

myCortex-LM308/608/808



11/16/2008

ARM Cortex-M3 코어를 사용한
LM3S308/LM3S608/LM3S808 개발 보드

32 비트 ARM Cortex-M3 코어를 내장한 Luminary Micro 사의 LM3S308/LM3S608/LM3S808 프로세서는 내부에 10 비트 ADC 8 채널, UART 2 채널, SPI 1 채널, i2c 1 채널, 32 비트 타이머 3 개를 내장하고 최대 50MHz 로 동작합니다. myCortex-LMx08 보드는 LM3S308/LM3S608/LM3S808 프로세서 개발을 보다 편리하게 진행할 수 있도록 도와드립니다. 8 비트 AVR 과 유사한 가격대로 32 비트 프로세서를 사용해 보시기 바랍니다.

myCortex-LM308/608/808

제품 소개

임베디드 프로세서 코어 설계 업체로 유명한 ARM 사에서 ARM7/ARM9/ARM11 으로 이어지는 개발 라인에서 새로운 신형 아키텍처를 추가하였습니다. 코어는 ARMv7 이며 제품군은 Cortex(코어텍스)라고 명명하였습니다. 코어텍스 단어를 사전에서 찾아보면 인간 뇌의 대뇌피질로 설명되어 있습니다. 인간의 모든 지능적 행위가 대뇌 피질에서 이뤄지듯이 ARM 사에서는 자사의 마이크로프로세서가 시스템에서 대뇌에 해당하는 지능적인 장치가 되길 바라는 마음에서 제품명을 지었습니다.

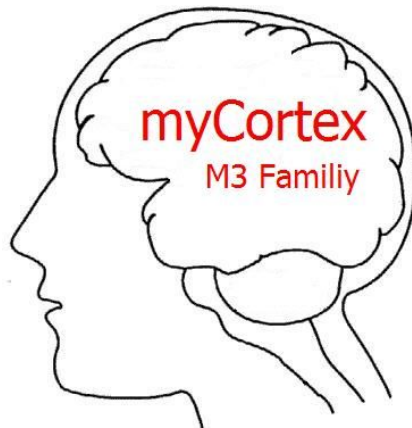


그림 1 위드로봇의 myCortex M3 계열 로고

미국 텍사스주에 위치한 Luminary Micro 사에서는 ARM 의 Cortex-M3 코어를 이용한 다양한 프로세서를 출시하고 있으며 제품명을 Stellaris 라고 부릅니다. Stellaris 300/600/800 계열은 44 핀 패키지에 각종 부가 기능을 가지고 있으면서 디바이스 가격은 8 비트 AVR 프로세서와 유사하여 기존 8 비트 시장을 대치하고자 출시된 모델입니다. 가격, 메모리 용량, 부가 기능은 유사하면서 32 비트로 최대 50MHz 까지 동작이 가능한 Stellaris 300/600/800 계열은 기존의 AVR 로 대표되던 8 비트 마이컴 시장을 대치할 것으로 기대됩니다.

모델 별 특징

myCortex-LMx08 에는 총 세 종류의 모델(LM308, LM608, LM808)이 준비되어 있습니다. 동일한 핀 레이아웃에 내부 메모리 및 동작 속도가 다르므로, 필요에 따라 하드웨어

변경 없이 적절히 선택하실 수 있습니다. Luminary Micro 의 LM3S308, LM3S608, LM3S808 프로세서를 사용한 모델로 각각 myCortex-LM308, myCortex-LM608, myCortex-LM808 로 모델명을 부르고 있습니다.

기능	myCortex-LM308	myCortex-LM608	myCortex-LM808
동작 속도	25MHz	50MHz	좌동
플래시/램	16KB/4KB	32KB/8KB	64KB/8KB
ADC	8 채널	좌동	좌동
UART	2 채널	좌동	좌동
SPI	1 채널	좌동	좌동
i2c	1 채널	좌동	좌동
타이머	32 비트 3 개	좌동	좌동

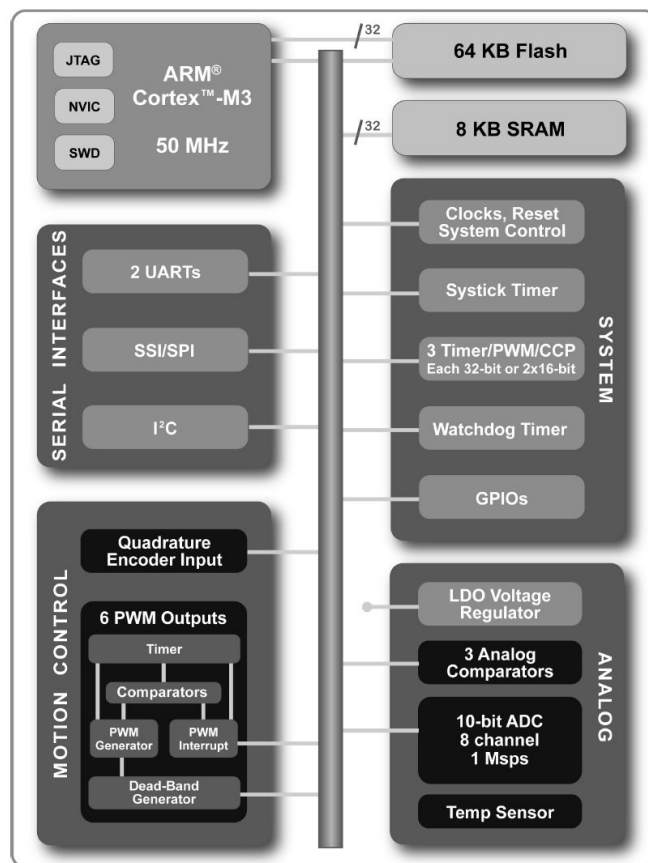


그림 2 Stellaris 800 계열 블록 다이어그램

응용 분야

- 마이크로 프로세서 응용 전분야
- 8 비트 마이컴 이상의 성능이 필요한 프로젝트
- ARM Cortex 학습

기타 정보 및 관련 제품

<http://www.withrobot.com/category/Microcontroller> 에 접속하면 myCortex-LMx08 I 이외에 다양한 마이크로 프로세서 응용 제품에 관한 정보를 보실 수 있습니다. 또한 myCortex-LMx08 제품들과 관련하여 데이터시트, 관련 예제 프로그램도 제공하고 있습니다. 개발 환경은 myCortex-LMx08 과 연결하는 [Stellaris-JTAG](#) 항목을 참고하시기 바랍니다.

Withrobot Lab

[검색]

| 소개 | 팀 구성 | 오시는 방법 | 수행과제 | 제품군 소개 | 구매방법 | 개발응역의회 | 오프라인 강의 | 기술문서

Home

문의 게시판

- ☑ 분류 전체보기 (58)
 - ☐ 공지 (7)
 - ☐ Image Processing (14)
 - ☐ Microcontroller (2)
 - ☐ Motion Control (0)
 - ☐ Interface (7)
 - ☐ Sensors (7)
 - ☐ System (0)
 - ☐ Projects (4)
 - ☐ Technical Reports (15)
 - ☐ Lecture (0)
 - ☐ Museum (2)

Tags

myCortex-LM808 지도 시리얼 통신 오
시는 방법 MRreader 영상처리 real
time vision system ARM
Robaa-Checker color tracking 바코드
수행과제 소개 Cortex-M3 임베디드
구매방법 Color-Eye Liminary Micro usb
myVision USB Vision MRcode Image
Processing Smart-I 2D barcode
myCortex-LM608 UART Project
myCortex-LMx08 myCortex-LM308

Recent Entries

myCortex-LM308/608/808.

