Semiconductors

AN10258

How to use the LPC900 In-circuit programming (ICP)

Rev. 02 — 21 October 2004 Application note

Document information

Info	Content
Keywords	LPC900 In-circuit programming (ICP)
Abstract	How to use the LPC900 In-circuit programming (ICP)

Revision history

Rev	Date	Description
02	20041021	Second version (9397 750 14227) Modification: • Updated Figure 5 “MCB900 hook-up”.
01	20040324	Initial version (9397 750 12995)

Contact information

For additional information, please visit: **http://www.semiconductors.philips.com**

For sales office addresses, please send an email to: **sales.addresses@www.semiconductors.philips.com**

1. Introduction

This document gives an overview of how the ICP (In Circuit Programming) on the P89LPC9xx can be done.

ICP is one of the programming methods of the LPC900 microcontroller family.

ICP uses a serial shift protocol that requires 5 pins to program: PCL, PDA, Reset, V_{DD} and V_{SS}. ICP is different from ISP (In System Programming) because it is done completely by the microcontroller's hardware and does not require a bootloader.

The ICP programming method is a serial programming method, shifting in programming commands and shifting data in and out with a clock and dataline.

Using ICP on the application board has many advantages:

- The latest software revision can be loaded into the product before shipping.
- Field updates can be made when reprogramming the chip on the application board.
- In the debug stage of development the chip can stay on the application board while being reprogrammed.

2. Hooking up ICP to the Application board

Figure 1 shows how an ICP programming device can be hooked up to the Application board. This allows the microcontroller to be (re)programmed while it is already soldered on the board.

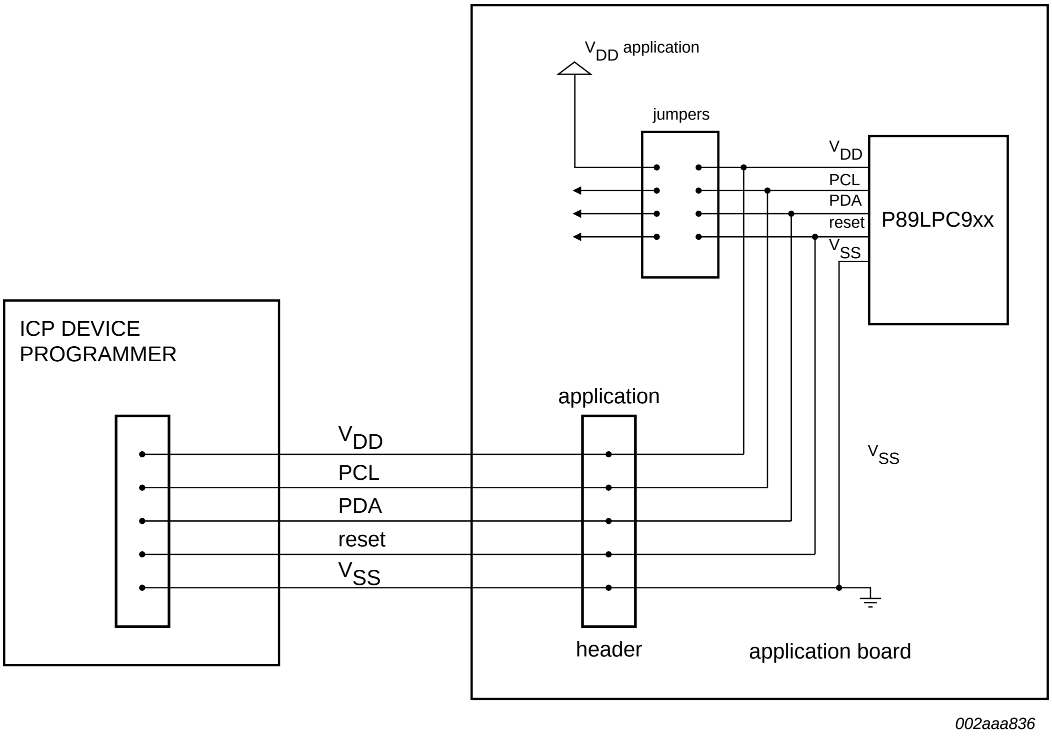


Fig 1. ICP set-up

Table 1 shows the pins needed for ICP programming.

Table 1: Programming pins used for ICP

Mnemonic	Type	Name and function	Disconnect regular connection?
V_{SS}	P	Ground: 0 V reference.	No
V_{DD}	P	Power Supply: 3 V	Yes[1]
PCL	I	Serial clock input for programming communication.	Yes[1]
PDA	I/O	Serial data I/O for programming communication.	Yes[1]
$\overline{\text{RESET}}$	I	ICP programming entry pin	Yes[1]

