Chipsmall
深圳市津利帝科技有限公司
Geelead Technology Limited
采购部

JTAG-Booster for

Samsung S3C24xx

Forth-Systeme GmbH

P.O: Box 1103 ● D-79200 Breisach, Germany
Kueferstrasse 8 ● D-79206 Breisach, Germany
Tel. +49 (7667) 908-0 ● Fax +49 (7667) 908-200
sales@fsforth.de ● http://www.fsforth.de

FS FORTH-SYSTEME GmbH
Postfach 1103, D-79200 Breisach, Germany

Release of Document:	September 11, 2003
Author:	Dieter Fögele
Filename:	JTAG_S3C24xxa.doc
Program Version:	4.xx

Table of Contents

1. General

The programs JTAG2410.EXE and JTAG2440.EXE use the JTAG port of the Samsung S3C24xx mircocontrollers in conjunction with the small JTAG-Booster:

- to program data into flash memory
- to verify and read the contents of a flash memory
- to make a memory dump
- to access an I²C Device
- to test CPU signals

All functions are done without any piece of software running in the target. No firmware or BIOS must be written. Bootstrap software may be downloaded to initially unprogrammed memories.

The JTAG-BOOSTER' s software is highly optimized to the JTAG chain of a specific target CPU. To give support for all processors of the Samsung S3C24xx family, there are two different programs on the distribution disk:

- JTAG2410.EXE : Tool for Samsung S3C2410
- JTAG2440.EXE : Tool for Samsung S3C2440

Please contact us, if you need support for other members of the Samsung S3C24xx family.

For latest documentation please refer to the file README.TXT on the distribution disk.

1.1. Ordering Information

The following related products are available

- 9015 JTAG-Booster Samsung S3C24xx, 3.3V,
  S3C2410 and S3C2440
  DOS/Win9x/WinNT/Win2000/WinXP,
  delivered with adapter type 285

1.2. System Requirements

To successfully run this tool the following requirements must be met:

- MSDOS, WIN3.x, WIN9x, WinNT, Win2000 or WindowsXP (WinNT/Win2000/WindowsXP is supported with an additional tool, see chapter 5 “Support for Windows NT, Windows 2000 and Windows XP”)

- Intel 80386 or higher

- 205 kByte of free DOS memory

- Parallel Port

1.3. Contents of Distribution Disk

- JTAG2410.EXE
JTAG2410.OVL — Tool for Samsung S3C2410

- JTAG2410.INI — Template configuration file for Samsung S3C2410. See chapter 1.9 "Initialization file JTAGxxxx.INI"

- JTAG2440.EXE
JTAG2440.OVL — Tool for Samsung S3C2440

- JTAG2440.INI — Template configuration file for Samsung S3C2440. See chapter 1.9 "Initialization file JTAGxxxx.INI"

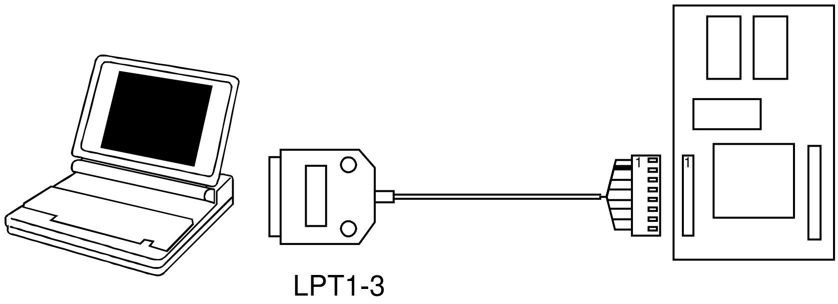
- WinNT.zip — Support for Windows NT and Windows 2000. See chapter 5 "Support for Windows NT, Windows 2000 and Windows XP"

- JTAG_V4xx_FLASHES.pdf — List of all supported Flash devices

- README.txt — Release notes, new features, known problems

1.4. Connecting your PC to the target system

The JTAG-Booster can be plugged into standard parallel ports (LPT1-3) with a DB25-Connector.