[MRreader] Smart-I V1.0을 이용한 2D

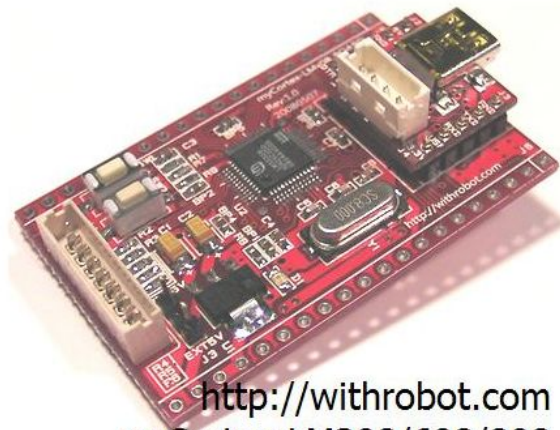
'Microcontroller'에 해당되는 최근 글 2건

1. 02:54:20 myCortex-LM308/608/808
2. 2008/03/19 myCortex-LM8962 (4)

myCortex-LM308/608/808

Microcontroller/myCortex-LMx08 2008/05/20 02:54

Modify : Modify(new windows) : (공개)→비공개로 변경합니다 : Trackback : Delete



<http://withrobot.com>
myCortex-LM308/608/808

myCortex-LMx08 보드는 ARM Cortex-M3 코어를 사용한 소형 개발 보드입니다. 32비트의 특성을 가지고 있으면서도 AVR 가

그림 3. 다양한 마이크로프로세서 응용 보드에 관한 정보가 있는 withrobot 홈페이지(<http://www.withrobot.com/category/Microcontroller>)

외형 치수 및 보드 설명

외형 치수

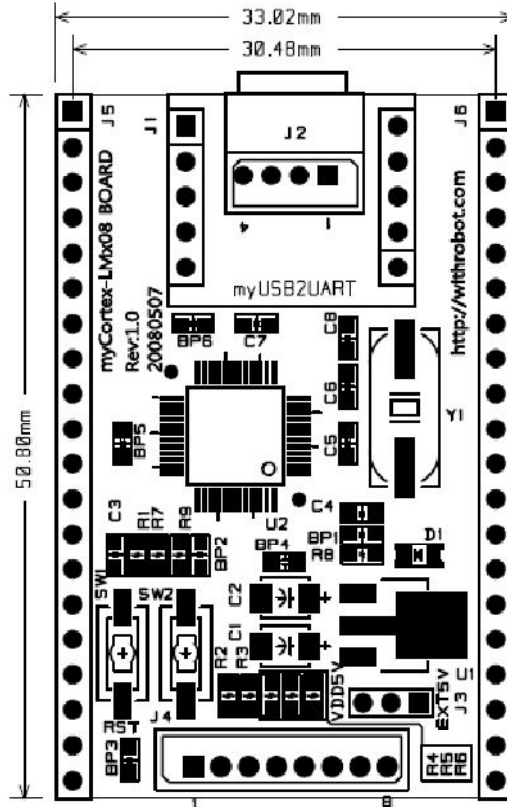


그림 4. myCortex-LMx08 외형 치수

보드 양측에 100mil(2.54mm) 간격으로 홀이 배치되어 있어 헤더핀을 이용하면 일반 공 PCB 에 손쉽게 연결하여 이용하실 수 있습니다. 필요에 따라서 헤더핀 또는 헤더핀 소켓을 연결해서 다른 PCB 와 연결하거나 케이블로 다른 마이크로프로세서와 연결하여 사용하시기 바랍니다. 사용자 기호에 맞게 사용할 수 있도록 인터페이스 헤더핀은 연결되지 않은 상태로 제공됩니다.

인터페이스 커넥터 설명

J5	CPU	신호	J6	CPU	신호
1	21	PA4/SSIRX	1	20	PA3/SSIFSS
2	22	PA5/SSITX	2	19	PA2/SSICLK
3	25	PD0	3	18	PA1/U0TX
4	26	PD1	4	17	PA0/U0RX

5	27	PD2/U1RX	5	14	PC4
6	28	PD3/U1TX	6	13	PC5/CCP1
7	29	PB0/CCP0	7	12	PC6/CCP3
8	30	PB1/CCP2	8	11	PC7/CCP4
9	33	PB2/I2CSCL	9	1	ADC0
10	34	PB3/I2CSDA	10	2	ADC1
11	44	PB4/C0-	11	3	ADC2
12	43	PB5/CCP5	12	4	ADC3
13	42	PB6/C0+	13	48	ADC4
14	41	PB7/TRST	14	47	ADC5
15	35	PE0	15	46	ADC6
16	36	PE1	16	45	ADC7
17	40	TCK	17	-	VDD(3.3V out)
18	39	TMS	18	-	VDD(3.3V out)
19	38	TDI	19	-	EXTVDD5V
20	37	TDO	20	-	GND

표 1. myCortex-LMx08 보드의 인터페이스 핀 배치

EXTVDD5V 는 보드를 동작시키기 위해 외부에서 5V 전압을 인가하는 핀입니다. myCortex-LMx08 보드는 외부에서 보드 전압을 인가해서 동작시킬 수도 있으며, USB 파워로도 동작시킬 수 있습니다. J6 의 EXTVDD5V 에 전압을 인가하여 보드를 동작시키는 경우에는 점퍼 J3 을 오른쪽으로 연결하신 후 사용하시면 됩니다.

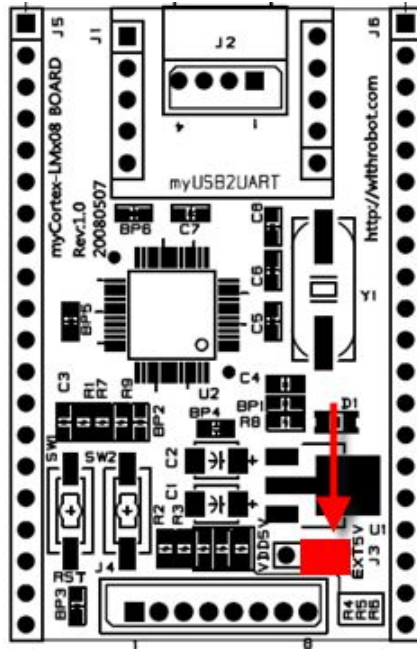


그림 5 J6 19 번 핀(EXTVDD5V)로 전원을 인가할 경우 J2 점퍼 설정

J2 커넥터 또는 J1 을 이용하여 myUSB2UART 로 전원을 인가하는 경우에는 점퍼 J3 을 위쪽으로 연결하신 후 사용하시면 됩니다. 공장 출하 상태는 점퍼는 VDD5V 쪽에 연결되어 있습니다.

