MSP430 Gang Programmer (MSP-GANG430)

User's Guide

Literature Number: SLAU101M
March 2003–Revised October 2008

Contents

Preface **7**

1 Introduction, Installation, and Setup **9**

1.1 Introduction 10

1.2 Software Installation 10

1.3 Hardware Installation 11

2 Operation **13**

2.1 Programming MSP430 Flash Devices Using the GUI 14

2.1.1 Procedure 14

2.1.2 Description of the MSP-GANG430 GUI 16

2.1.3 Status Messages 17

2.1.4 Error Messages 18

2.1.5 Description of the Gang430.ini File 19

2.1.6 Target Connector Functional Check 20

2.2 Programming MSP430 Flash Devices in Stand-Alone Mode 21

2.3 Programming MSP430 Flash Devices With User-Configured UART Handler 21

2.4 Programming the MSP430 Devices With GANG430.DLL 21

3 Firmware **23**

3.1 Commands 24

3.2 Firmware Interface Protocol 24

3.3 Synchronization Sequence 24

3.4 Data Frame 24

3.4.1 Frame Structure 25

3.4.2 Checksum 25

3.5 Detailed Description of Commands 27

3.5.1 General 27

3.5.2 Load Parameters 27

3.5.3 Start 28

3.5.4 Transmit Diagnostic 28

3.5.5 Erase Image 28

3.5.6 Load Image Block 28

3.5.7 Write Target Selective 28

3.5.8 Read Target Selective 29

3.5.9 Set Target V_{CC} 29

3.5.10 Load Image Checksum 29

3.5.11 Select Baud Rate 30

3.5.12 Execute Self Test 31

3.5.13 Set Signals 31

3.5.14 Select Image 31

3.5.15 Firmware Commands 31

4 Dynamic Link Library GANG430.DLL **33**

4.1 GANG430.DLL Description 34
4.2 Return Values/Error Codes From GANG430.DLL 54

5 Hardware 55
5.1 Specifications 56
5.2 Programming Times vs Code Size for the Gang Programmer 56
5.3 Recommendations for Target Connections 57
5.4 MSP-GANG430 Target Connector Signals 57
5.5 MSP-GANG430 Schematics 61
5.6 MSP-GANG430 Component Locations 62
5.7 Gang_Exp Target Expansion Board Layout 63

6 Schematics 65
6.1 Schematics 66
6.2 Revision History 72

A Hex File Formats 73
A.1 Intel Hex Object Format 74
A.2 TI-txt File Format 75

B Frequently Asked Questions 77
B.1 FAQs 78

C Preproduction Checklist 81

Document Revision History 82

Important Notices 83

List of Figures

1-1 MSP-GANG430 Gang Programmer 12
2-1 MSP-GANG430 GUI 15
5-1 25-Pin Sub-D at MSP-GANG430 58
5-2 14-Pin Connector at End of Interconnect Cable 58
5-3 Signal Connections for 4-Wire JTAG Communication 60
5-4 Signal Connections for 2-Wire JTAG Communication (SpyBiWire) 61
5-5 MSP-GANG430 Component Locations 62
5-6 Gang_Exp Layout 63
6-1 MSP-GANG430 Schematic (1 of 5) 66
6-2 MSP-GANG430 Schematic (2 of 5) 67
6-3 MSP-GANG430 Schematic (3 of 5) 68
6-4 MSP-GANG430 Schematic (4 of 5) 69
6-5 MSP-GANG430 Schematic (5 of 5) 70
6-6 MSP-GANG430 Expansion Board Schematic 71
A-1 Intel Hex Object Format 74

List of Tables

2-1 Function Buttons and Descriptions 16
2-2 Status Messages 17
2-3 Error Messages 18
2-4 Values for MiscErrors 20
2-5 Control Bit Definitions for Target Connector Pins 20
3-1 Data Frame of Firmware Commands 26
4-1 Return Values/Error Codes 54
5-1 MSP-GANG430 Hardware Specifications 56
5-2 MSP-GANG430 Target Connector Signal Functions 59
5-3 MSP-GANG430 Signal Levels 59
C-1 Preproduction Checklist for the Gang Programmer 81

All trademarks are the property of their respective owners.

Read This First

About This Manual

This user's guide documents the MSP430 gang programmer MSP-GANG430.