The target end of the cable has the following reference:

1	2*	3	4	5	6	7	8
TCK	GND	TMS	TRST#	NC	TDI	TDO	+3.3V

*PIN 2 can be detected by the thick cable.

To connect your design to the JTAG-BOOSTER you need a single row berg connector with a spacing of 2.54mm on your PCB. The names refer to the target: Pin 7 is the target's TDO pin and is connected to the JTAG-Booster's TDI pin.

The 3.3V version of the JTAG-Booster (FS part number 285) is delivered together with this package. Don't use the 5V version of the JTAG-Booster (FS part number 227) with a 3.3V target. **Don't apply 5V to the 3.3V version of the JTAG-Booster!**

Your target must be able to power the JTAG-Booster, it draws about 100mA.

Before you start the program, the JTAG-BOOSTER must be plugged to a parallel interface of your PC and to the 8 pin JTAG connector on the target.

The utility is started with the general command line format: JTAGxxx

```
JTAGxxxx /function [filename] [/option_1] ... [/option_n].
```

Note that the function must be the first argument followed (if needed) by the filename.

If you want to cancel execution of JTAGxxxx, press CTRL-Break-Key.

On any error the program aborts with an MSDOS error level of one.

1.5. First Example with Samsung S3C2410

In the following simple example it is assumed that the JTAG-Booster is connected to LPT1 of your PC and target power is on.

Typing

```
        JTAG2410  /P  MYAPP.BIN
```

at the DOS prompt results in the following output:

```
        JTAG2410 --- JTAG utility for Samsung S3C2410
        Copyright © FS FORTH-SYSTEME GmbH, Breisach
        Version 4.xx of mm/dd/yyyy

(1)     Configuration loaded from file JTAG2410.INI
(2)     Target: Generic Target
(3)     Using LPT at I/O-address 0378h
(4)     JTAG Adapter detected

(5)     1 Device detected in JTAG chain
        Device 0: IDCODE=0032409D  Samsung S3C2410, Revision 0
(6)     Sum of instruction register bits  : 4
(7)     CPU position                      : 0
(8)     Instruction register offset       : 0
(9)     Length of boundary scan reg       : 427

        Looking for a known flash device. Please wait..
(10)    AMD 29LV160B, 3,3V, Boot Block Bottom  detected
(11)    Bus size is 16 Bit
(12)    Erasing Flash-EPROM Block #:0  1  2  3
        Programming File MYAPP.BIN
        65536 Bytes programmed successfully

        Erase Time          :         0.8 sec
        Programming Time    :        48.1 sec
```

(1) The initialization file JTAG2410.INI was found in the current directory.

(2) The target identification line of the initialization file is printed here.

(3) The resulting I/O-address of the parallel port is printed here. With WinNT/Win2000/WinXP you must specify the option /LPT2 to access to the standard address 378h.

(4) A JTAG-Booster is found on the parallel port

(5) The JTAG chain is analyzed. There may be several parts in the JTAG chain. The chain is analyzed and all parts except the Samsung S3C2410 are switched to bypass mode.

(6) The length of all instruction registers in the JTAG chain are added.

(7) The position of the Samsung S3C2410 in the JTAG chain is assumed to be zero, if not specified in the command line (see option /CPUPOS=).

(8) The position of the JTAG instruction register of the Samsung S3C2410 is assumed to be zero, if not specified in the command line (see option /IROFFS=).

(9) The real length of the boundary scan register is displayed here and compared with the boundary scan register length of a Samsung S3C2410.

(10) A Flash AMD 29LV160B selected with GCS0# was found.

(11) The resulting data bus size is printed here.

(12) In this example 4 blocks must be erased.

1.6. First Example with Samsung S3C2440

In the following simple example it is assumed that the JTAG-Booster is connected to LPT1 of your PC and target power is on.