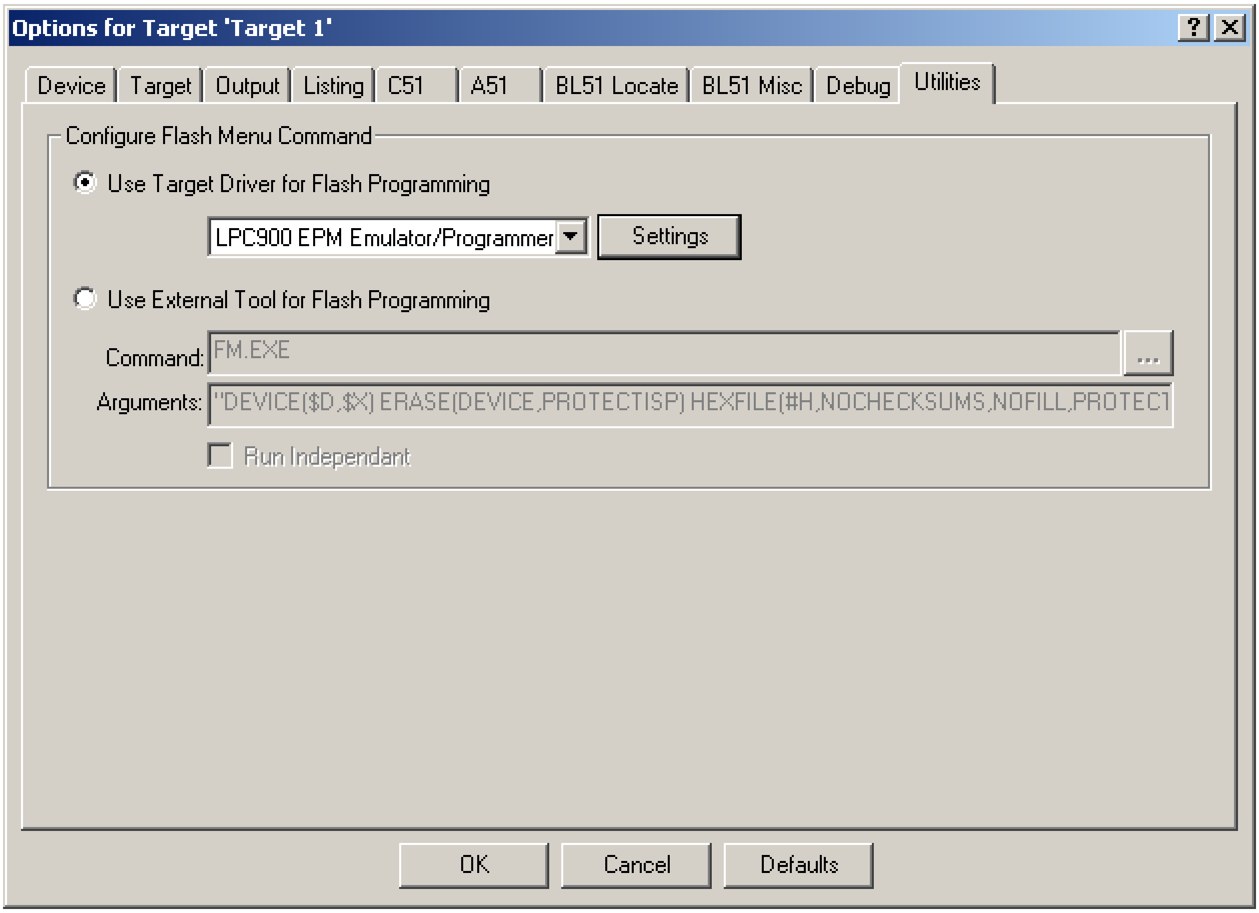
[1] The connection to the application has to be disconnected during ICP programming.

The majority of the LPC900 family have the 5 pins necessary for ICP located in the same configuration. Only the LPC901/902/903 differ from the same configuration, see Section 5 for the pin configurations of the LPC900 parts.

3. Using the EPM900 as an ICP programmer

The EPM900 Emulator / Programmer from Keil Software supports ICP programming of all devices that have ICP. The EPM900 can be setup to do the ICP programming using the Keil µVision IDE. Under the 'Utilities' tab in 'Options for Target', select 'Use Target Driver for Flash Programming' and select the Target to be 'LPC900 EPM Emulator/ Programmer'.

Click on 'Settings' to select the device you want to program.



Options for Target 'Target 1'