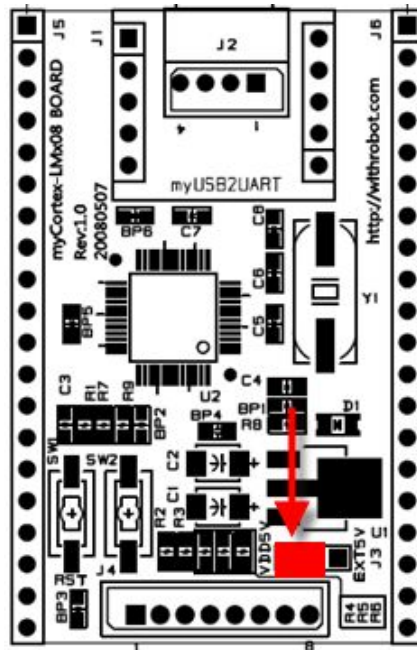
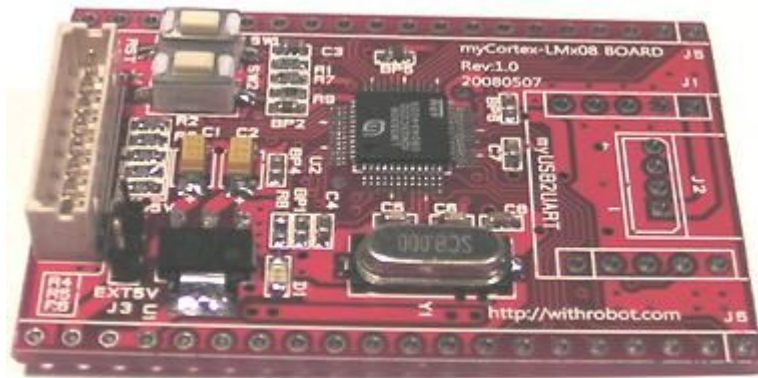


그림 6 J1 또는 J2 를 통해 전원을 인가할 경우 J2 점퍼 설정

UART 커넥터 연결 방법

myCortex-LMx08 보드에는 UART0 를 다양한 방법으로 연결할 수 있도록 준비해 두었습니다. 아래 보드 사진에서 우측의 공백 공간 J1, J2 가 배치된 곳이 UART0 를 연결할 수 있는 커넥터를 연결하는 곳입니다. 필요에 따라 4 핀 스트레이트 타입의 커넥터(53014-04) 또는 앵글 타입의 커넥터(53015-04)를 연결하여 사용할 수 있습니다. USB 에 바로 연결하여 테스트하고 싶으신 경우는 [myUSB2UART\(별매\)](#)를 이용하여 손쉽게 기능을 확장할 수 있습니다. 공장 출시 기본 설정은 스트레이트 타입 커넥터를 부착해 두었습니다. 필요에 따라 이 커넥터를 제거하시고 적용 프로젝트에 따라 적합한 형태를 선택하여 사용하시기 바랍니다.



myCortex-LMx08 보드의 UART0 연결 커넥터 부분(보드의 우측)

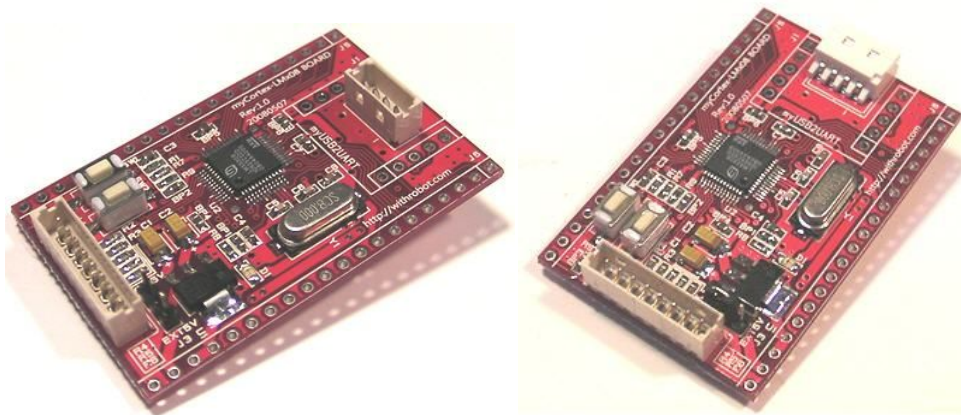


그림 7 스트레이트 타입의 커넥터 적용(좌측)과 앵글 타입 커넥터를 적용(우측)한 경우
공장 출하 시 좌측 형태로 출시



그림 8 myUSB2UART 를 myCortex-LMx08 보드에 바로 연결한 모습
4 핀 커넥터는 제거하고 myUSB2UART 를 연결

사용 칩셋 소개

- myCortex-LM808: Luminary Micro 사의 LM3S808 프로세서
- myCortex-LM608: Luminary Micro 사의 LM3S608 프로세서
- myCortex-LM308: Luminary Micro 사의 LM3S308 프로세서

STELLARISWARE

StellarisWare 는 Luminary Micro 에서 제공하는 각종 라이브러리 및 예제 모음으로, 모든 Stellaris 시리즈 프로세서에 사용할 수 있습니다. StellarisWare 의 API 함수를 이용하면 Stellaris 시리즈 프로세서의 내부 H/W 와 peripheral 구성에 대해 잘 모르는 개발자도 쉽게 Stellaris 시리즈를 다룰 수 있는 강력한 라이브러리입니다. myCortex-LMx08 프로세서 보드에서도 StellarisWare 의 peripheral 라이브러리들을 사용할 수 있습니다.

StellarisWare 는 Luminary Micro 의 홈페이지나 위드로봇 홈페이지에서 다운로드 받으실 수 있습니다.

- http://www.luminarymicro.com/products/software_updates.html
- <http://withrobot.com/entry/myCortex-LM808>

StellarisWare 를 다운로드 받으신 다음 더블 클릭하여 실행시키면 설치가 진행됩니다. StellarisWare 에는 여러 Stellaris 칩셋들을 위한 예제 소스 코드들도 풍부하게 포함되어 있습니다. 각 예제의 소스 코드를 참고하면 여러 가지 개발과 학습에 많은 도움이

됩니다. 참고로 여기에서 말하는 Example Applications 들은 Luminary Micro 에서 제공하는 예제들이며, myCortex-LMnnnn 시리즈 보드에 바로 사용할 수 있는 예제는 아닙니다. Withrobot lab 에서는 myCortex-LMnnnn 시리즈 프로세서 보드를 위한 예제들을 별도로 제공하고 있습니다. 자세한 내용은 다음 절을 참고하시기 바랍니다.

myCortex-LMx08 보드는 StellarisWare 를 그대로 사용할 수 있습니다. 다만 다음 절에서 설명 드리는 예제 소스를 정상적으로 사용하기 위해서는 **StellarisWare 의 일부 내용을 수정**해야 합니다.

bl_check.c 파일 수정

boot_loader 소스 중 bl_check.c 파일(StellarisWare/boot_loader/bl_check.c)은 부트로더가 처음 부팅될 때 다운로드 대기상태로 전환할 것인지 사용자 펌웨어를 실행시킬 것인지를 결정하는 부분의 코드입니다. 이 소스에 문제가 조금 있어 스위치를 이용해 모드를 선택하는 것이 잘 동작하지 않습니다. 이 문제를 해결하기 위해 원래 bl_check.c 파일을 아래와 같이 수정하도록 합니다.

```
#include "bl_check.h"
#include "bl_config.h"

#if 1 // withrobot Co.
extern void Delay(unsigned long ulCount);
#define FORCED_UPDATE_PIN_MASK (1 << FORCED_UPDATE_PIN)
#endif

//*****
//
//!! \addtogroup bl_check_api
```

먼저 bl_check.c 파일의 앞부분에 위와 같이 4 줄을 삽입합니다.

```
#ifdef ENABLE_UPDATE_CHECK
//
// Enable the required GPIO module.
//
HWREG(SYSCTL_RCGC2) = FORCED_UPDATE_PERIPH;
#if 1 // withrobot Co.
Delay(10);
HWREG(FORCED_UPDATE_PORT + GPIO_O_DR2R) |= FORCED_UPDATE_PIN_MASK;
HWREG(FORCED_UPDATE_PORT + GPIO_O_PUR) |= FORCED_UPDATE_PIN_MASK;
HWREG(FORCED_UPDATE_PORT + GPIO_O_DEN) |= FORCED_UPDATE_PIN_MASK;
#endif
#endif
```

ENABLE_UPDATE_CHECK 단어를 검색하여 첫 번째 발견되는 코드에 위와 같이 여섯 줄을 삽입합니다.