Typing

```
        JTAG2440  /P  MYAPP.BIN
```

at the DOS prompt results in the following output:

```
        JTAG2440 --- JTAG utility for Samsung S3C2440
        Copyright © FS FORTH-SYSTEME GmbH, Breisach
        Version 4.xx of mm/dd/yyyy

(1)     Configuration loaded from file JTAG2440.INI
(2)     Target: Generic Target
(3)     Using LPT at I/O-address 0378h
(4)     JTAG Adapter detected

(5)     1 Device detected in JTAG chain
          Device 0: IDCODE=xxxxxxxx  Samsung S3C2440, Revision ?
(6)     Sum of instruction register bits  : 4
(7)     CPU position                      : 0
(8)     Instruction register offset       : 0
(9)     Length of boundary scan reg       : xxx

        Looking for a known flash device. Please wait..
(10)    AMD 29LV160B, 3.3V, Boot Block Bottom  detected
(11)    Bus size is 16 Bit
(12)    Erasing Flash-EPROM Block #:0  1  2  3
        Programming File MYAPP.BIN
        65536 Bytes programmed successfully

        Erase Time          :          0.8 sec
        Programming Time    :         xx.0 sec
```

(1) The initialization file JTAG2440.INI was found in the current directory.

(2) The target identification line of the initialization file is printed here.

(3) The resulting I/O-address of the parallel port is printed here.

(4) A JTAG-Booster is found on the parallel port

(5) The JTAG chain is analyzed. There may be several parts in the JTAG chain. The chain is analyzed and all parts except the Samsung S3C2440 are switched to bypass mode.

(6) The length of all instruction registers in the JTAG chain are added.

(7) The position of the Samsung S3C2440in the JTAG chain is assumed to be zero, if not specified in the command line (see option /CPUPOS=).

(8) The position of the JTAG instruction register of the Samsung S3C2440is assumed to be zero, if not specified in the command line (see option /IROFFS=).

(9) The real length of the boundary scan register is displayed here and compared with the boundary scan register length of a Samsung S3C2440.

(10) A Flash AMD 29LV160B selected with GCS0# was found.

(11) The resulting data bus size is printed here.

(12) In this example four block must be erased.

1.7. Trouble Shooting

Avoid long distances between your Host-PC and the target. If you are using standard parallel extension cable, the JTAG-BOOSTER may not work. Don't use Dongles between the parallel port and the JTAG-BOOSTER.

Switch off all special modes of your printer port (EPP, ECP, ...) in the BIOS setup. Only standard parallel port (SPP) mode is allowed.

If there are problems with autodetection of the flash devices use the /DEVICE= option. To speed up autodetection specify one of the options /8BIT /16BIT or /32BIT.

Don't use hardware protected flash memories.

The used chip selects must be defined as output and inactive in the initialization file (see chapter 1.9 “Initialization file JTAGxxxx.INI”). Also the address bits must be defined as output.

Use the option /NOWRSETUP to speed up flash programming.

1.8. Error Messages

- **80386 or greater required**
  The JTAG-BOOSTER does not work on a 8088/8086 or a 80286 platform.

- **Cable not connected or target power fail**
  The JTAG-Booster (or one of the simple Parallel Port JTAG adapters selected with the options /LATTICE /WIGGLER /PLS) wasn't found. Please check connection to parallel port and connection to target. Check target power. Check the command line options. Check your BIOS-Setup. If you are using this program with WinNT, Win2000 or WinXP you must specify /LPT2 or /LPT-BASE=378 to get access to the standard printer port.

- **Can't open x:\yyy\zzz\JTAGxxxx.OVL**
  The overlay file JTAGxxxx.OVL must be in the same directory as JTAGxxxx.EXE.

- **Configuration file XYZ not found.**
  The file specified with the option /INI= wasn't found.

- **Device offset out of range**
  The value specified with the option /OFFSET= is greater than the size of the detected flash device.

- **Disk full**
  Writing a output file was aborted as a result of missing disk space.

- **Do not specify option /NOCS with any other chip select**
  There is a conflict in the command line.

- **Do not specify option /BYTE-MODE. *Flash device* does not have a byte mode pin.**
  The flash device specified with the option /DEVICE= does not support switching between 16 (or 32) bit mode and 8 bit mode. In practice it does not have a pin with the name BYTE#

- **Error creating file:**
  The output file could not be opened. Please check free disk space or write protection.

- **Error: *Pin-Name* is an output only pin**
  The specified pin cannot be sampled. Check the command line. Check the initialization file.