How to Use This Manual

This document contains the following chapters:

- Chapter 1 – Introduction, Installation, and Setup
- Chapter 2 – Operation
- Chapter 3 – Firmware
- Chapter 4 – Dynamic Link Library GANG430.DLL
- Chapter 5 – Hardware
- Chapter 6 – Schematics

Information About Cautions and Warnings

This book may contain cautions and warnings.

CAUTION

This is an example of a caution statement.

A caution statement describes a situation that could potentially damage your software or equipment.

WARNING

This is an example of a warning statement.

A warning statement describes a situation that could potentially cause harm to you.

The information in a caution or a warning is provided for your protection. Please read each caution and warning carefully.

If You Need Assistance...

Support for the MSP430 device and the MSP-GANG430 is provided by the Texas Instruments Product Information Center (PIC). Contact information for the PIC can be found on the Texas Instruments web site at www.ti.com. Additional device-specific information can be found on the MSP430 web site at www.ti.com/msp430.

Introduction, Installation, and Setup

This chapter introduces the MSP-GANG430 and guides you through the installation of the software and hardware.

Topic **Page**

1.1 Introduction **10**
1.2 Software Installation **10**
1.3 Hardware Installation **11**

1.1 Introduction

The MSP-GANG430 is an MSP430 device programmer that can program up to eight identical MSP430 flash devices at the same time. The MSP-GANG430 connects to the host PC using a standard RS232 serial connection. The MSP-GANG430 provides flexible device programming options as described in Chapter 2.

The MSP-GANG430 is not a gang programmer in the traditional sense; i.e., there are not eight sockets provided to program the target devices. Instead, the MSP-GANG430 is designed to connect to the target devices in-circuit (i.e., the target devices are mounted in the final circuit/system). The MSP-GANG430 accesses the target devices using connectors to the JTAG signals.

Chapter 6 contains a schematic that shows how the signals from the MSP-GANG430 can be brought out to each of the target devices via an MSP430-standard JTAG connector. The circuit could easily be modified to connect the signals to the target device pins directly (via a socket) if a traditional gang programmer is desired.

The MSP-GANG430 is provided with an expansion board that implements the interconnections between the MSP-GANG430 and multiple target devices. Eight cables are provided that connect the expansion board to eight target devices (via a JTAG connector).

1.2 Software Installation

Always use the latest software version, which can be downloaded from the MSP430 web site at www.msp430.com.

To install the MSP-GANG430 software:

1. Insert the MSP430 CD-ROM into the CD-ROM drive of the computer. Setup automatically opens the default browser of the computer and displays the MSP430 start page.

   If the start page does not open automatically, open the file index.htm located in the root directory of the CD-ROM using a web browser. The MSP430 start page is displayed in the browser window.
2. Select Tools & Software.
3. Select MSP-GANG430 Gang Programmer.
4. Select Save this program to disk under the File Download options and click OK. A Save As dialog is displayed.
5. Using the Save As dialog, save the GANG430_Rxxx.exe program to the computer. Make a note of the directory path to this file.
6. Navigate to this file (GANG430_Rxxx.exe) and run it. A welcome message is displayed on the screen.
7. The setup program guides you through the installation process. Follow the setup instructions on the screen.
8. On completion of the setup program execution, the MSP-GANG430 program icons are created in the selected folder. Click on the GANG430 Read Me First icon to obtain important information about the MSP-GANG430 hardware and software.
9. The appropriate program group and icons are added to the Windows desktop.
10. To start the MSP-GANG430 software, click the GANG430 icon in the selected program group (the default program group is ADT430).

1.3 Hardware Installation

To install the MSP-GANG430 hardware:

1. Connect the MSP-GANG430 to the serial port (COM1 to COM15 or extension board) of the PC using the 9-pin Sub-D connector.
2. Connect an external power supply to the MSP-GANG430. The voltage of the power supply must be between 9 V and 15 V dc and must be capable of providing a minimum current of 300 mA. The center post of the power supply connector on the MSP-GANG430 is the positive-voltage terminal.

   The power supply connection status is indicated by the yellow system LED on the MSP-GANG430.
3. Attach the expansion board to the 25-pin Sub-D connector on the MSP-GANG430. The expansion board provides connectivity for up to eight targets, using the included 14-pin cables. The target MSP430 flash devices can be in stand-alone sockets or can be on an application's PCB, and they are accessed via the JTAG signals.