Configure Flash Menu Command

● Use Target Driver for Flash Programming

LPC900 EPM Emulator/Programmer

○ Use External Tool for Flash Programming

Command: FM.EXE

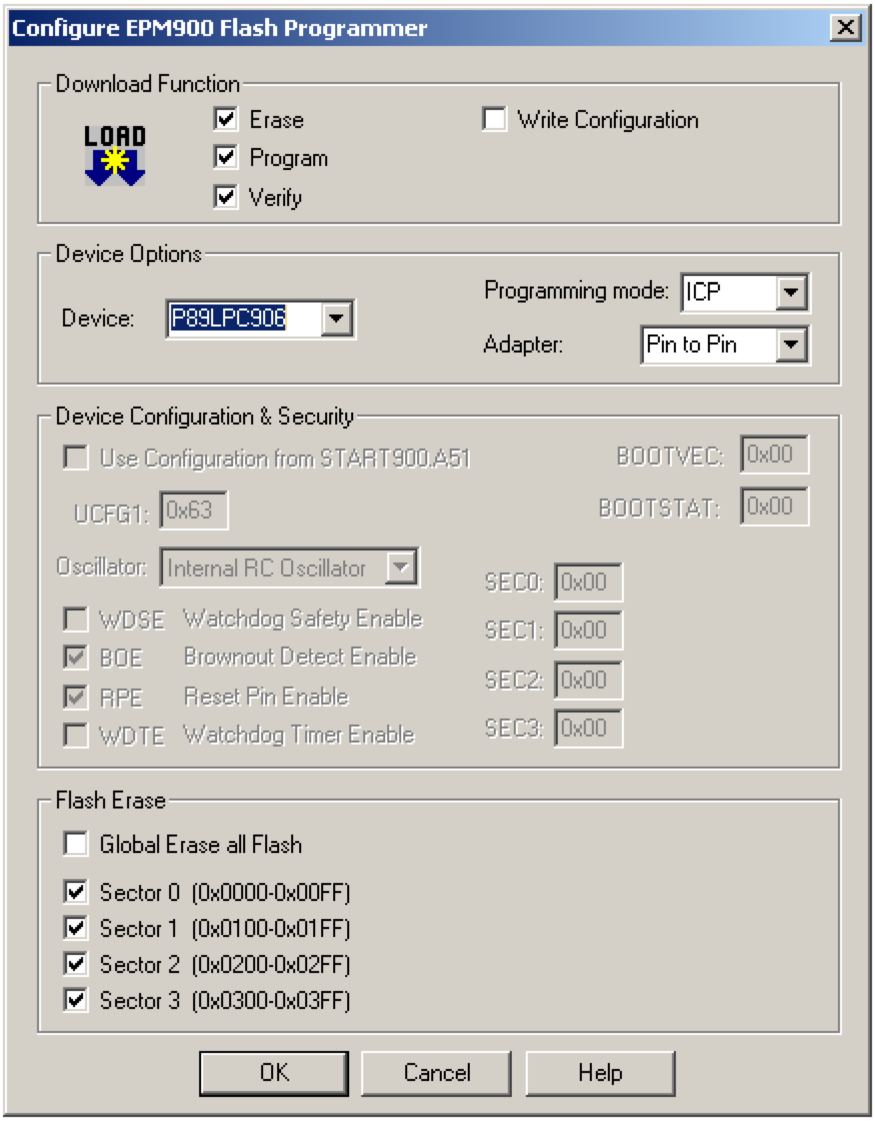
Arguments: "DEVICE($D,$X) ERASE(DEVICE,PROTECTISP) HEXFILE(#H,NOCHECKSUMS,NOFILL,PROTECT

□ Run Independant

Fig 2. Options for the EPM900

3.1 Set-up

In the 'Settings' menu you can select the device you want to program from the Device pull-down menu. The programming mode will be ICP by default on devices that only have ICP programming capability. The parts can be hooked up one-to-one from the EPM900 to the application board as shown in Figure 1. Connect all 5 ICP pins and then program with the EPM900.



Configure EPM900 Flash Programmer

Download Function

☑ Erase

☑ Program

☑ Verify

□ Write Configuration

Device Options

Device: P89LPC906

Programming mode: ICP

Adapter: Pin to Pin

Device Configuration & Security

□ Use Configuration from START900.A51

BOOTVEC: 0x00

BOOTSTAT: 0x00

UCFG1: 0x63

Oscillator: Internal RC Oscillator

□ WDSE Watchdog Safety Enable

☑ BOE Brownout Detect Enable

☑ RPE Reset Pin Enable

□ WDTE Watchdog Timer Enable

SEC0: 0x00

SEC1: 0x00

SEC2: 0x00

SEC3: 0x00

Flash Erase

□ Global Erase all Flash

☑ Sector 0 (0x0000-0x00FF)

☑ Sector 1 (0x0100-0x01FF)

☑ Sector 2 (0x0200-0x02FF)

☑ Sector 3 (0x0300-0x03FF)

Fig 3. Settings for the EMP900 as ICP programmer

Table 2: ICP connection to EPM900 pin

ICP Pin	Hook up to EPM900 pin
P0.4/PDA programming Data pin	P0.4
P0.5/PCL programming clock pin	P0.5
P1.5/RST ICP mode entry pin	P1.5
V_{DD} needed for ICP mode entry	V_{DD}
V_{SS}	V_{SS}

When using the LPC901/902/903 make sure that the ‘Pin to Pin’ is selected in the ‘Adapter’ option.

When the EPM900 is hooked up to the Application board and the project you want to program into the microcontroller is open press the ‘LOAD’ button on the Build toolbar to start downloading code to the microcontroller.

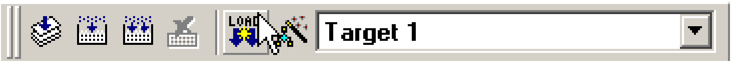


Fig 4. LOAD button on the Build toolbar

4. Modifying the MCB900 as ICP programmer

The MCB900 evaluation board from Keil Software can be modified to be able to work as an ICP programming device. Only 1 PNP transistor and an ICP ribbon cable to the application board is needed.

The LPC932 on the MCB900 can be programmed to interface between the ISP protocol used by the FlashMagic ISP programming software and the ICP protocol needed to program the LPC9xx.

4.1 Hardware set-up of the MCB900

Table 3 shows how to hook up the MCB900 to the ICP target board.

Table 3: ICP connection to MCB900 pin

ICP Pin	Hook up to MCB900 pin
P0.4/PDA programming Data pin	P0.4
P0.5/PCL programming clock pin	P0.5
P1.5/RST ICP mode entry pin	P1.6
V_{DD} needed for ICP mode entry	P2.3[1]
V_{SS}	V_{SS}

[1] To control V_{DD}, a PNP transistor is needed, because P2.3 cannot supply enough current for the LPC9xx during programming.

