# CRZ S3C6410 Mango6410-R2 Enviroment and Boot Mode

http://www.mangoboard.com/ http://cafe.naver.com/embeddedcrazyboys Crazy Embedded Laboratory

# **Document History**

Revision	Date	Change note

## 목차

1.	S3C6410	Mango-R6410 환경구성 및 부팅모드	4
	1.1.	Mango_R6410 연결	4
	1.1.1.	전체연결	4
	1.1.2.	JTAG 연결	5
	1.2.	Mango_R6410 부팅모드 설정	9
	1.2.1.	NAND Boot	9
	1.2.2.	NOR Boot	10
	1.3.	Nor Flash Write하기	10
	1.3.1.	환경구성	10
	1.3.2.	OponOCD 연결하기	10
	1.3.3.	Flash 정보 보기	11
	1.3.4.	Flash 정보 및 erase 하기	14
	1.3.5.	Flash 에 Write하기	16
	1.3.6.	Verify image 하기	17

3

# 1. S3C6410 Mango-R6410 환경구성 및 부팅모드

## 1.1. Mango\_R6410 연결

1.1.1. 전체연결



## 1.1.2. JTAG 연결













7





## 1.2. Mango\_R6410 부팅모드 설정

## 1.2.1. NAND Boot



9

## 1.2.2. NOR Boot



## 1.3. Nor Flash Write하기

### 1.3.1. 환경구성

버전은 Openocd 0.1.0 버전입니다.

```
> version
```

Open On-Chip Debugger 1.0 (2009-04-30-01:48) svn:1547

망고64 보드를 Nor Flash boot Mode로 SW100 스위치를 변경 및 shunt switch CON202,CON200을 0,1로 변경을 해야 합니다.

### 1.3.2. OponOCD 연결하기

그리고, 보드에 FT2232 usb jtag 보드를 연결합니다. 전원을 인가 후 Dos Command창을 실행합니다. 물론 , openocd.exe 파일이 있는 디렉토리에서 실행해야 합니다. 첨부파일에서 >openocd.exe -f mango64nor.cfg 입력하면 됩니다.

J:#Mango_project#Work#Mango-JIAG#Mango64-OpenOCD-Cygwin-091210>openocd.exe  -f mango64nor.cfg Open On-Chip Debugger 1.0 (2009-04-30-01:48) svn:1547
BUGS? Read http://svn.berlios.de/svnroot/repos/openocd/trunk/BUGS
\$URL: svn://svn.berlios.de/openocd/trunk/src/openocd.c \$
jtag_speed: 5
1000 kHz
Info : JTAG tap: s3c6410.unknown tap/device found: 0x2b900f0f (Manufacturer: 0x787, Part: 0xb900, Version: 0x2)
Info : JTAG Tap/device matched
Info : JTAG tap: s3c6410.cpu tap/device found: 0x07b76f0f (Manufacturer: 0x787, Part: 0x7b76, Version: 0x0)
Info : JTAG Tap/device matched
Info : found ARM1176
Warn : no telnet port specified, using default port 4444
Warn : no gdb port specified, using default port 3333
Warn : no tcl port specified, using default port 6666
Info : accepting 'telnet' connection from 0
Open On-Chip Debugger 1.0 (2009-04-30-01:48) svn:1547

openocd 실행

Mango64nor.cfg 파일을 살펴보면,

source [find ./mango64.cfg]

\$\_TARGETNAME configure -work-area-phys 0x0c000000 -work-area-size 0x2000

flash bank cfi 0x0000000 0x00400000 2 2 \$\_TARGETNAME

#### mango64nor.cfg

위의 내용을 살펴보면,

\$\_TARGETNAME configure -work-area-phys 0x0c000000 -work-area-size 0x2000

은 OpenOCD가 steppingstone(SRAM) 영역을 사용하겠다는 의미 입니다.

flash bank cfi 0x0000000 0x00400000 2 2 \$\_TARGETNAME

CFI(Common Flash interface) 의 약자입니다.

0x0000000: 시작 address(정하기 나름)

0x00400000 : 망고64에 달려있는 Flash Size입니다.