Note: Maximum Signal Path Length: 60 cm

The maximum length of the signal path between the 25-pin Sub-D connector on the MSP-GANG430 and a target device is 60 cm. If cables between the programmer and targets are longer than 20 cm, serial resistors of 100 Ω should be inserted on signals TCK and TMS on the expansion board. TCK is on pin 4 and TMS is on pin 3 of the 25-pin Sub-D connector.

4. The MSP-GANG430 provides the selected power-supply voltage V_{CC} (VCC_MSP on pins 7 and 14 of the 25-pin Sub-D target connector; pin 2 on the 14-pin cable) to the target devices in application PCBs. A maximum total current of 150 mA can be provided by the programmer to the targets.

   When an external supply voltage V_{CC} is used for the target systems, it also should be connected to the sense input pin (MSP_VCC_IN on pin 15 of the 25-pin Sub-D target connector; pin 4 on the 14-pin cable) of the programmer. This connection allows the integrated level shifters to match the target JTAG signal levels with the external supply voltage.

   If the expander PCB (25-pin Sub-D to 8 x 14-pin header) is used, the external supply voltage must be connected to pin 4 of the 14-pin header at position 1.

CAUTION

When an external supply voltage is used for the target systems, it is important to disconnect VCC_MSP from the targets to avoid power-supply conflicts that could potential damage the MSP-GANG430 and the targets. Connect the external supply voltage to MSP_VCC_IN.

It is not possible to simultaneously use the MSP-GANG430 with targets powered by the MSP-GANG430 and with targets powered by an external supply.

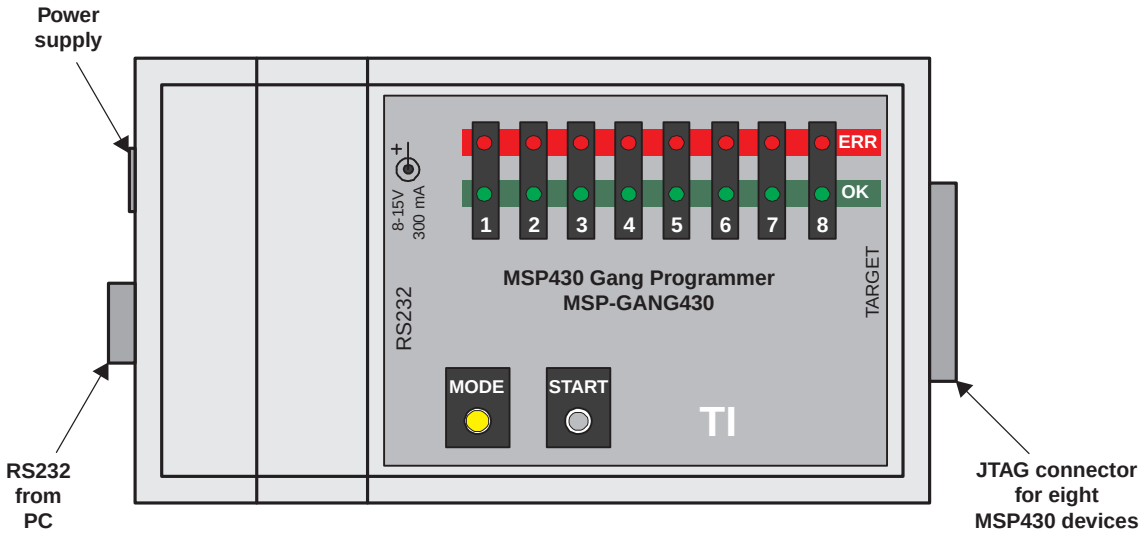


Figure 1-1. MSP-GANG430 Gang Programmer

Operation

This chapter describes how to program MSP430 flash devices using the MSP-GANG430.

Topic **Page**

2.1 Programming MSP430 Flash Devices Using the GUI 14
2.2 Programming MSP430 Flash Devices in Stand-Alone Mode 21
2.3 Programming MSP430 Flash Devices With User-Configured UART Handler..... 21
2.4 Programming the MSP430 Devices With GANG430.DLL..... 21

2.1 Programming MSP430 Flash Devices Using the GUI

2.1.1 Procedure

Use the following sequence to start the GUI and program MSP430 flash devices using the MSP-GANG430.

1. Connect the MSP-GANG430 hardware and the targets as described in Section 1.3.
2. Click on the GANG430 icon located in the program group specified during installation of the software (the default group is ADT430). The MSP430 FLASH Gang Programmer GUI is displayed on the screen (see Figure 2-1).