- **Error: *Pin-Name* is an input only pin**
  The specified pin cannot be activated. Check the command line. Check the initialization file.

- **Error: *Pin-Name* may not be read back**
  The specified pin can be switched to tristate, but cannot be read back. Check the command line.

- **illegal function:**
  The first parameter of the command line must be a valid function. See chapter 2 “JTAGxxxx Parameter Description” for a list of supported functions.

- **illegal number:**
  The specified number couldn’t be interpret as a valid number. Check the relevant number base.

- **illegal option:**
  See chapter 2 “JTAGxxxx Parameter Description” for a list of supported options.

- **illegal Pin Type:**
  The name specified with the option /PIN= must be one of the list of chapter 1.9 ”Initialization file JTAGxxxx.INI”

- **illegal Flash Type:**
  The name specified with the option /DEVICE= must be one of the list of chapter 1.10 ”Supported flash devices”.

- **Input file not found:**
  The specified file cannot be found

- **Input file is empty:**
  Files with zero length are not accepted

- **"   " is undefined**
  Please check the syntax in your configuration file. (See chapter 1.9 “Initialization file JTAGxxxx.INI”).

- **LPTx not installed**
  The LPT port specified with /LPTx cannot be found. Please check the LPT port or specify a installed LPT port. Check your BIOS setup. If you are using this program with WinNT, Win2000 or WinXP you 1st must install the WinNT support package as described in chapter 5"Support for Windows NT, Windows 2000 and Windows XP

- **missing filename**
  Most functions need a filename as second parameter.

- **missing option /I2CCLK=**
  Some functions need the option /I2CCLK= to be defined.

- **missing option /I2CDAT=**
  Some functions need the option /I2CDAT= or the options /I2CDATO= and /I2CDATI= to be defined.

- **missing option /LENGTH=**
  Some functions need the option /LENGTH= to be defined.

- **missing option /PIN=**
  Some functions need the option /PIN= to be defined.

- **More than 9 devices in the JTAG chain or TDO pin stuck at low level**
  The JTAG chain is limited to 9 parts. Check target power. Check the target’s TDO pin.

- **No devices found in JTAG chain or TDO pin stuck at high level**
  A stream of 32 high bits was detected on the pin TDO. TDO may stuck at high level. Check the connection to your target. Check the target power. Check the target’s TDO pin.

- **Option /CPUPOS= out of range**
  The number specified with the option /CPUPOS= must be less or equal to the number of parts minus 1.

- **Option /IROFFS= out of range**
  Please specify a smaller value

- **Part at specified position is not a Samsung S3C24xx**
  The option /CPUPOS= points to a part not a Samsung S3C24xx

- **Pins specified with /I2CCLK= and /I2CDAT= must have different control cells**
  The pin specified with the option /I2CDAT= must be able to be switched to high impedance while the pin specified with option /I2CCLK= is an active output. See chapter 1.9 “Initialization file JTAGxxxx.INI”.

- **Pins specified with /I2CCLK= and /I2CDATI= must have different control cells**
  The pin specified with the option /I2CDATI= must be able to be switched to high impedance while the pin specified with option /I2CCLK= is an active output. See chapter 1.9 “Initialization file JTAGxxxx.INI”.

- **Pins specified with /I2CDATO= and /I2CDATI= must have different control cells**
  The pin specified with the option /I2CDATI= must be able to be switched to high impedance while the pin specified with option /I2CDATO= is an active output. See chapter 1.9 “Initialization file JTAGxxxx.INI”.

- **Specify only one of these options:**
  Some options are exclusive (i.e. /8BIT and /16BIT). Don't mix them.

- **Sum of instruction register bits to low. Should be at least 5 bits for a Samsung S3C24xx**
  The sum of all instruction register bits in the JTAG chain does not fit to the Samsung S3C24xx. Check the target connection. Check the target CPU type. Check the settings for /IROFFS= and /CPUPOS= , if there are several parts in the JTAG chain.

- **Target no longer connected**
  There is a cyclic check of the JTAG chain. Check target power. Check target connection.

- **There are unknown parts in the JTAG chain. Please use the option /IROFFS= to specify the instr. reg. offset of the CPU.**
  If there are unknown parts in the JTAG chain, the program isn't able to determine the logical position of the CPU's instruction register.