2 : flash chip width

2: bus width

\$\_TARGETNAME : target

http://openocd.berlios.de/doc/html/Flash-Commands.html

에 자세히 설명이 나와 있습니다.

또 다른 DOS Command 창을 실행 후

>telnet localhost 4444

실행합니다.

실행 후

#### 1.3.3. Flash 정보 보기

>flash banks 명령을 입력합니다.

```
> flash banks
#0: cfi at 0x00000000, size 0x00400000, buswidth 2, chipwidth 2
\
```

#### flash banks 실행모습

#0: 현재 연결 된 bank number입니다.

>flash probe 0

명령을 수행합니다.

> flash probe 0
Target not halted
unknown error when probing flash bank '#0' at 0x0000000
> halt
Debug entry: JTAG HALT
target state: halted
target halted due to debug-request
cpsr: 0x000001db pc: 0x0000004
> flash probe 0
Flash Manufacturer/Device: 0x00ec 0x257e
flash 'cfi' found at 0x0000000

#### flash probe 실행모습

Flash에 Manufacture/Device 값을 읽어 옵니다.

이것은 flash(K8P3215UQB)의 datasheet를 참조하시면 됩니다.

Table 7. K8P3215UQB Autoselect Codes, (High Voltage Method)

Description		CE	ŌE	WE	A20 to A12	A10	A9	A8	A7	A6	A5 to A4	A3	A2	A1	A0	DQ15 to DQ8	DQ7 to DQ0
Manuf	Manufacturer ID		L	н	DA	х	Vio	х	E	L	х	L	L	Ľ	L	х	ECH
Devi	Read Cycle 1						Vip	x	L			L	L	L	н	25H	7EH
ce	Read Cycle 2	L	L	н	DA	х				L	L	н	н	H	L	25H	03H
ID	Read Cycle 3		8	a		6'	2 3			a - 1		н	н	н	н	25H	01H

Manufacturer ID는 0xEC 이고, Device ID는 0x257E 라는 것을 알 수가 있습니다.

명령을 수행한 값과 동일하다는 것을 알 수 있습니다.

>flash info 0

명령을 수행 해 보겠습니다.

> flash	inf	fo (	3					
#0: cfi	at	Øx	00000000, s:	ize 0x0040	,0000	buswidth	2, chip	width 2
ner den ten - en en en er et et	#	0:	0×00000000	(0x2000 8	8kB) p	rotection	state u	nknown
	#	1:	0×00002000	(0x2000 8	3kB) p	rotection	state u	nknown
	#	2:	0×00004000	(0x2000 8	3kB) p	rotection	state u	nknown
	#	3:	0×00006000	(0x2000 8	8kB) p	rotection	state u	nknown
	#	4:	0×00008000	(0x2000 8	8kB) p	rotection	state u	nknown
	#	5:	0x0000a000	(0x2000 8	8kB) p	rotection	state u	nknown
	#	6:	0x0000c000	(0x2000 8	3kB) p	rotection	state u	nknown
	#	7:	0x0000e000	(0x2000 8	3kB) p	rotection	state u	nknown
	#	8:	0×00010000	(0x10000	64kB)	protectio	n state	unknown
	#	9:	0×00020000	(0x10000	64kB)	protectio	n state	unknown
	# 1	10:	0×00030000	(0x10000	64kB)	protectio	n state	unknown
	# 1	11:	0×00040000	(0x10000	64kB)	protectio	n state	unknown
	# 1	12:	0×00050000	(0x10000	64kB)	protectio	n state	unknown
	# 1	13:	0×00060000	(0x10000	64kB)	protectio	n state	unknown
	# 1	14:	0×00070000	(0x10000	64kB)	protectio	n state	unknown
	# 1	15:	0×00080000	(0x10000	64kB)	protectio	n state	unknown
	# 1	16:	0×00090000	(0x10000	64kB)	protectio	n state	unknown
	# 1	17:	0x000a0000	(0x10000	64kB)	protectio	n state	unknown
	# 1	18:	0×000b0000	(0x10000	64kB)	protectio	n state	unknown
	# 1	19:	0x000c0000	(0x10000	64kB)	protectio	n state	unknown
	# 2	20:	0x000d0000	(0x10000	64kB)	protectio	n state	unknown
	# 2	21:	0x000e0000	(0x10000	64kB)	protectio	n state	unknown
	# 2	22:	0x000f0000	(0x10000	64kB)	protectio	n state	unknown
	# 2	23:	0×00100000	(0x10000	64kB)	protectio	n state	unknown
	# 2	24:	0×00110000	(0×10000	64kB)	protectio	n state	unknown
	# 2	25:	0×00120000	(0×10000	64kB)	protectio	n state	unknown