   If a Firmware Update dialog appears, see FAQ #3 in Section B.1. The status line in the GUI displays the message "MSP-GANG430 Gang Programmer connected." If this message is not displayed, check the COM Port selection in the communication settings and the MSP-GANG430 connections.
3. Select the required device using the Device Type menu.
4. Select the object code file to be programmed into the device(s) using the File Name menu. The formats supported for the object code file are TI TXT (.txt) and Intel hex (.a43, .hex, .int).
5. Use the Load Image button to download the object code file and its checksum to the MSP-GANG430. The data is written to an internal memory called the image buffer.
6. Select the supply voltage using the Supply Voltage menu.
7. Select the options in Main Process as required.
8. Click on the Start button in the Main Process section to start the programming operation. The progress and completion of the operation are displayed in the Status section.

   Before the gang programming operation starts, a reset of all targets is performed. The duration of the reset pulse and the delay after reset are specified by Settling Time. The reset feature is available on hardware V1.30 and later.

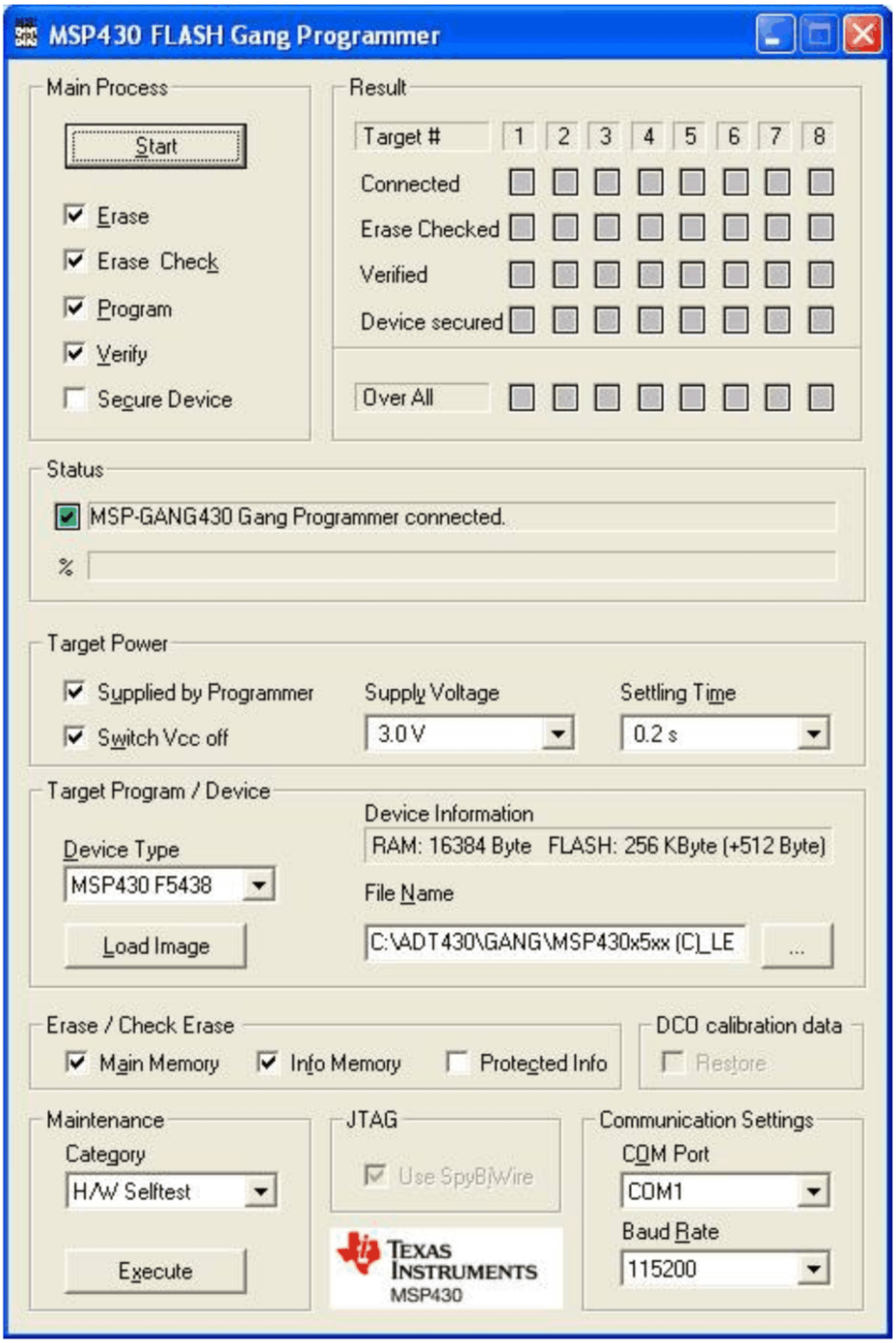


Figure 2-1. MSP-GANG430 GUI

2.1.2 Description of the MSP-GANG430 GUI

Table 2-1 describes the function buttons and option settings in the MSP-GANG430 GUI.

Table 2-1. Function Buttons and Descriptions

Button Group	Functions	Description
Main Process	Start	Executes the function(s) selected in Main Process
	Erase	Erases the target's flash memories as selected on the panel Erase/Check Erase
	Erase Check	Checks if the target's memories are erased as selected on the panel Erase/Check Erase
	Program	Programs the image buffer into the target's flash memory
	Verify	Compares the target's flash memory and the image buffer
	Secure Device	Makes MSP430 devices secure after successful completion of the selected operations (permanently disables device access through JTAG)
Target Power	Supplied by Programmer	The connected targets are powered by the MSP-GANG430 via VCC_MSP pin.
	Switch Vcc Off	The supply voltage provided by the VCC_MSP pin is switched off after programming the targets.
	Supply Voltage	Selects the voltage supplied to the targets
	Settling Time	Specifies the time to allow the target capacitors to fully charge
Target Program/Device	Load Image	Loads the selected object code file into the image buffer
	Device Type	Selects the memory model of the target