- **There is no Samsung S3C24xx in the JTAG chain**
  No Samsung S3C24xx was found in the JTAG chain. Check the target power. Try with option /DRIVER=4 again.

- **Value of option /FILE-OFFSET out of range**
  The value of the option /FILE-OFFSET= points behind end of file.

- **wrong driver #**
  The value specified with the option /DRIVER= is out of range.

- **Wrong Flash Identifier (xxxx)**
  No valid identifier found. Check the specified chip select signal and the bus width. Try with the option /DEVICE= . Use the option /8BIT or /16BIT or /32BIT to specify the correct data bus size.

- **Wrong length of boundary scan register. Should be 427 for a Samsung S3C2410. (Should be ??? for a Samsung S3C2440.)**
  The length of the boundary scan register of the selected part (if there are more than one in the chain) does not fit to the Samsung S3C24xx. Check the target connection. Check the target CPU type. Check the settings for /IROFFS= and /CPUPOS= , if there are several parts in the JTAG chain.

1.9. Initialization file JTAGxxxx.INI

This file is used to define the default direction and level of all CPU signals. This file **must be carefully adapted** to your design with the Samsung S3C24xx. The Target-Entry is used to identify your design which is displayed with most commands.

When the program JTAGxxxx.EXE is started it scans the current directory for an existing initialization file named JTAGxxxx.INI. If no entry is found the default values are used. You may also specify the initialization file with the option /INI= . If the specified file isn't found, the program aborts with an error message.

The CPU pins can also be used with the functions /BLINK (chapter 2.9), /PIN? (chapter 2.10) and /SAMPLE (chapter 2.11) to test the signals on your design.

The sample file below represents the values which are used for default initialization when no initialization file could be found in the current directory and no initialization file is specified with the option /INI=.

Changes to the structure of the file could result in errors. Remarks can be added by using //.