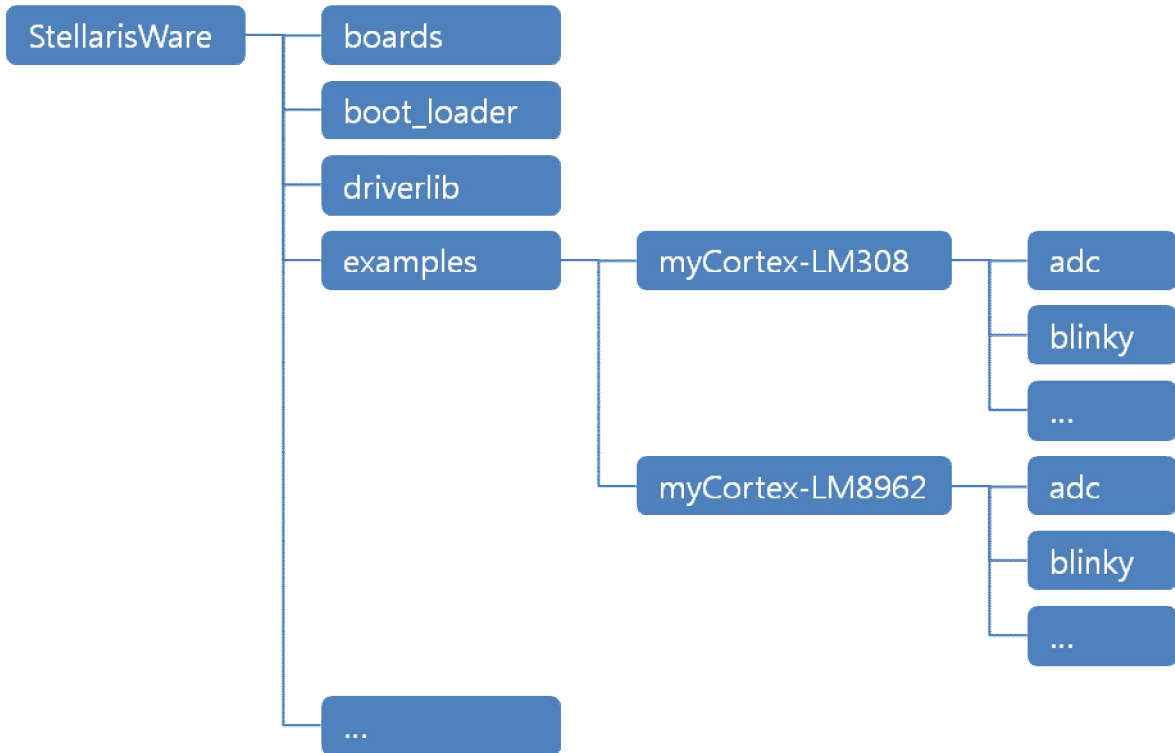
```
#ifndef ENABLE_UPDATE_CHECK
//
// Check the pin to see if an update is being requested.
//
#if 0 // withrobot Co.
HWREG(FORCED_UPDATE_PORT + GPIO_O_DEN) |= 1 << FORCED_UPDATE_PIN;
#endif
if (HWREG(FORCED_UPDATE_PORT + (1 << (FORCED_UPDATE_PIN + 2))) ==
    (FORCED_UPDATE_POLARITY << FORCED_UPDATE_PIN))
{
    return(1);
}
#endif
```

ENABLE_UPDATE_CHECK 가 두 번째로 발견되는 부분의 코드에 위와 같이 #if 0 ~ #endif 로 한 줄의 코드를 막습니다.

개발 예제 소스

위드로봇에서는 myCortex-LMx08 보드를 이용한 개발에 참고할 수 있도록 개발 예제 소스 코드를 함께 제공합니다. 예제 소스 코드는 위드로봇 홈페이지(<http://withrobot.com/entry/myCortex-LM808>)에서 다운로드 받으실 수 있습니다.

다운로드 받은 예제는 압축을 풀어 그대로 사용하게 됩니다. 압축을 푸는 폴더는 StellarisWare 가 설치된 폴더 아래에 다음과 같이 만드는 것을 추천합니다.



만일 다른 폴더에 설치하신 경우에는 Makefile.def 파일의 내용 중 ROOT 변수의 값을 StellarisWare 가 설치된 폴더까지의 상대 경로명으로 설정하면 됩니다. 단 **이 경우에는 위드로봇 제작 예제만 정상 동작하고, Luminary Micro 제작 예제는 동작하지 않습니다.**

myCortex-LMx08 예제는 위드로봇에서 제작한 예제와 Luminary Micro 에서 제작한 예제로 구성됩니다. Luminary Micro 에서 제작한 예제는 DriverLib 에 포함된 예제 소스를 myCortex-LMx08 보드에서 동작 가능하도록 수정한 것입니다.

모든 예제는 Makefile 을 가지고 있습니다. 각 예제 폴더 내에서 make 를 수행하면 컴파일을 수행하고, 그 결과물은 gcc 폴더 아래에 *.axf, *.bin 형태로 만들어집니다. *.axf 파일은 바이너리 이미지와 디버그 정보를 포함한 파일로써 JTAG 장비를 이용하여 디버깅을 할 때 사용하게 됩니다. *.bin 파일은 펌웨어 바이너리 파일이며, 부트 로더나 JTAG 장비를 이용하여 보드의 플래시 메모리에 다운 로드 할 때 사용합니다. 위드로봇에서는 myCortex 시리즈 프로세서 보드를 위해 Stellaris-JTAG 을 개발, 판매하고 있습니다. <http://www.withrobot.com/category/Interface/Stellaris-JTAG> 에서 관련 내용을 보실 수 있으며, myCortex-LMx08 시리즈용 예제를 다운로드하고 소스 디버깅 하는 방법에 대해서는 Stellaris-JTAG 의 문서를 참고하시기 바랍니다. JTAG 장비가 없는 경우를 위해서 myCortex 시리즈 보드는 부트로더를 탑재하고 출시됩니다. 부트로더를 이용한 펌웨어 다운로드 방법은 다음 절을 참고하시기 바랍니다.

부트 로더

시리얼 부트로더와 테스트 프로그램

위드로봇에서는 보드의 전수 검사를 수행하면서 내부 플래시 메모리에 부트로더와 기본 예제 코드(timer 예제)를 기록해 두었습니다. 따라서 구입한 보드에 최초로 전원을 인가하면 플래시에 프로그래밍된 예제 프로그램이 동작하면서 LED 가 깜박이는 것을 확인할 수 있습니다. 추가로 내장한 부트로더를 이용하면 시리얼 포트로 프로그램을 플래시에 손쉽게 기록할 수 있습니다.