	Device Information	Shows memory details of the selected device type
	File Name	Selects the file containing the object code
Erase/Check Erase	Main Memory	Select to specify that the main memory should be erased or checked according to setting in group Main Process.
	Info Memory	Select to specify that the info memory should be erased or checked according to setting in group Main Process.
	Protected Info	Select to specify that the protected info memory (INFO A) should be erased and/or checked according to setting in group Main Process. If this check box is checked and Restore check box unchecked, the DCO calibration data will be lost.
DCO Calibration Data	Restore	Select to specify that the DCO calibration data should be restored after protected segment (INFO A) is erased. If unchecked, the calibration DCO data will be lost. It can be used if the calibration data is not needed in the application. Can be used if the Protected Info is checked.
Maintenance	Execute	Starts the operation that is selected in Category
	H/W Self Test	Initiates hardware self test All green LEDs on = test pass. All red LEDs on = test fail.
	Update Firmware	Updates MSP-GANG430 firmware with the firmware update file
	Read out Target	Reads out exclusively one of eight target modules
	Erase Image	Erases the image buffer
	Target Connector (Test)	Sets the selected pin on the 25-pin target connector of the MSP-GANG430
JTAG	Use SpyBiWire	Check this option if the 2-wire SpyBiWire interface should be used for device operation. This option is not available for all devices. **Note:** The 2-wire SpyBiWire interface operation is slower than the regular 4-wire JTAG.
Communication Settings	COM port	Selects the PC serial port used to communicate with the MSP-GANG430
	Baud Rate	Selects the baud rate of the serial port

Table 2-1. Function Buttons and Descriptions (continued)

Button Group	Functions	Description
System Menu	System Menu Mode	The System Menu is displayed by clicking on the icon at the upper left corner of GUI (or {Alt} and {Space} pressed together).
	Lock Settings	Locks all settings on the GUI. Only the Start button, Result group box, and Status group box are displayed. Other group boxes are hidden.
	Unlock Settings	The locked settings and the hidden group boxes are enabled.
	Help	Opens the user help window
	About Gang Programmer	Displays the desktop application software and firmware version and filename information (or press F1)

2.1.3 Status Messages

Table 2-2 shows the status messages that can be displayed in the MSP-GANG430 GUI during operation.