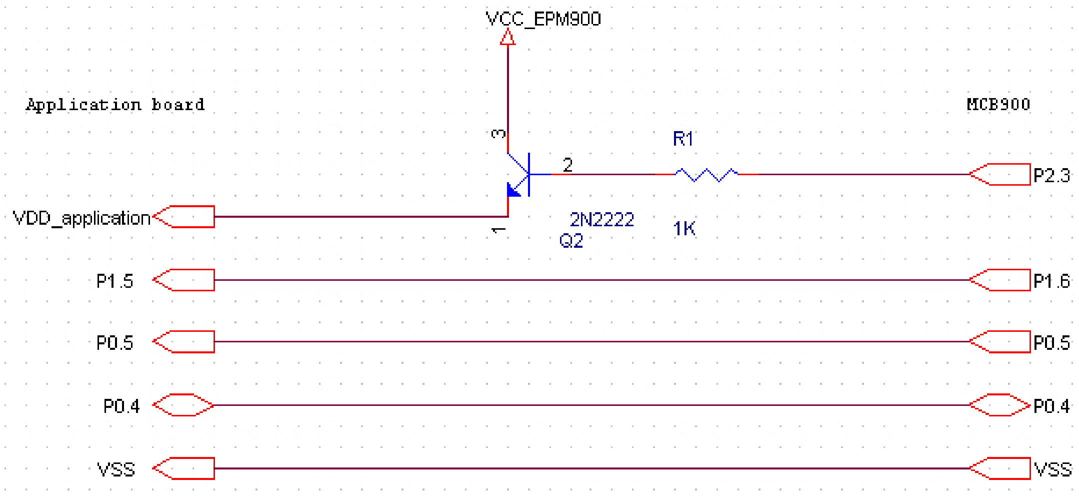


Fig 5. MCB900 hook-up

4.2 Programming the ISP-ICP bridge code

For FlashMagic to be able to interface to the ICP protocol the ISP commands from FlashMagic have to be converted into ICP commands. This can be done by programming the ISP-ICP bridge code that will do the translation from FlashMagic’s ISP commands to ICP commands.

Step 1:

- If not yet installed or if you have a version before 1.77, install the latest FlashMagic from: www.esacademy.com/software/flashmagic.

Step 2:

- Program the LPC932 device on the MCB900 board with the ISP-ICP bridge code using the FlashMagic software you installed in step 1.
- Set the jumper on the MCB900 board to the RESET position:

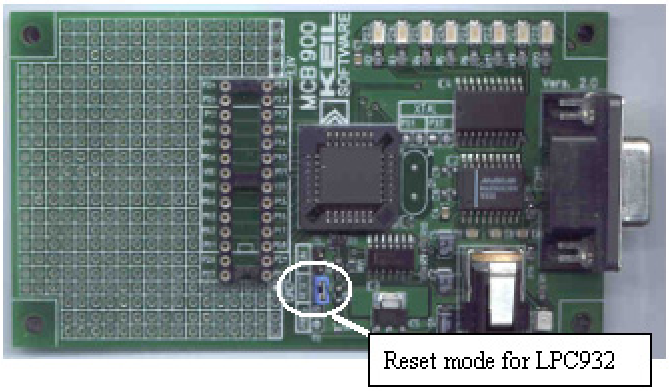
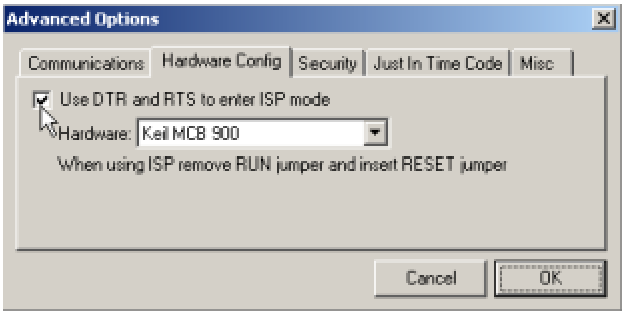


Fig 6. MCB900 in Reset mode

- Connect the MCB900 to your PC COM port using a serial cable.
- Power up the MCB900 board.
- Start FlashMagic (Start | Programs | FlashMagic | FlashMagic).
- Go to ‘Options | Advanced Options | Hardware Config’ and make sure the box ‘Use DTR and RTS to enter ISP mode’ is selected:



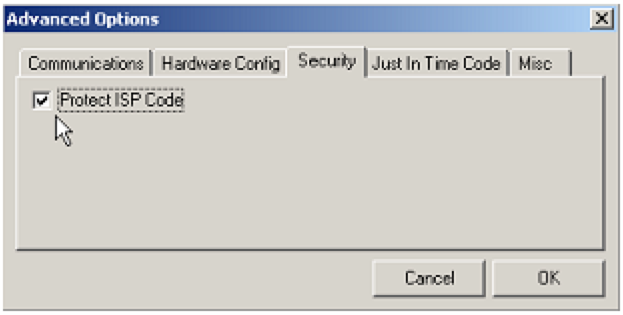
☑ Use DTR and RTS to enter ISP mode

Hardware: Keil MCB 900

When using ISP remove RUN jumper and insert RESET jumper

Fig 7. Use DTR and RTS

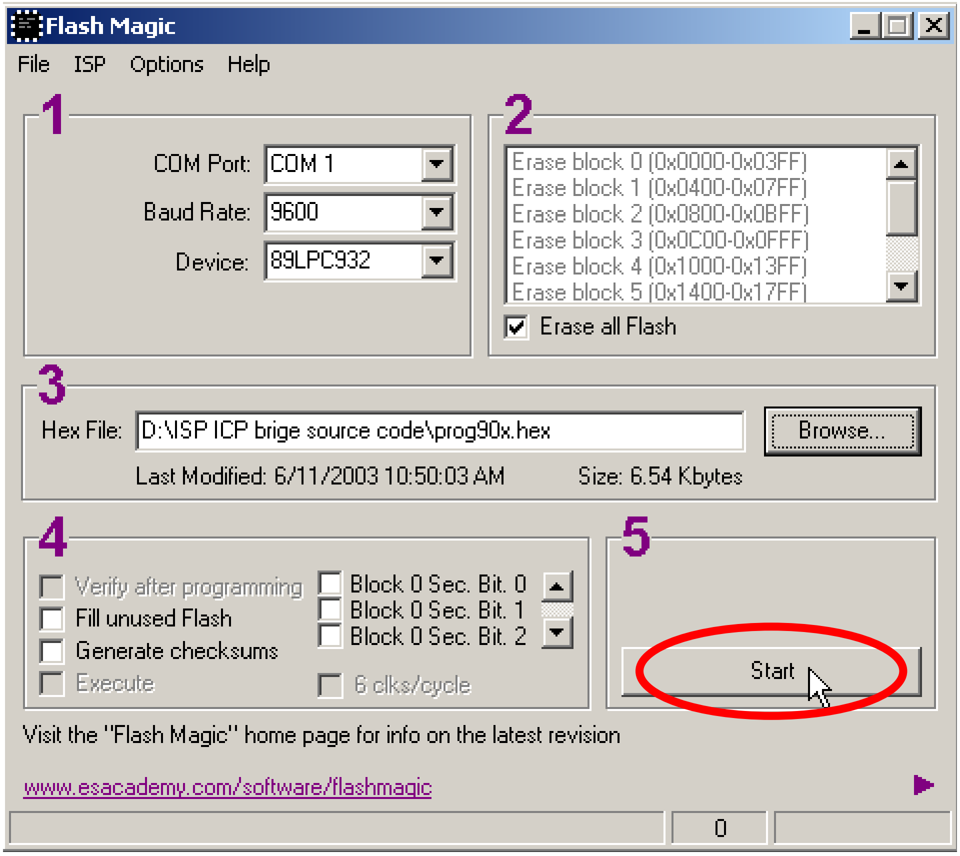
- Go to ‘Options | Advanced Options | Security’ and make sure the ‘Protect ISP code’ option is selected:



☑ Protect ISP Code

Fig 8. Protect ISP code

- Select the COM port you will be using to program the LPC932.
- Select the P89LPC932 as the device to be programmed.
- Select baud rate 9600.
- Browse to the prog90x.hex file (this first has to be unzipped from the ZIP that includes the appnote).
- Check the ‘erase all’ Flash box.
- Click start program the ISP-ICP bridge code into the P89LPC932:



COM Port: COM 1

Baud Rate: 9600

Device: 89LPC932

Erase block 0 (0x0000-0x03FF)
Erase block 1 (0x0400-0x07FF)
Erase block 2 (0x0800-0x0BFF)
Erase block 3 (0x0C00-0x0FFF)
Erase block 4 (0x1000-0x13FF)
Erase block 5 (0x1400-0x17FF)

☑ Erase all Flash

Hex File: D:\ISP ICP brige source code\prog90x.hex

Last Modified: 6/11/2003 10:50:03 AM Size: 6.54 Kbytes

□ Verify after programming
□ Fill unused Flash
□ Generate checksums
□ Execute

□ Block 0 Sec. Bit. 0
□ Block 0 Sec. Bit. 1
□ Block 0 Sec. Bit. 2
□ 6 clks/cycle

Visit the "Flash Magic" home page for info on the latest revision

www.esacademy.com/software/flashmagic

Fig 9. Start programming

4.3 Programming the LPC900

Once all setups from the previous chapter have been made, using FlashMagic the LPC900 devices can be programmed in 5 steps corresponding to the numbered steps on the FlashMagic User Interface.

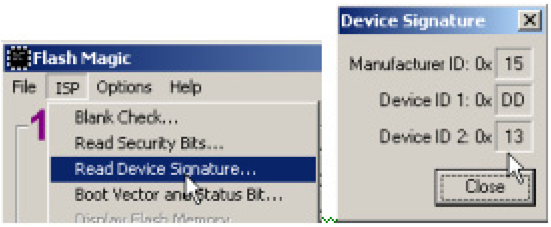
Step 0:

- Disconnect the power cable from the MCB900 board
- Make sure that the jumpers on the application board are set for ICP programming mode and reset your application board is connected properly to the MCB900 board.
- Plug the power cable back in to enter ICP mode on the LPC9xx (the ICP mode is only entered on power-up of the MCB900). LED P2.3 on the MCB900 board should light up.
- Start FlashMagic (Start | Programs | FlashMagic | FlashMagic).