flash info 실행모습

cfi information*
LI INDRALION.
mfr: 0x00ec, id:0x257e
qry: 'QRY', pri_id: 0x0002, pri_addr: 0x0040, alt_id: 0x0000, alt_addr: 0x0000
Ucc min: 2.7, Ucc max: 3.6, Upp min: 0.0, Upp max: 0.0
typ. word write timeout: 8, typ. buf write timeout: 1, typ. block erase timeout: 512, typ. chip erase timeout: 1
max. word write timeout: 128, max. buf write timeout: 1, max. block erase timeout: 8192, max. chip erase timeout: 1
size: 0x400000, interface desc: 1, max buffer write size: 1
Spansion primary algorithm extend information:
pri: 'PRI', version: 0.0
Silicon Rev.: 0x0, Address Sensitive unlock: 0x0
Erase Suspend: 0x2, Sector Protect: 0x1
VppMin: 08.5, VppMax: 09.5

flash info 실행모습 2

훨씬 많은 정보를 볼수가 있습니다. Manufacture ID와 VCC 전압까지 나오고, Spansions primary algorithm을 사용 했다는 내용까지 나옵니다. 멋집니다.

## 1.3.4. Flash 정보 및 erase 하기

>flash info 0 명령으로 보면, 23번 sector부터 1Mbyte가 시작됩니다. 현재 Shunt Switch를 조정을 해 놓았기 때문에 0x00100000부터 Write하면,됩니다. http://cafe.naver.com/embeddedcrazyboys/1606

참조 하시기 바랍니다.



Mango64 보드 Nor Flash 결선도

Nor flash 메모리 MAP을 살펴보면 아래와 같습니다.

г	4MBytes NOR Flash	NOR Address S3C6410 Address
NOR_A19[CON200] : HIGH NOR_A20 <u>[CQN202] : HIGH</u>	BANK 3	0x30_0000_0000 0x00_0000
NOR_A19[CON200] : LOW NOR_A20 <u>[CON202] : HIGH</u>	BANK 2	0x20_0000_0000 0x00_0000
NOR_A19[CON200] : HIGH NOR_A20[CON202] : LOW	BANK 1	0x10_0000_0000 0x00_0000_0000
NOR_A19[CON200] : LOW NOR_A20[CQN202] : LOW	BANK 0	<i>0x09_0000_0000</i> _0x00_0000_0000

[그림 2] NOR Flash 주소 공간 분할 구조 NOR Flash 주소 분할 구조

결국, S3C6410입장에서는 시작번지는 0입니다. 현재 CON200: High, CON202: Low 상태로 해 놓았습니다.

>flash erase\_address 0x00100000 0x10000

명령을 살펴보면, 0x00100000 시작해서, 0x10000(64KB) 만큼 erase합니다.

64KB만 erase하는 이유는 6410nor.bin 이 51436Byte 사이즈이기 때문입니다.

대충 봐도 64KB를 넘지 않습니다.

> flash erase\_address 0x00100000 0x10000 erased address 0x00100000 length 65536 in 0.469000s

flash erase address 실행모습

또 다른 방법은 Block Number로 지우는 방법입니다.

