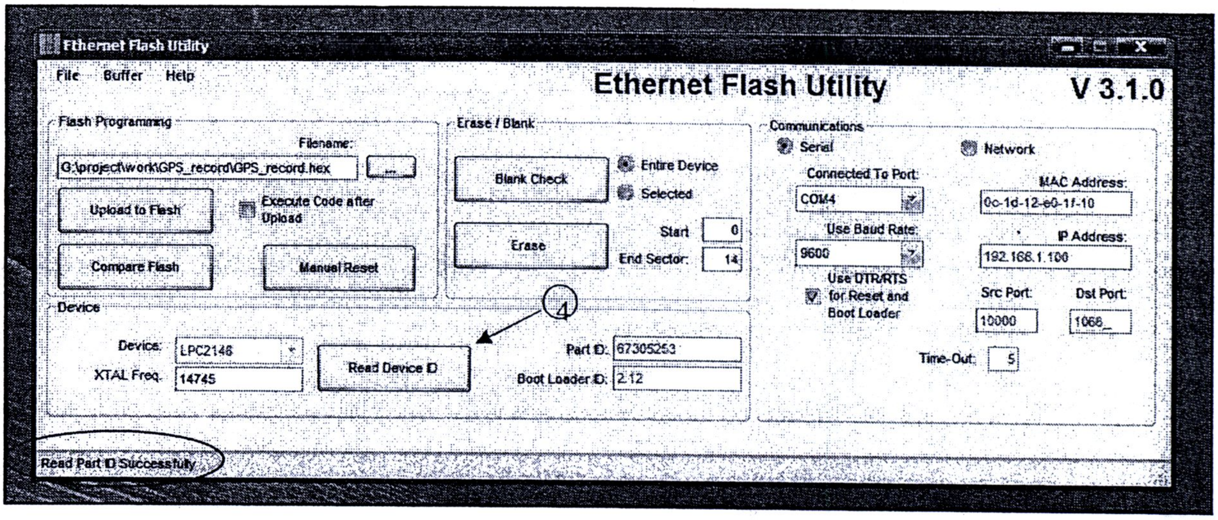
เปิดโปรกแกรม Ethernet Flash Utility จะได้หน้าต่างดังรูป



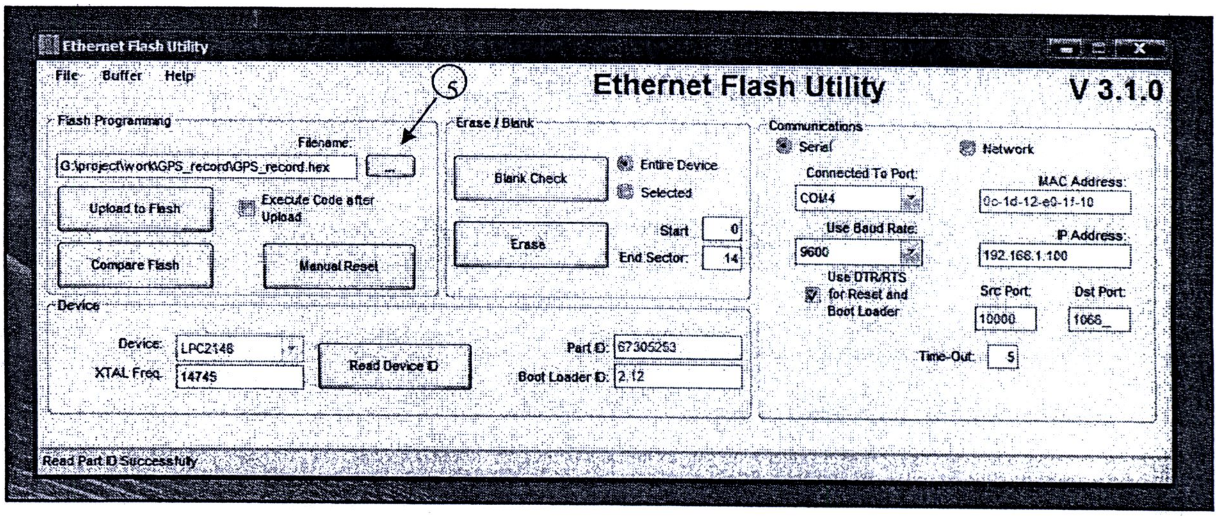
เลือกรูปแบบการเชื่อมต่อเป็นแบบ Serial ➔ เลือก COM ให้ถูกต้อง ➔ Baud Rate 9600 ➔ เลือกที่ช่อง Use DTR/RTS for Reset and Boot Loader