Step 1:

- Select the COM port (make sure it is not already taken by another application, e.g. some PDA software)

- Select a baud rate of 19200 baud (the ISP-ICP bridge application has a fixed baud rate of 19200 baud)
- Select the LPC900 device you would like to program.
- At this point you can do a quick read of the device signature bytes to make sure you're connected to the LPC9xx:



Manufacturer ID: 0x 15

Device ID 1: 0x DD

Device ID 2: 0x 13

Fig 10. Checking signature bytes (example for LPC908)

If FlashMagic cannot read the Device Signature or you get all FF's, please check your setup. See chapter 5.1 for the correct ID bytes.

Step 2

Select which Flash blocks should be erased before the programming operation.

Step 3

Select the hex file to be programmed.

Step 4

Can be used to set security bits and fill unused Flash (optional)

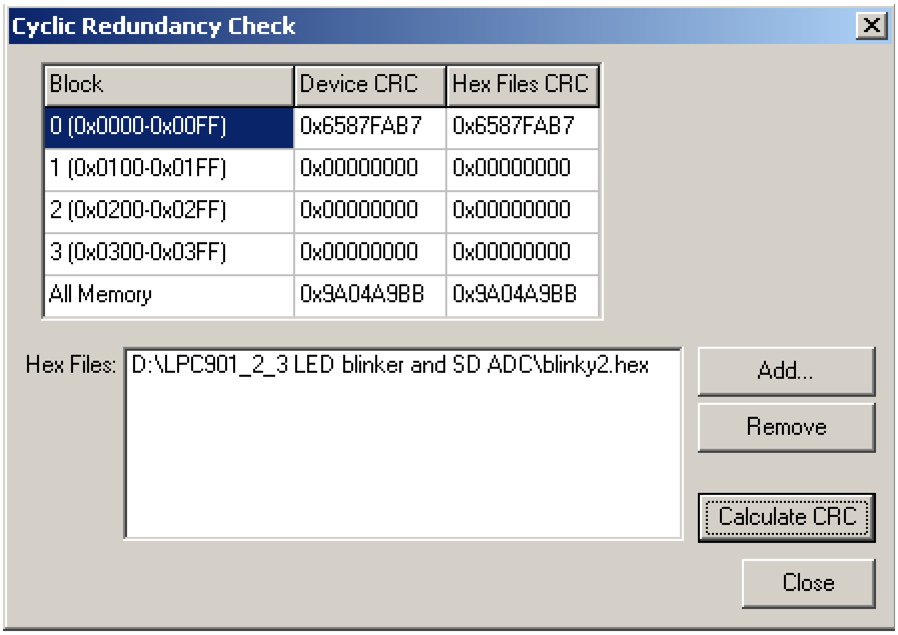
Step 5

Start programming!

4.4 Verification of the programmed code using CRCs

The LPC90x verifies using CRC (Cyclic Redundancy Check). The make sure the LPC90x has been programmed correctly, use the 'ISP | Cyclic Redundancy Check' option in FlashMagic.

For example, after the LED-Blinker code for the LPC901/2/3 is programmed into an LPC901 device, go to 'ISP | Cyclic Redundancy Check':



Cyclic Redundancy Check

Block	Device CRC	Hex Files CRC
0 (0x0000-0x00FF)	0x6587FAB7	0x6587FAB7
1 (0x0100-0x01FF)	0x00000000	0x00000000
2 (0x0200-0x02FF)	0x00000000	0x00000000
3 (0x0300-0x03FF)	0x00000000	0x00000000
All Memory	0x9A04A9BB	0x9A04A9BB

Hex Files: D:\LPC901_2_3 LED blinker and SD ADC\blinky2.hex

Fig 11. CRC check window

The Device CRC column gives you the CRC checksum for each block plus the Global CRC checksum for the whole device. As you can see from the image above, The Device CRC matches the corresponding Hex File CRC from the hex file on the CD ROM. This means that the device was programmed successfully. (Please note that the code for the LED Blinker on the LPC906/7/8 has a different CRC).