시리얼 부트로더의 사용법

시리얼 부트로더를 사용하여 펌웨어를 다운로드하려면 PC 에 LM Flash Programmer 를 설치해야 합니다. LM Flash Programmer 는 아래 웹페이지에서 다운로드 받을 수 있습니다.

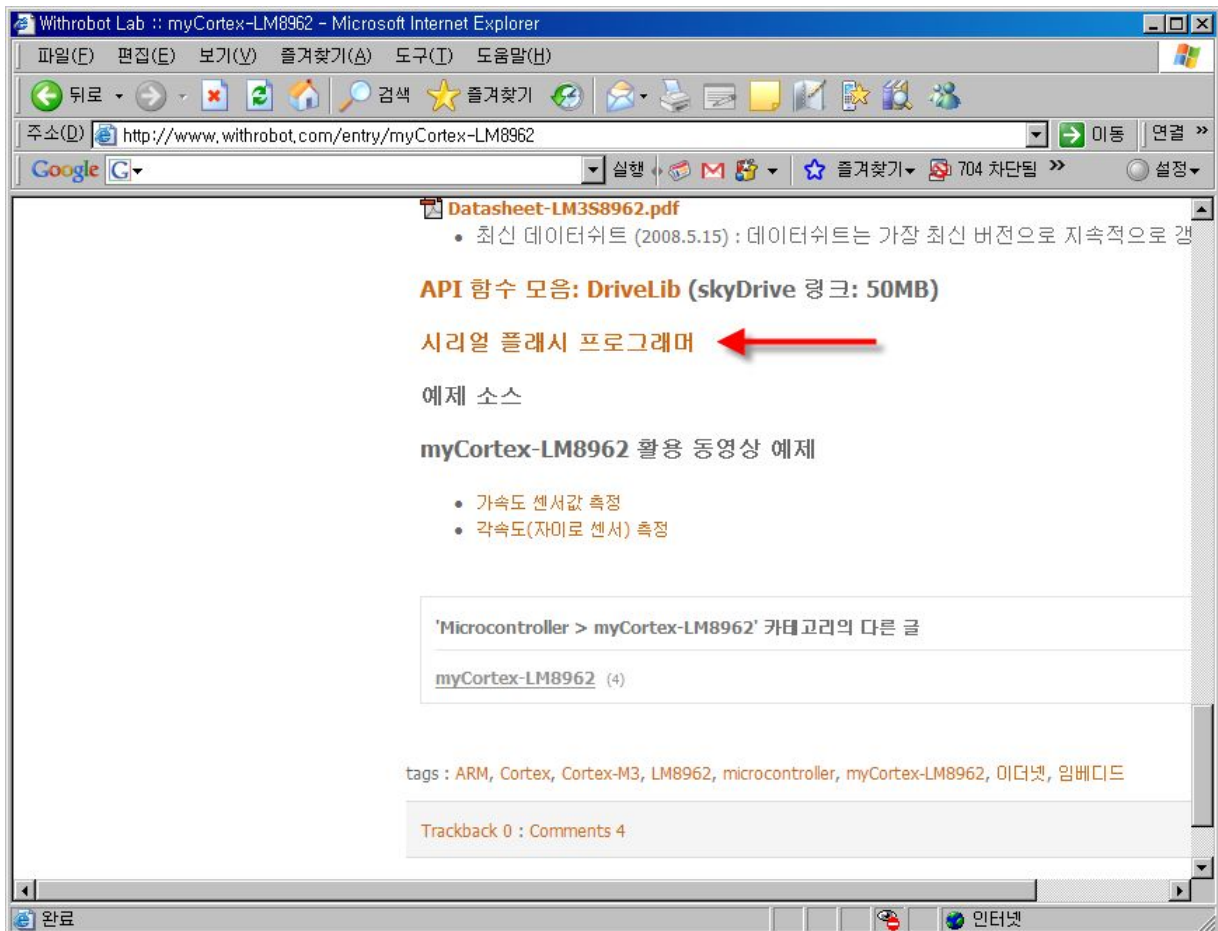


그림 9 <http://www.withrobot.com/entry/myCortex-LM808> 에서

시리얼 플래시 프로그래머를 내려 받을 수 있습니다.

다운로드 받은 파일을 실행하면 다음과 같은 과정을 거쳐 PC 에 설치하게 됩니다.

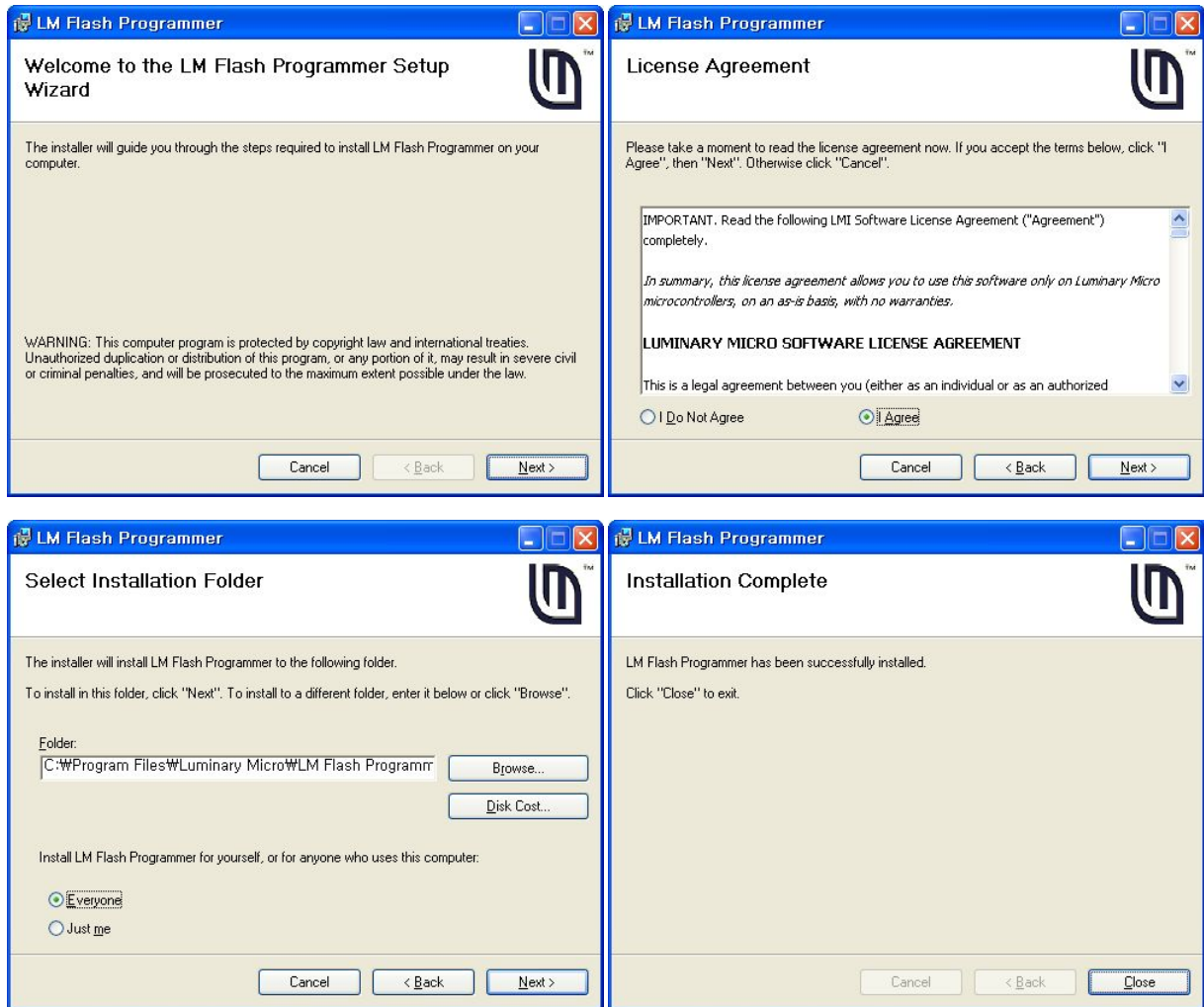


그림 10 LM Flash Programmer 설치 과정

설치가 완료되면 시작 메뉴에 LM Flash Programmer 항목이 새로 만들어집니다.