Table 2-2. Status Messages

Status Type	Status Message
Progress	Executing main process...
	Erasing image buffer...
	Loading image buffer...
	Loading checksum of image buffer...
	Erasing firmware section...
	Loading firmware section...
	Initializing firmware...
	Changing baud rate...
	Executing self test...
	Setting signals at target connector...
System	Operation successful
	MSP-GANG430 gang programmer connected.
	Result of the operation: See the result section.
Setting	COM port settings modified. MSP-GANG430 gang programmer connected.
	Target power settings modified
Maintenance	Operation successful
	Selftest passed

2.1.4 Error Messages

Table 2-3 shows the error messages that can be displayed in the MSP-GANG430 GUI when an error occurs during operation.

Table 2-3. Error Messages

Error Type	Error Message
Communication	ERROR: Unable to read Target!
	ERROR: Unable to open COM port – already in use?
	ERROR: Unable to close COM port!
	ERROR: Synchronization failed. Programmer connected?
	ERROR: Time out during operation – Correct COM port selected?
	ERROR: Communication – Frame has errors!
Setting System	ERROR: Select baud rate command not accepted!
	WARNING: Target voltage too low for erase / program operation! Set to > 2.7-V.
	WARNING: Could not set target voltage – Externally supplied?
	ERROR: Value wrong or out of range!
	ERROR: Selected file is of unrecognizable format!
	ERROR: Selected file is not a firmware update file!
	ERROR: Firmware section cannot be erased!
	ERROR: Unable to load firmware. Correct file format?
	ERROR: Unable to refresh gang programmer system parameters!
	ERROR: Unable to load Image buffer! Image buffer erased? Correct file format?
	ERROR: Result of operation not received!
	FATAL ERROR: Image memory corrupted or erased!
	ERROR: Unexpected end of file!
	ERROR: File contains invalid record!
	ERROR: Wrong target device connected!
	ERROR: Image range XXXX–YYYY placed outside of existing memory
	ERROR: No target device connected
	General error!
Maintenance	ERROR: Self test failed – No access to one ore more image buffer devices!
	ERROR: Self test failed – No JTAG access to one or more target channels!
	ERROR: Self test failed – Target voltage generator (VCCT) does not work properly!
	ERROR: Self test failed – System voltage (VCC) not in range!
	ERROR: Self test failed – Blow fuse voltage (VPP) not in range!
Windows	ERROR: Unable to open file!
	ERROR: Error during file I/O!
	ERROR: Unable to open INI file!

2.1.5 Description of the Gang430.ini File

The Gang430.ini file contains the MSP-GANG430 default settings and the most recently used GUI settings.

2.1.5.1 Section [System]

The most recently used settings of the MSP-GANG430 GUI are stored in the Gang430.ini file when the GUI application exits. This information is stored under the [System] section of the file. There are few entries in this section that cannot be changed via the GUI.

JtagSpeed = 0 (default value is 0)

Set JtagSpeed = 1 to make the JTAG connection slower. This can be used for quality testing of the cable between target devices and the MSP-GANG430.

VerifyMarginal = 1 (default value is 1)

Set VerifyMarginal = 0 to avoid the marginal check on devices with the marginal read feature. It can be used to reduce the verification time. For more information, see the *MSP430x2xx Family User's Guide* or *MSP430x4xx Family User's Guide*.

2.1.5.2 Section [User]

The initial settings for the following parameters are in the [User] section and may be modified by the user.

CaptionIndexed = 0 (default). The caption of the MSP-GANG430 GUI in the default state is *MSP430 FLASH Gang Programmer*.

CaptionIndexed = 1. The caption is extended with a suffix [n] to allow multiple-instance operation for programming in automated production environment that uses caption titles for process control. The suffix n represents the unique COMn port number for that instance. For example, for COM3, the caption is displayed as *MSP430 FLASH Gang Programmer [3]*.

2.1.5.3 Section [Timing]

The [Timing] section shows the approximate time, in milliseconds, that has elapsed for various operations of the MSP-GANG430. The parameters under this section are updated after successful completion of the associated operations. The information on the elapsed time is useful for development and setup. The parameter that is most often useful is the time taken to complete a gang programming cycle, which is reported in the TimeGangProg key in this section.

2.1.5.4 Section [Diagnostic]

The [Diagnostic] section is useful for automatic production using the MSP-GANG430. During gang programming operation, Busy = 1. After completion of the operation, the information in the Result group box in the GUI is also listed in this section for the following: Connected, EraseChecked, Verified, SecureDevice, OverAll, and MiscErrors, if any. A Busy = 0 terminates that process and the diagnostics can be read from the .ini file.