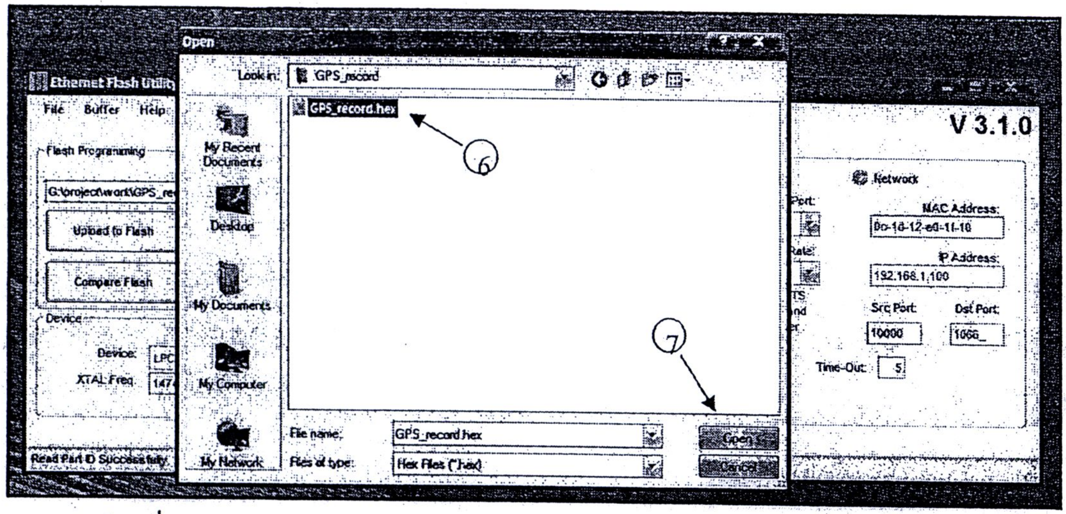


กด Switch Reset ที่ บอร์ดไมโครคอนโทรลเลอร์ แล้วเลือกที่ Read Device ID หากเชื่อมต่อกับไมโครคอนโทรลเลอร์สำเร็จจะแสดงข้อความ Read Part ID Successfully