설치한 LM Flash Programmer 를 실행하면 아래의 그림 11 과 같은 화면을 볼 수 있습니다. Configuration 탭에서 Quick Set 항목은 Manual Configuration – see below 를 선택해야 합니다. Interface 를 Serial(UART) Interface 로 선택하고, Baud Rate 는 115,200 을 선택합니다. 또한 Disable Auto Baud Support 체크박스에도 체크하도록 합니다. COM Port 는 myUSB2UART 혹은 Stellaris-JTAG 의 UART 포트 번호를 선택하면 됩니다. 포트 번호를 확인하고자 할 때에는 Device Manager 버튼을 클릭해서 장치 관리자를 실행하면 확인할 수 있습니다.



그림 11 LM Flash Programmer 설정

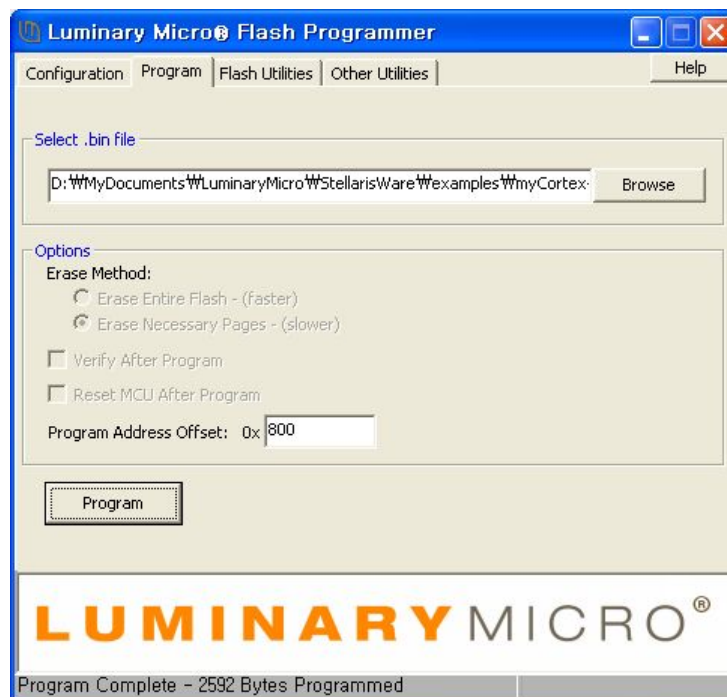


그림 12 LM Flash Programmer 프로그래밍

Configuration 탭의 설정을 마친 후 Program 탭으로 전환하면 그림 12 와 같은 화면을 볼 수 있습니다. 이 화면에서 다운로드 하고자 하는 바이너리를 선택하고 부트로더를 경유하여 플래시 메모리에 다운로드 할 수 있습니다.

Select .bin file 항목에 다운로드 하고자 하는 바이너리 파일의 경로와 이름을 입력합니다. Program Address Offset 은 800 을 입력합니다.

Program Address Offset 은 다운로드 할 이미지가 플래시 메모리의 어느 주소에 저장될 것인지를 알려주기 위한 입력 항목입니다. myCortex-LMnnnn 프로세서 보드에 탑재되어있는 부트로더는 자체 크기가 2KB 보다 작으며, 플래시 메모리의 0 번지부터 기록되어있습니다. 따라서 사용자 프로그램은 2KB, 즉 0x800 주소부터 플래시에 기록되어야 합니다.

이제 myCortex-LMx08 프로세서 보드를 UART 를 통하여 PC 와 연결하고 SW2 를 누른 상태에서 RST(SW1)버튼을 짧게 눌렀다 때도록 합니다. 이처럼 PC7 에 low 가 인가된 상태에서 리셋이 되면 탑재되어있는 부트로더가 기동되면서 프로그램 다운로드 대기 상태로 진입합니다.

다운로드 대기상태에서 LM Flash Programmer 의 Program 버튼을 클릭하면 프로그램이 다운로드 되면서 플래시 메모리에 저장되며, 진행 상태는 LM Flash Programmer 의 하단 상태표시줄에 진행막대로 표시됩니다.

프로그램 다운로드가 완료되면 myCortex-LMx08 프로세서 보드의 RST 버튼을 짧게 눌러서 리셋 시켜 방금 다운로드 한 펌웨어가 실행되도록 하면 됩니다.

개발 장치(STELLARIS-JTAG)

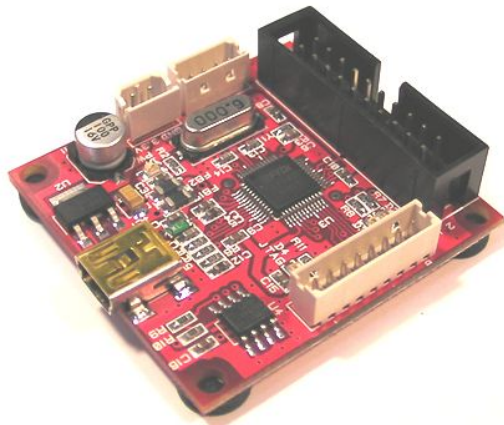


그림 13 Stellaris-JTAG

위드로봇에서 제작한 myCortex-LM 시리즈의 개발을 지원하기 위해 플래시 프로그래밍이 가능하고 소스 레벨 디버깅이 가능한 USB JTAG 을 Stellaris-JTAG 이라는

이름으로 판매합니다. 기존의 수 백만원 넘는 고가의 JTAG 에 비하면 여러 가지 제한점이 많지만 Stellaris 제품군을 개발하는 데에는 큰 불편함이 없어 보다 빠르게 디버깅을 수행하실 수 있습니다.

Stellaris-JTAG 을 이용하면 오픈 소스들의 조합으로 추가 비용 지출 없이 편리한 개발환경을 구축할 수 있습니다.

위드로봇에서 제안하는 개발환경 구성은 다음과 같습니다.

- 통합 에디터: 이클립스(Eclipse)
- 컴파일러: CodeSourcery G++ Lite
- 디버거: GDB + OpenOCD
- JTAG: Stellaris-JTAG

제안한 방식으로 시스템을 구성하는 경우 JTAG 하드웨어를 제외한 나머지 소프트웨어 부분은 모두 무상으로 이용하실 수 있습니다.