Sample File JTAG2410.INI:

```
// Description file for Samsung S3C2410
Target: Generic Target, 2003/08/21
// Adapt this file carefully to your design!!
// All chip select signals are set to output and inactive.
// All signals should be defined. Undefined signals are set to their defaults.
// Pin names are defined in upper case.
// Low active signals are signed with a trailing #.

// Group 99:   All pins in this group must be set to the same direction
//             These pins are bidirectional
//             During flash programming these pins are switched between
//             input/inactive and output/active.
//             For Flash programming and other memory accesses
//             these pins should be set to Input
DATA0            Inp     //
DATA1            Inp     //
DATA2            Inp     //
DATA3            Inp     //
DATA4            Inp     //
DATA5            Inp     //
DATA6            Inp     //
DATA7            Inp     //

// Group 82:   All pins in this group must be set to the same direction
//             These pins are bidirectional
//             During flash programming these pins are switched between
//             input/inactive and output/active.
//             For Flash programming and other memory accesses
//             these pins should be set to Input
DATA8            Inp     //
DATA9            Inp     //
DATA10           Inp     //
DATA11           Inp     //
DATA12           Inp     //
DATA13           Inp     //
DATA14           Inp     //
DATA15           Inp     //
```

```
// Group 65:  All pins in this group must be set to the same direction
//            These pins are bidirectional
//            During flash programming these pins are switched between
//            input/inactive and output/active.
//            For Flash programming and other memory accesses
//            these pins should be set to Input
DATA16          Inp       //
DATA17          Inp       //
DATA18          Inp       //
DATA19          Inp       //
DATA20          Inp       //
DATA21          Inp       //
DATA22          Inp       //
DATA23          Inp       //

// Group 48:  All pins in this group must be set to the same direction
//            These pins are bidirectional
//            During flash programming these pins are switched between
//            input/inactive and output/active.
//            For Flash programming and other memory accesses
//            these pins should be set to Input
DATA24          Inp       //
DATA25          Inp       //
DATA26          Inp       //
DATA27          Inp       //
DATA28          Inp       //
DATA29          Inp       //
DATA30          Inp       //
DATA31          Inp       //

// Group 138: All pins in this group must be set to the same direction
//            These pins are tristateable, but can not be read back
//            For Flash Programming these pins must be set to output
ADDR1           Out,Lo  //
ADDR2           Out,Lo  //
ADDR3           Out,Lo  //
ADDR4           Out,Lo  //
ADDR5           Out,Lo  //
ADDR6           Out,Lo  //
ADDR7           Out,Lo  //
ADDR8           Out,Lo  //
ADDR9           Out,Lo  //
ADDR10          Out,Lo  //
```

```
ADDR11          Out,Lo  //
ADDR12          Out,Lo  //
ADDR13          Out,Lo  //
ADDR14          Out,Lo  //
ADDR15          Out,Lo  //

// The following pins are tristateable, but can not be read back
// For Flash Programming these pins must be set to output
ADDR0           Out,Lo  // GPA0
ADDR16          Out,Lo  // GPA1
ADDR17          Out,Lo  // GPA2
ADDR18          Out,Lo  // GPA3
ADDR19          Out,Lo  // GPA4
ADDR20          Out,Lo  // GPA5
ADDR21          Out,Lo  // GPA6
ADDR22          Out,Lo  // GPA7
ADDR23          Out,Lo  // GPA8
ADDR24          Out,Lo  // GPA9
ADDR25          Out,Lo  // GPA10
ADDR26          Out,Lo  // GPA11

// The following pins are tristateable, but can not be read back
GCS5#           Out,Hi  // GPA16
GCS4#           Out,Hi  // GPA15
GCS3#           Out,Hi  // GPA14
GCS2#           Out,Hi  // GPA13
GCS1#           Out,Hi  // GPA12

// Group 166:  All pins in this group must be set to the same direction
//             These pins are tristateable, but can not be read back
GCS7#           Out,Hi  // SCS1#
GCS6#           Out,Hi  // SCS0#
SCLK1           Out,Lo  // SDRAM Clock Output
SCLK0           Out,Lo  // SDRAM Clock Output
SRAS#           Out,Hi  //
SCAS#           Out,Hi  //
```

```
// Group 153:  All pins in this group must be set to the same direction
//             These pins are tristateable, but can not be read back
GCS0#           Out,Hi  //
SCKE            Out,Hi  // SDRAM Clock Enable
WE#             Out,Hi  // Write Enable
OE#             Out,Hi  // Output Enable