> flash	info	0					
#0: cfi	at Øx	00000000, s:	ize 0x0040	<i>,</i> 00000	buswidth	2, chip	width 2
68925-04 - 10049868	# Ø:	0x00000000	(0x2000 8	8kB) p	rotection	state u	nknown
	# 1:	0×00002000	(0x2000 8	8kB) p	rotection	state u	nknown
	# 2:	0×00004000	(0x2000 8	8kB) p	rotection	state u	nknown
	# 3:	0×00006000	(0x2000 8	8kB) p	rotection	state u	nknown
	# 4:	0×00008000	(0x2000 8	8kB) p	rotection	state u	nknown
	# 5:	0x0000a000	(0x2000 8	8kB) p	rotection	state u	nknown
	# 6:	0x0000c000	(0x2000 8	8kB) p	rotection	state u	nknown
	# 7:	0x0000e000	(0x2000 8	8kB) p	rotection	state u	nknown
	# 8:	0×00010000	(0x10000	64kB)	protectio	n state	unknown
	# 9:	0×00020000	(0x10000	64kB)	protectio	n state	unknown
	# 10:	0×00030000	(0x10000	64kB)	protectio	n state	unknown
	# 11:	0×00040000	(0x10000	64kB)	protectio	n state	unknown
	# 12:	0×00050000	(0x10000	64kB)	protectio	n state	unknown
	# 13:	0×00060000	(0x10000	64kB)	protectio	n state	unknown
	# 14:	0×00070000	(0x10000	64kB)	protectio	n state	unknown
	# 15:	0×00080000	(0x10000	64kB)	protectio	n state	unknown
	# 16:	0×00090000	(0x10000	64kB)	protectio	n state	unknown
	# 17:	0x000a0000	(0x10000	64kB)	protectio	n state	unknown
	# 18:	0х000Ъ0000	(0x10000	64kB)	protectio	n state	unknown
	# 19:	0x000c0000	(0x10000	64kB)	protectio	n state	unknown
	# 20:	0x000d0000	(0x10000	64kB)	protectio	n state	unknown
	# 21:	0x000e0000	(0×10000	64kB)	protectio	n state	unknown
	# 22:	0×000f0000	(0×10000	64kB)	protectio	n state	unknown
	# 23:	0×00100000	(0×10000	64kB)	protectio	n state	unknown
	# 24:	0×00110000	(0×10000	64kB)	protectio	n state	unknown
	# 25:	0×00120000	(0x10000	64kB)	protectio	n state	unknown

block number 23

Block Number 23이 64Kbyte를 영역을 가지고 있습니다.

>flash erase\_sector 0 23 23

하면 됩니다. 명령 형식은 아래와 같습니다.

flash	erase_address	erase	address range <address> <length></length></address>
flash	erase_check	check	erase state of sectors in flash bank <num></num>
flash	erase_sector	erase	sectors at 〈bank〉 〈first〉 〈last〉

flash erase sector 형식

실행 결과 입니다.

> flash erase\_sector 0 23 23 erased sectors 23 through 23 on flash bank 0 in 0.500000s

flash erase sector 결과

1.3.5. Flash 에 Write하기

명령은

>flash write\_bank 0 6410nor.bin 0x100000

write binary data to <bank> <file> <offset> flash write\_bank flash bank write 형식 살펴 보면, bank은 >flash banks 명령을 보면 0입니다. File은 PC에 6410nor.bin 파일이름을 입력합니다. Offset은 0x100000가 Write하기 위한 시작 address입니다. 실행 결과 > flash write\_bank 0 6410nor.bin 0x100000 not enough working area available(requested 32768, free 8096) not enough working area available(reguested 16384, free 8096) not enough working area available(requested 8192, free 8096) Debug entry: breakpoint target state: halted target halted due to breakpoint cpsr: 0x600001d3 pc: 0x0c00005c Debug entry: breakpoint target state: halted target halted due to breakpoint cpsr: 0x600001d3 pc: 0x0c00005c flash bank write 실행 결과 1.3.6. Verify image 하기 명령은 >verify\_image 6410nor.bin 0x00100000 bin 명령 형식을 보면, verify\_image <file> [offset] [type] verify\_image verify image 형식 <file> : flash에 Write한 이미지 이름 <offset>: Write 시작 address <type>: binary,axf, 등 format형태 실행 결과 > verify\_image 6410nor.bin 0x00100000 bin checksum mismatch – attempting binary compare BUG: keep\_alive() was not invoked in the 1000ms timelimit. GDB alive packet not sent! (55484) verified 51436 bytes in 55.437000s verify 실행결과

또 다른 방법은

>dump\_image nordump.bin 0x00100000 0x10000

명령으로 ,nordump.bin과 6410nor.bin을 비교해 보는 방법입니다.