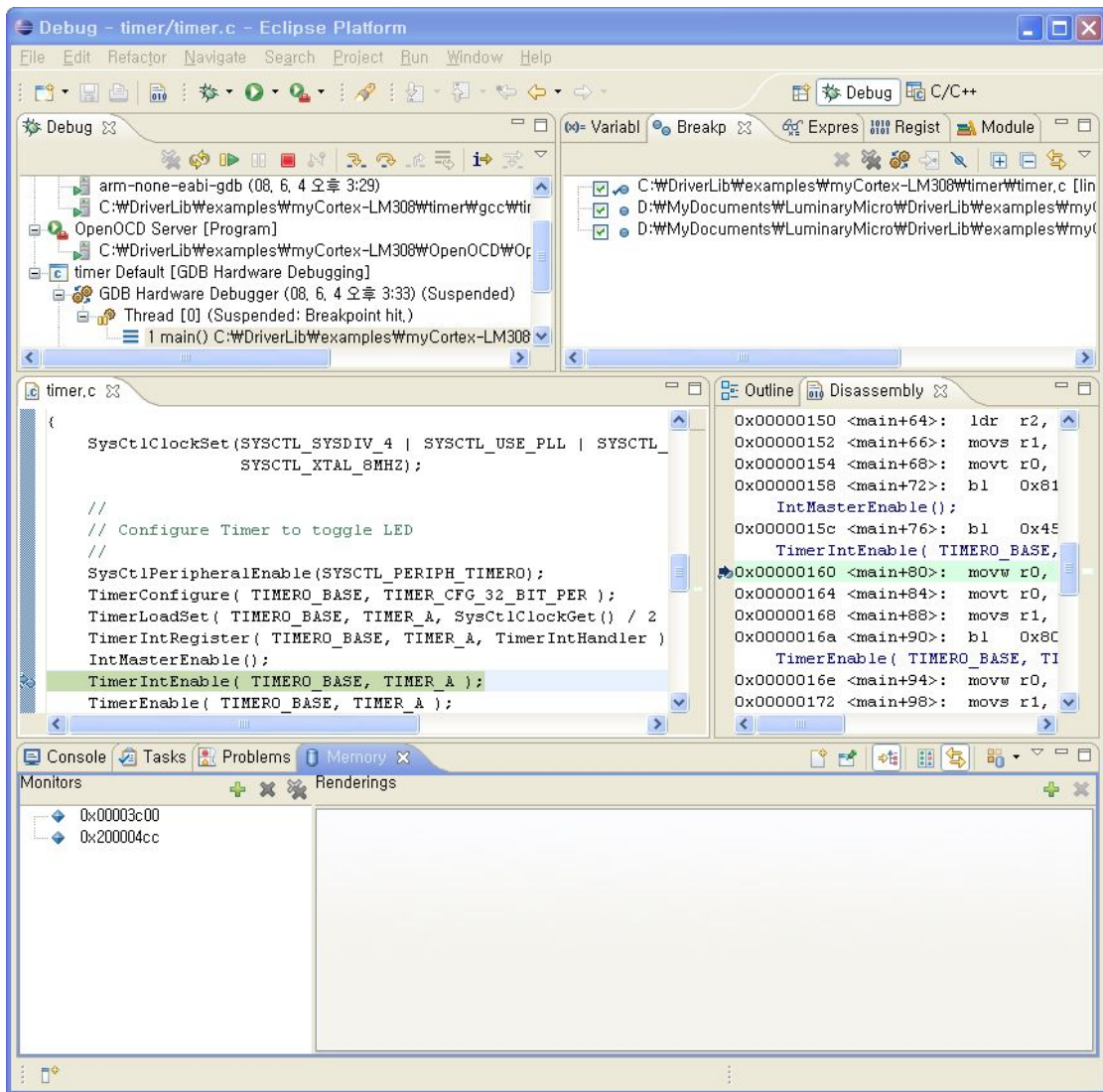
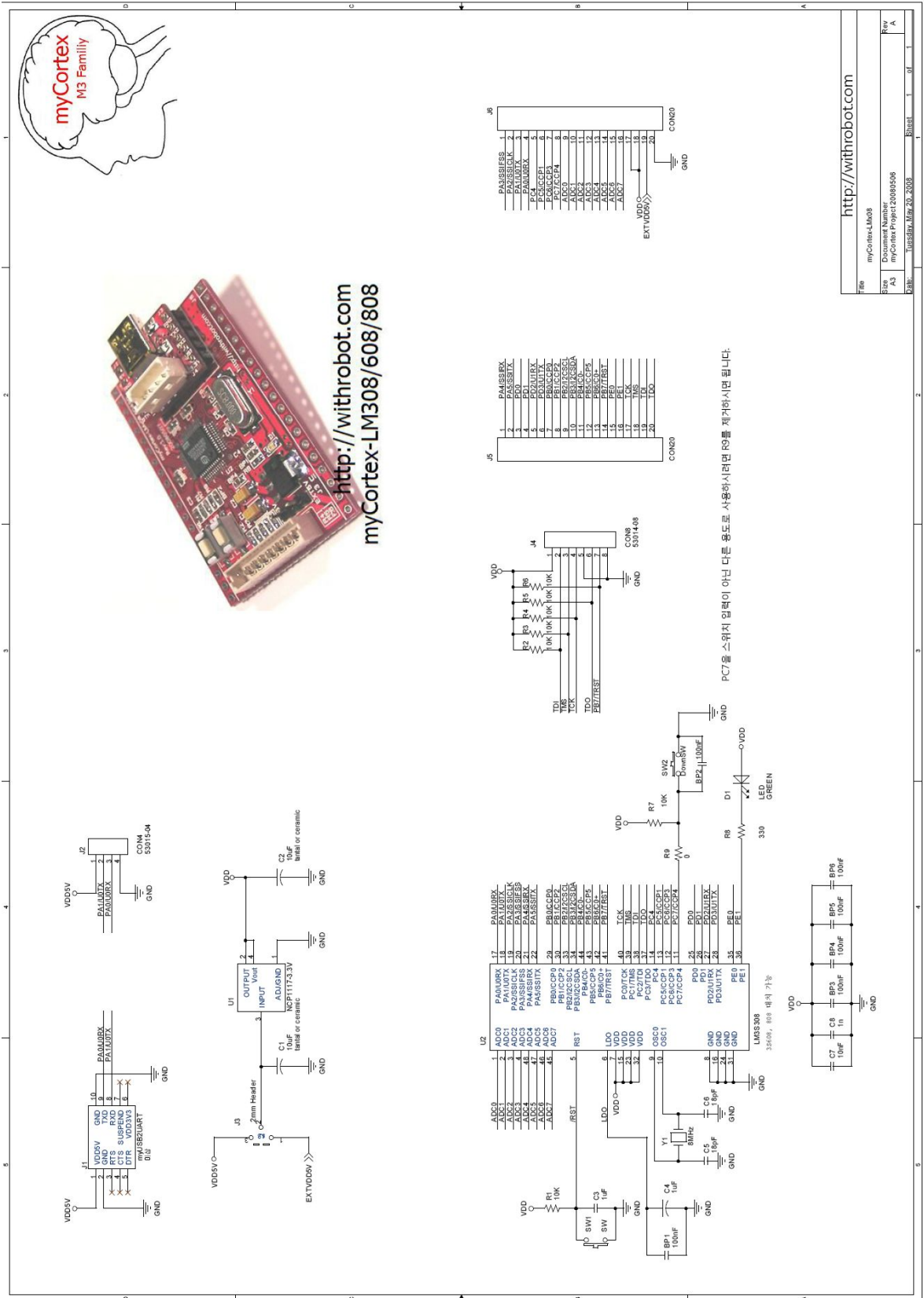