BE0#            Out,Lo  // WBE0#:DQM0, Byte Enable
BE1#            Out,Lo  // WBE1#:DQM1, Byte Enable
BE2#            Out,Lo  // WBE2#:DQM2, Byte Enable
BE3#            Out,Lo  // WBE3#:DQM3, Byte Enable

// The following pins are complete bidirectional pins.
// The direction of each pin can be set independent of the other pins.
// Each pin can be used as an input.
LEND            Inp     // GPC0/STH, TFT Line End Signal
VCLK            Inp     // GPC1/LCD_HCLK, STN/TFT LCD clock signal
VLINE           Inp     // GPC2/HSYNC:CPV, STN LCD line signal
VFRAME          Inp     // GPC3/VSYNC:STV, STN Frame signal
VM              Inp     // GPC4/VDEN:TP, STN alternate polarity
LCDVF0          Inp     // GPC5, SEC_TFT Timing control signal
LCDVF1          Inp     // GPC6, SEC_TFT Timing control signal
LCDVF2          Inp     // GPC7, SEC_TFT Timing control signal
VD0             Inp     // GPC8, STN/TFT/SEC_TFT Data Bus
VD1             Inp     // GPC9
VD2             Inp     // GPC10
VD3             Inp     // GPC11
VD4             Inp     // GPC12
VD5             Inp     // GPC13
VD6             Inp     // GPC14
VD7             Inp     // GPC15
VD8             Inp     // GPD0
VD9             Inp     // GPD1
VD10            Inp     // GPD2
VD11            Inp     // GPD3
VD12            Inp     // GPD4
VD13            Inp     // GPD5
VD14            Inp     // GPD6
VD15            Inp     // GPD7
VD16            Inp     // GPD8
VD17            Inp     // GPD9
VD18            Inp     // GPD10
VD19            Inp     // GPD11
VD20            Inp     // GPD12
```

```
VD21        Inp    // GPD13
VD22        Inp    // GPD14/SS1#, SPI chip select slave
VD23        Inp    // GPD15/SS0#, SPI chip select slave
I2SLRCK     Inp    // GPE0, IIS-bus channel select
I2SSCLK     Inp    // GPE1, IIS-bus serial clock
CDCLK       Inp    // GPE2, CODEC system clock
I2SDI       Inp    // GPE3/SS0#, IIS-bus serial data input
I2SDO       Inp    // GPE4/I2SSDI, IIS-bus serial data output
SDCLK       Inp    // GPE5, SD clock
SDCMD       Inp    // GPE6, SD receive responce/transmit command
SDDAT0      Inp    // GPE7, SD receive/transmit data
SDDAT1      Inp    // GPE8
SDDAT2      Inp    // GPE9
SDDAT3      Inp    // GPE10
SPIMISO0    Inp    // GPE11
SPIMOSI0    Inp    // GPE12
SPICLK0     Inp    // GPE13
EINT0       Inp    // GPF0
EINT1       Inp    // GPF1
EINT2       Inp    // GPF2
EINT3       Inp    // GPF3
EINT4       Inp    // GPF4
EINT5       Inp    // GPF5
EINT6       Inp    // GPF6
EINT7       Inp    // GPF7
EINT8       Inp    // GPG0
EINT9       Inp    // GPG1
EINT10      Inp    // GPG2/SS0#
EINT11      Inp    // GPG3/SS1#
EINT12      Inp    // GPG4/LCD_PWREN
EINT13      Inp    // GPG5/SPIMISO1
EINT14      Inp    // GPG6/SPIMOSI1
EINT15      Inp    // GPG7/SPICLK1
EINT16      Inp    // GPG8
EINT17      Inp    // GPG9
EINT18      Inp    // GPG10
EINT19      Inp    // GPG11/TCLK1
EINT20      Inp    // GPG12/XMON, touch
EINT21      Inp    // GPG13/XPON#
EINT22      Inp    // GPG14/YMON
EINT23      Inp    // GPG15/YPON#
UCLK        Inp    // GPH8, UART clock signal
CLKOUT0     Inp    // GPH9
```

```
CLKOUT1      Inp      // GPH10
CTS0#        Inp      // GPH0
RTS0#        Inp      // GPH1
TXD0         Inp      // GPH2
RXD0         Inp      // GPH3
TXD1         Inp      // GPH4
RXD1         Inp      // GPH5
TXD2         Inp      // GPH6/RTS1#
RXD2         Inp      // GPH7/CTS1#
TOUT0        Inp      // GPB0, Timer Output
TOUT1        Inp      // GPB1, Timer Output
TOUT2        Inp      // GPB2, Timer Output
TOUT3        Inp      // GPB3, Timer Output
TCLK0        Inp      // GPB4, External timer clock input
XBACK#       Inp      // GPB5, Bus Hold Acknowledge Output
XBREQ#       Inp      // GPB6, Bus Hold Request Input
XDACK1#      Inp      // GPB7, External DMA Acknowledge Output
XDREQ1#      Inp      // GPB8, External DMA Request Input
XDACK0#      Inp      // GPB9, External DMA Acknowledge Output
XDREQ0#      Inp      // GPB10, External DMA Request Input

// The following pins are open drain types, may be inverted!!
// The direction of each pin can be set independent of the other pin.
// Each pin can be used as an input.
IICSCL       Inp      // GPE14
IICSDA       Inp      // GPE15

// The following pins are output only pins.
// Setting to input (tristate) one of these pins results in an error.
PWREN        Out,Hi   // Core Power Enable
FCE#         Out,Hi   // GPA22, Nand Flash Chip Enable
RSTOUT#      Out,Hi   // GPA21, Reset Output
FRE#         Out,Hi   // GPA20, Nand Flash Read Enable
FWE#         Out,Hi   // GPA19, Nand Flash Write Enable
ALE          Out,Lo   // GPA18, Nand Flash Address Enable
CLE          Out,Lo   // GPA17, Nadn Flash Command Latch Enable
```