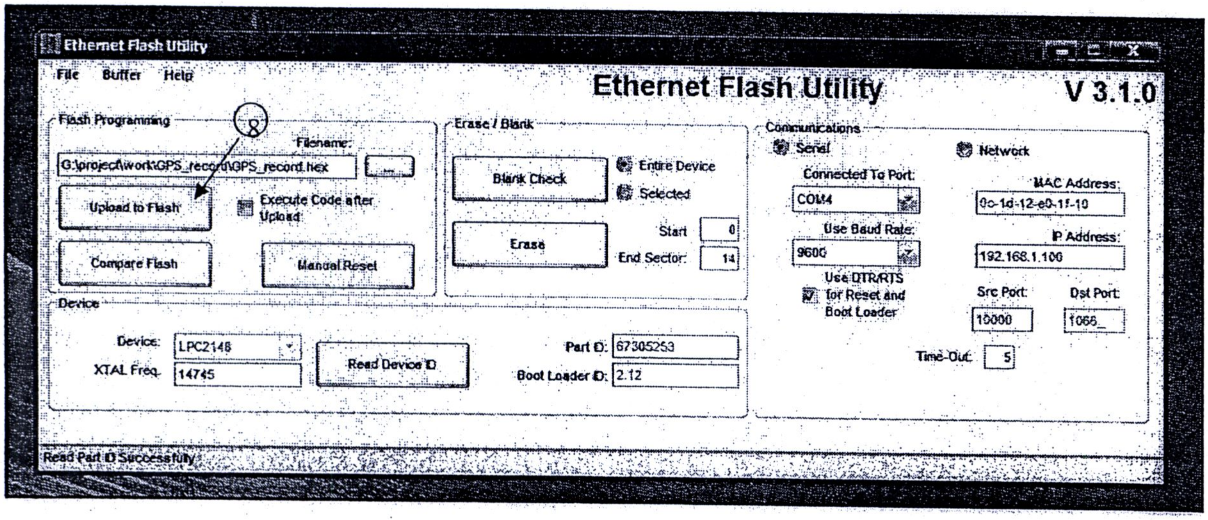


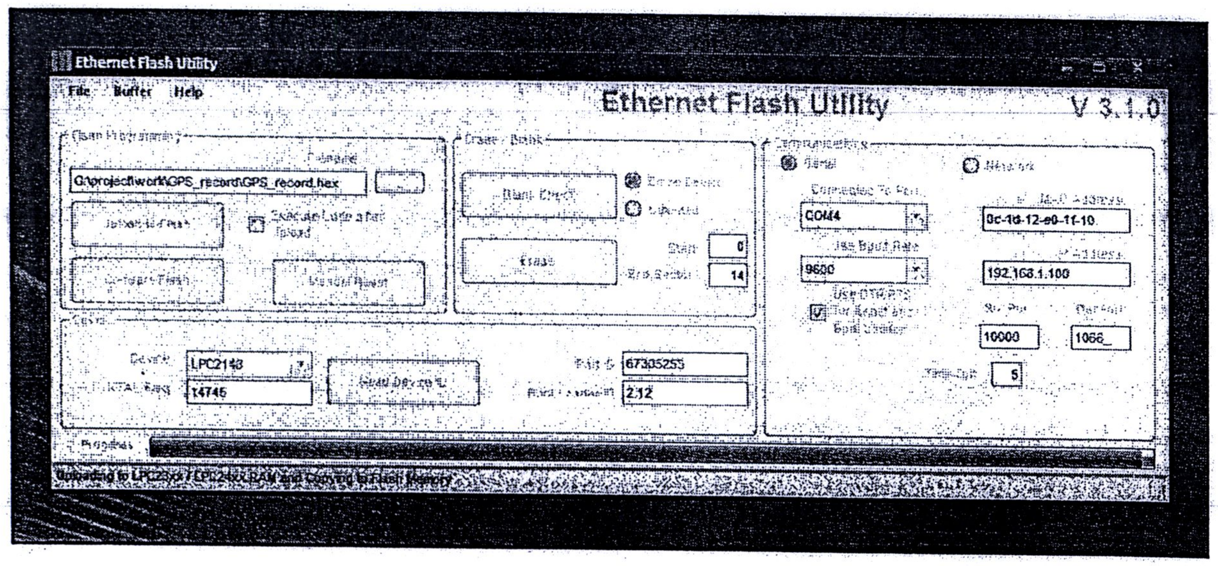
เมื่อเชื่อมต่อเรียบร้อยแล้ว ทำการโหลดไฟล์ .HEX โดยเลือกที่ Brown เลือกไฟล์ .HEX ที่ต้องการ แล้วเลือก Open ดังรูป





เลือกที่ Upload to Flash และรอจนกว่าโปรแกรมจะทำการโหลดเสร็จ ดังรูป





ประวัติผู้วิจัย

ผู้ช่วยศาสตราจารย์ ดร. รังสรรค์ ทองทา สำเร็จการศึกษาวิศวกรรมไฟฟ้า จากมหาวิทยาลัยขอนแก่น เมื่อปี พ.ศ. 2531 และ MS และ PH.D. จาก Florida Institute of Technology, USA เมื่อปี 1993 และ 1998 ตามลำดับ เริ่มงานในตำแหน่งวิศวกรที่บริษัท Elcom Research เมื่อปี พ.ศ. 2531 และบริษัท การบินไทย จำกัด ในปี พ.ศ. 2532 เมื่อปี พ.ศ. 2533 ได้เริ่มในตำแหน่งอาจารย์ที่ภาควิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยขอนแก่น และมหาวิทยาลัยเทคโนโลยีสุรนารีในปี พ.ศ. 2536

สำนักงานคณะกรรมการวิจัยแห่งชาติ
ห้องสมุดงานวิจัย