그림 14 Stellaris-JTAG 으로 이클립스에서 소스 레벨 디버깅 모습

전기적 특성

- 동작 전원: 5V 외부 입력 또는 USB BUS power
- I/O 전압: 3.3V(5V tolerant)
- JTAG 신호 전압: 3.3V

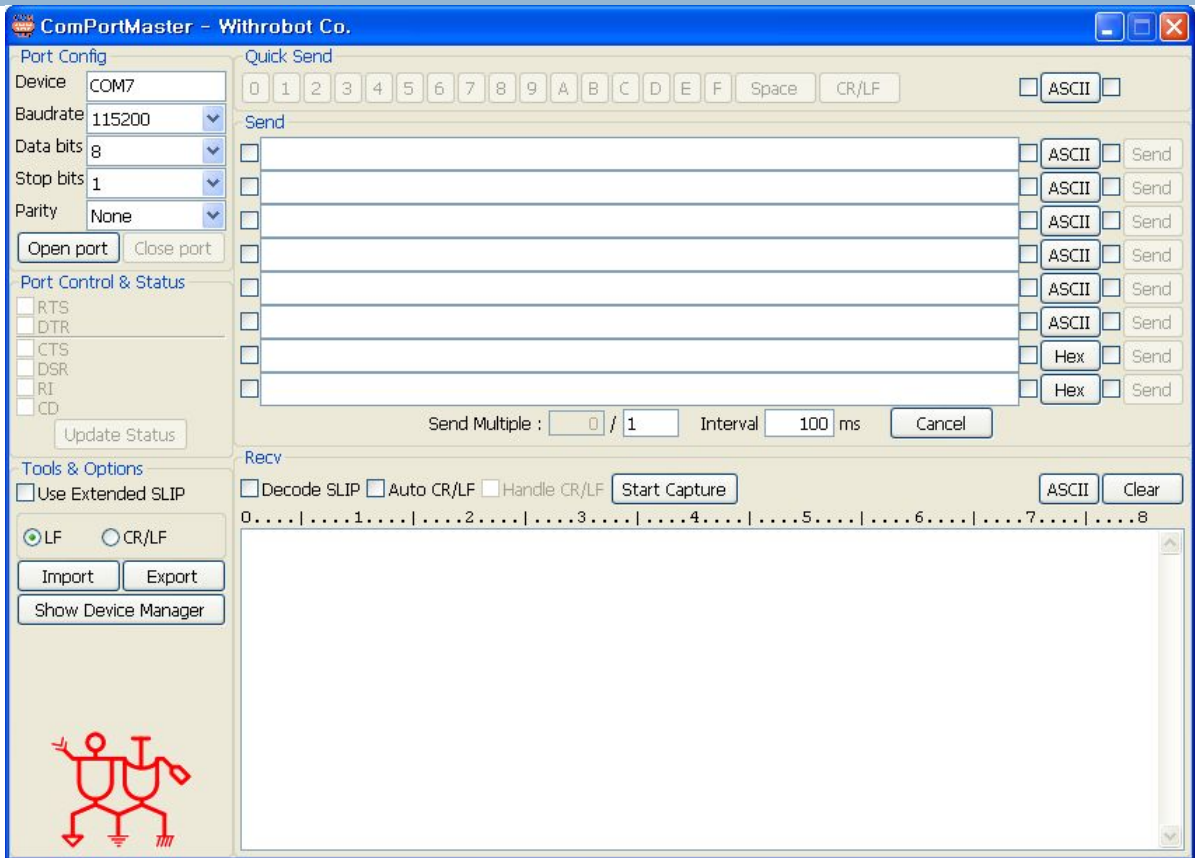
회로도



사용 시 주의 사항

- myCortex-LMx08 보드에 전원을 인가할 때 전원의 극성을 확인하시기 바랍니다. 전원을 바꾸어 연결하거나, 정격 전압이 아닌 전원을 인가하면 myCortex-LMx08 보드가 파손될 수 있으며 **판매사는 이에 대한 책임을 지지 않습니다. 상품 출하 시 전수 검사를 통해 보드의 동작을 확인하고 있으며, 사용 중 파손에 의한 수리는 불가합니다.**
- 다른 디바이스와 연결할 때 마이크로 프로세서의 IO 전압(3.3V)과 동일한지 확인하시기 바랍니다. myCortex-LMx08 보드의 IO 들은 5V 입력에도 파손되지 않도록 설계되어 있지만 가능하면 동작 범위를 일치시키는 것을 추천합니다. 마이크로 프로세서의 IO 전압과 다른 전압을 GPIO 핀에 인가하면 오동작의 원인이 되며, myCortex-LMx08 보드가 파손될 수 있습니다.

부록 - COMPORTMASTER



기능상 특징

- PC 의 COM 포트를 이용한 통신 프로그램
- ASCII/HEX 선택적으로 데이터 송수신 가능
- SLIP 및 위드로봇에서 제안하는 확장 SLIP 인코딩, 디코딩 기능 선택적 사용 가능
- 명령어 2 바이트 데이터 길이 1 바이트로 구성된 WRPu 패킷 자동 생성 기능 지원
- 수신 데이터에 CR/LF 자동 추가 기능
- 패킷 반복 전송 가능 및 데이터 패킷 간 지연 시간 설정 가능
- 흐름 제어 신호들의 모니터링 및 값 설정 가능
- 비상업적인 용도로 무료로 이용 가능(<http://withrobot.com/category/Interface/ComPortMaster>)

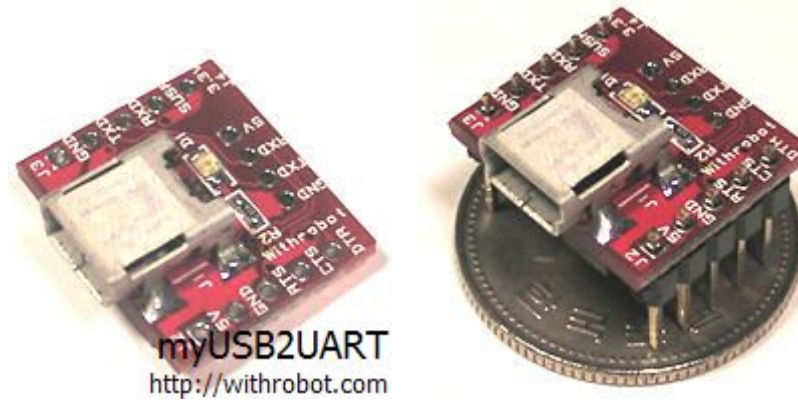
응용 분야

- 임베디드 보드와 PC 의 시리얼 통신 테스트 및 각종 패킷 통신 실험

지원 OS

- Windows 98/XP/Vista

부록 – MYUSB2UART



특징

- 초소형 USB2UART
- MAX232 칩 필요 없이 임베디드 시스템의 UART 를 바로 PC 의 USB 에 연결
- USB 버스 전원 공급 기능(5V, 500mA) 및 전원 보호 회로 내장
- 가상용 COM 포트(VCP) 드라이버 제공
- 2mm 간격의 커넥터 및 흐름제어신호를 포함한 100mil 간격의 커넥터 조합으로 다양한 형태의 결합 가능

Designed by withrobot Lab.

- Homepage: <http://www.withrobot.com>

Release Information

The following changes have been made in this document.

Change history

Date	Issue	변동 사항
2008.6	A	The first draft
2008.11	B	StellarisWare 변경사항 적용

Copyright(c) 2003-2008 withrobot Lab. All right reserved.



www.withrobot.com