5. Devices that are supported with the ICP programming interface

5.1 8-pin packages

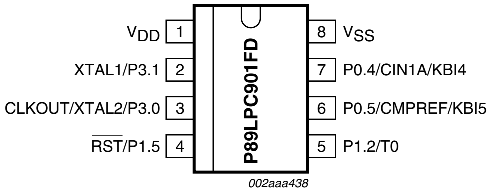


Fig 12. P89LPC901 pinout

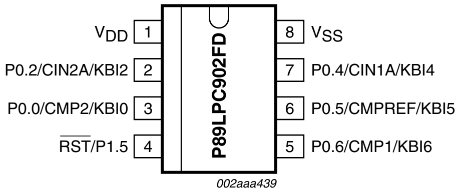


Fig 13. P89LPC902 pinout

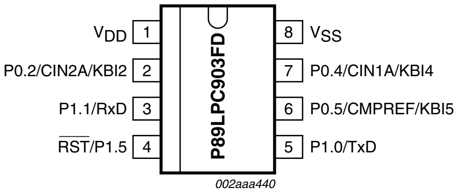


Fig 14. P89LPC903 pinout

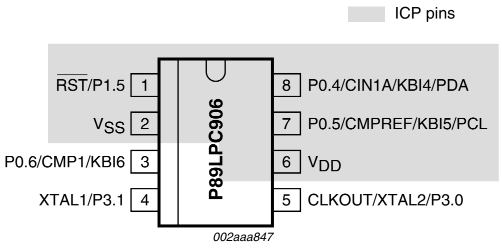


Fig 15. P89LPC906 pinout

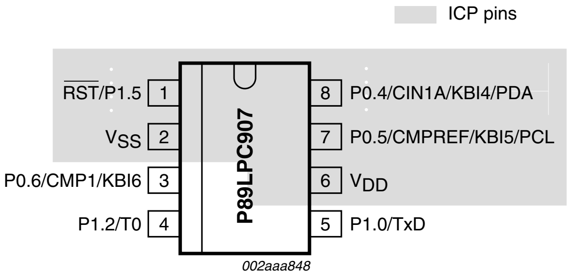


Fig 16. P89LPC907 pinout

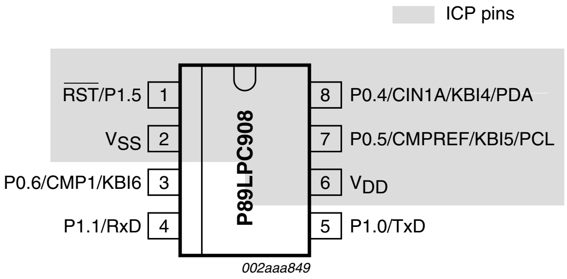


Fig 17. P89LPC908 pinout

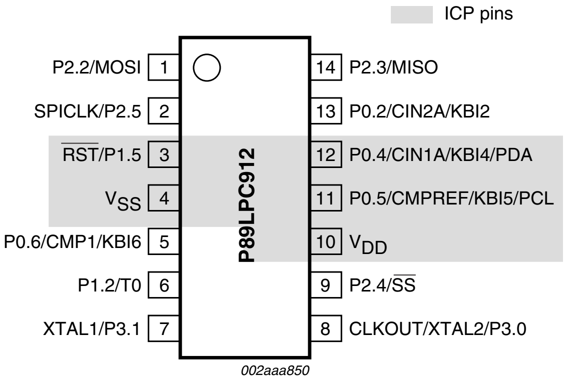


Fig 18. P89LPC912 pinout

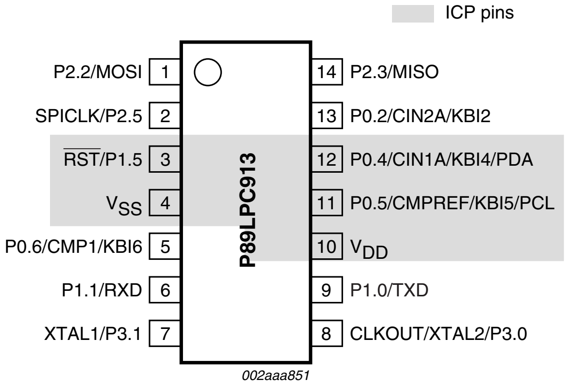


Fig 19. P89LPC913 pinout

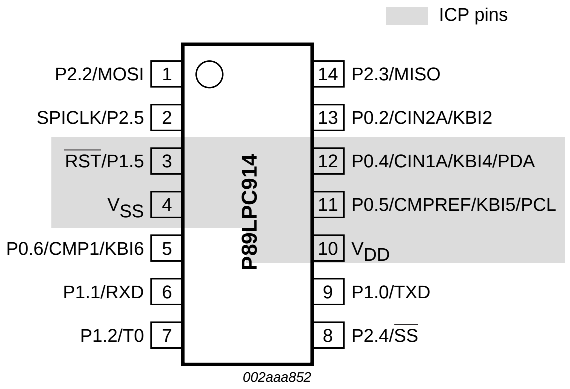


Fig 20. P89LPC914 pinout

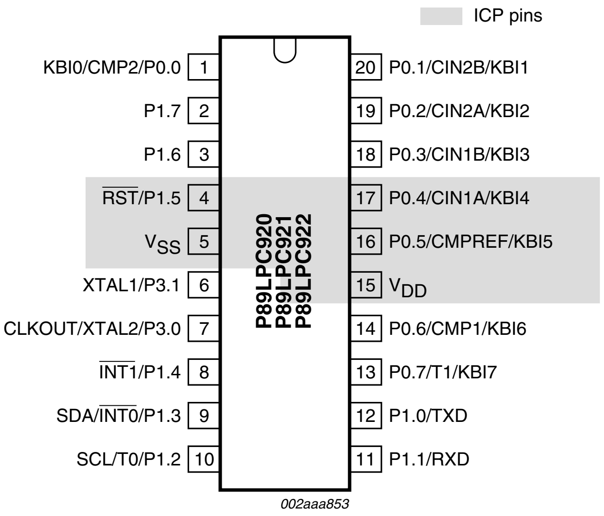


Fig 21. P89LPC922/921/920 pinout

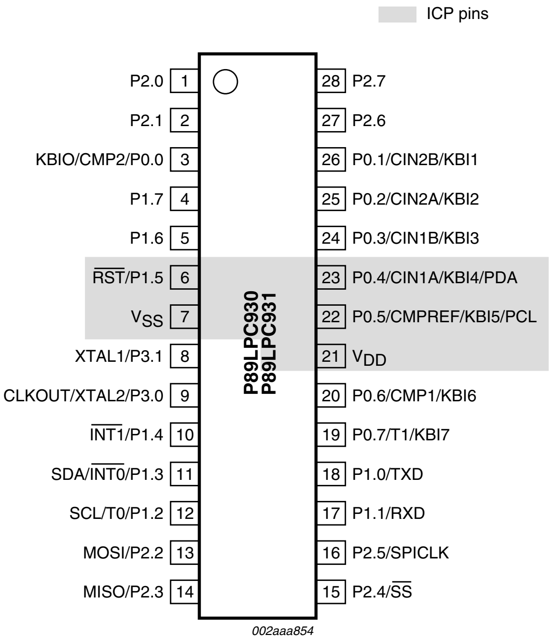


Fig 22. P89LPC930/931 pinout

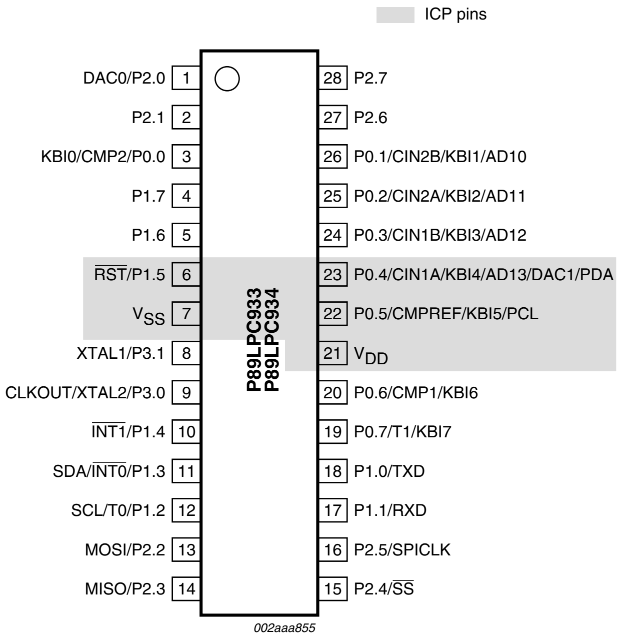


Fig 23. P89LPC933/934 pinout

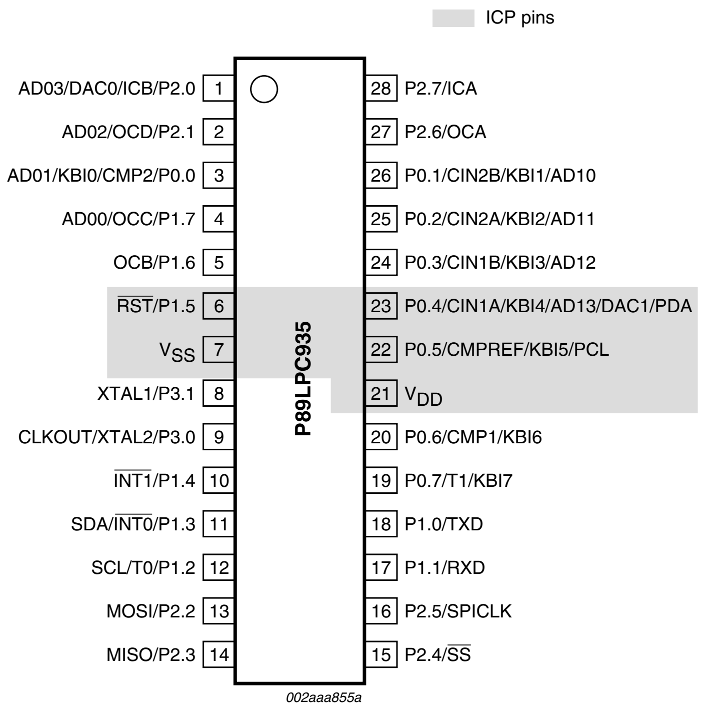


Fig 24. P89LPC935 pinout

5.2 Device ID bytes

Table 4: Device ID bytes

Device	MFGID	ID1	ID2
P89LPC901	15h	DD	0Dh
P89LPC902	15h	DD	0Fh
P89LPC903	15h	DD	10h
P89LPC906	15h	DD	11h
P89LPC907	15h	DD	12h
P89LPC908	15h	DD	13h
P89LPC912	15h	DD	14h
P89LPC913	15h	DD	15h
P89LPC914	15h	DD	16h
P89LPC920	15h	DD	1Ah
P89LPC921	15h	DD	0Bh
P89LPC922	15h	DD	0Ch
P89LPC930	15h	DD	19h
P89LPC931	15h	DD	09h
P89LPC933	15h	DD	A0h
P89LPC934	15h	DD	1Dh
P89LPC935	15h	DD	1Eh

6. Disclaimers

Life support — These products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. Philips Semiconductors customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Philips Semiconductors for any damages resulting from such application.

Right to make changes — Philips Semiconductors reserves the right to make changes in the products - including circuits, standard cells, and/or software - described or contained herein in order to improve design and/or performance. When the product is in full production (status ‘Production’), relevant changes will be communicated via a Customer Product/Process Change Notification (CPCN). Philips Semiconductors assumes no responsibility or liability for the use of any of these products, conveys no licence or title under any patent, copyright, or mask work right to these products, and makes no representations or warranties that these products are free from patent, copyright, or mask work right infringement, unless otherwise specified.

7. Contents

Date of release: 21 October 2004
Document number: 9397 750 14227

Published in U.S.A.