작은 이미지인 경우만 사용하세요. 이미지 크기가 크면, 하 세월입니다.

> dump\_image nordump.bin 0x00100000 0x10000 dumped 65536 byte in 70.655998s

dump image 실행결과

Beyond Compare 프로그램으로 비교결과 값이 동일합니다.

0xC8EC(51436)Byte 동일합니다.

ordump.t	oin		> 6	410	Ino	r.b	in -	- H	ex	Cor	np	are	-	Bey	/on	d C	om																	JC	
sion <u>F</u> ile	Sea	arch	⊻i	зw	Τo	ols	H	elp																						Nev	v ve	rsio	n av	ailat	ole
Sessions	•	*	¥	=	2		*****	¥	4	A		9	¢	1	]																				
J:₩₩M	lang	o64	Оре	nO	CD-	Су	a wir	n-09	1210	)₩n	ordu	ımp	, bin	_	~	Ð	2	-	J:₩₩Ma	ang	o64-	Оре	nOC	D-0	Cygw	in-O	91210	)₩64	410n	or, b	in		*	2	2
2010-01-	13 오	章 (	3:08:	32 1	65,5	36 E	yte	s <	def	ault	•								2009-03-1	8 오	章 3	8:55:	22 5	51,43	36 by	tes	<de< td=""><td>fault</td><td>&gt; •</td><td></td><td></td><td></td><td></td><td></td><td></td></de<>	fault	> •						
0000C7F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	1	^	0000C7F0	00	00	00	00	00	00 0	0 0	0 00	00	00	00	00	00	00	00	
000000800	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00			000000800	00	00	00	00	00	00 0	0 0	0 00	00	00	00	00	00	00	00	
00000810	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00			000000810	00	00	00	00	00	00 0	0 0	0 00	00	00	00	00	00	00	00	
00000620	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00			000000820	00	00	00	00	00	00 0	0 0	0 00	00	00	00	00	00	00	00	
000000840	AA	AA	AA	AA	AA	AA	AA	AA	EE	EE	EE	EE	EE	EE	EE	EE			000000840	AA	AA	AA	AA	AA	AA A	AA	A EE	EE	EE	EE	EE	EE	EE	EE	
00000850	FE	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	7F	BF	DF	EF			00000850	FE	FF	FF	FF	FF	FF I	FF	F FF	FF	FF	FF	7F	BF	DF	EF	
00000860	F7	FB	FD	FC	7E	BF	DF	EF	F7	FB	FD	7E	00	00	00	00			000000860	F7	FB	FD	FC	7E	BF I	FE	E E7	FB	FD	7E	00	00	00	00	
00000870	00	00	00	00	00	10	21	50	00	00	00	00	00	00	00	00			00000870	00	00	00	00	00	10 2	1 5	0 00	00	00	00	00	00	00	00	
000000880	00	C2	01	00	00	00	00	00	00	03	00	00	01	01	01	01			000000880	00	C2	01	00	00	00 0	0 0	0 00	03	00	00	01	01	01	01	
000000830	01	00	00	00	00	00	01	00	00	00	00	00	00	03	00	00			000000830	01	00	00	00	00	00 0	0 1		00	00	00	00	03	00	00	
000000880	00	03	00	00	01	01	01	01	01	00	00	00	00	C2	01	00			000000880	00	03	00	00	01	01 0	1 0	1 01	00	00	00	00	C2	01	00	
000000800	00	00	00	00	00	03	00	00	01	01	01	01	01	00	00	00			000000800	00	00	00	00	00	03 0	0 0	0 01	01	01	01	01	00	00	00	
000000800	00	C2	01	00	00	00	00	00	00	03	00	00	01	01	01	01			000000800	00	C2	01	00	00	00 0	0 0	0 00	03	00	00	01	01	01	01	
0000C8E0	01	00	00	00	00	00	00	00	00	00	00	00	FF	FF	FF	FF			0000C8E0	01	00	00	00	00	00 0	0 0	0 00	00	00	00					

compare 결과