For example:

OverAll = 239 (0xEF) indicates that all channels except channel 5 were successfully processed (channel 1 represents the LSB, channel 8 represents the MSB of a byte).

Note: When MiscErrors is nonzero, the diagnostic values are not valid for evaluation. Possible values for MiscErrors are shown in Table 2-4.

Table 2-4. Values for MiscErrors

MiscErrors	Description
0	Operation successful
1	FATAL ERROR: Image memory could be corrupted or erased
11	ERROR: Main process parameters not yet set
16	ERROR: Could not set target voltage (VCCT) – MSP_VCC_IN pin connected?

2.1.6 Target Connector Functional Check

The MSP-GANG430 GUI diagnostics support a functional check of the target connector pins. The functional check is invoked by selecting the Target Connector category in the Maintenance group box, and clicking the Execute button with the appropriate Set Signals defined. The Set Signals value is the decimal value that represents the combination of the appropriate control bits. The defined pin's signal level is available for measurement after execution of the functional check. The signal levels remain unchanged until another configuration is set. The target connector functional check always must be terminated by executing the End of Check setting (0x8000 = 32768) or by executing an hardware self test to initialize the MSP-GANG430. Table 2-5 shows the details of the control bits.

Table 2-5. Control Bit Definitions for Target Connector Pins

Bit	Pin Function and Level	Signal Level	Connector Pins
0x0000	Select TDO/TDI1	–	1, 8
0x0001	Select TDO/TDI2	–	9
0x0002	Select TDO/TDI3	–	10
0x0003	Select TDO/TDI4	–	11
0x0004	Select TDO/TDI5	–	12
0x0005	Select TDO/TDI6	–	13
0x0006	Select TDO/TDI7	–	25
0x0007	Select TDO/TDI8	–	24
0x0008	Logic level at TDO/TDIx pin Logic level at TDI/VPP pin	V_{CC} or 0 V_{CC}/VPP or 0	1, 8–13, 25, 24 2
0x0010	Logic level at TCK pin	V_{CC} or 0	4
0x0020	Logic level at TMS pin	V_{CC} or 0	3
0x0040	Logic level at TEST/VPP pin	V_{CC} – 0.25 V or 0	17
0x0080	Logic level at TDI/VPP pin	V_{CC} or 0	2
0x0100	VCC on/off at VCC_MSP pin	V_{CC} or 0	7, 14
0x0200	VPP on/off at TDI/VPP pin, overwrites bit 0x0080	VPP or 0	2
0x0400	VPP on/off at TEST/VPP pin	VPP or 0	17
0x0800	RESET activate/deactivate on RST pin	V_{CC} or 0	6
0x7000	Reserved		
0x8000	End of check, reinitialize MSP-GANG430	Initial	All

Example 1

To check the logic-high level on TDO/TDI3 and VPP voltage level on TEST/VPP:

Enter 0x040A (0x0002 + 0x0008 + 0x0400) into Set Signal and click Execute.

The V_{CC} at pin 10 and V_{PP} at pin 17 can be measured now. All other pins remain at 0 V. The connector case is electrically grounded, as are pins 19 to 23.

Example 2

To let free run target 3 programmed via SpyBiWire (2-wire mode):

Select TDO/TDI3 = 0x0002

Logic level at TDO/TDIx = 0x0008

VCC on at VCC_MSP pin = 0x0100

Enter 0x010A (0x0002 + 0x0008 + 0x0100) into Set Signal and click Execute.

2.2 Programming MSP430 Flash Devices in Stand-Alone Mode

The MSP-GANG430 supports a stand-alone mode of programming the target MSP430 flash devices.

The programming options for the MSP-GANG430 while it operates in stand-alone mode are configured using the GUI. When the Start button in Main Process is clicked, all selected options are downloaded into the flash memory of the MSP-GANG430. Once the download is complete, the MSP-GANG430 can be disconnected from the PC and used stand-alone to program the target devices. The Start button on the MSP-GANG430 operates the same as the Start button of the GUI. Note that after performing additional MSP-GANG430 operations using the GUI, the options to be used for stand-alone programming need to be re-downloaded into the flash memory of the MSP-GANG430 by pressing of the Start button of the GUI. Note that after performing additional MSP-GANG430 operations using the GUI, the options to be used for stand-alone programming need to be downloaded again into the flash memory of the MSP-GANG430. Progress of the operation in stand-alone mode is indicated by the flashing yellow LED.

The result status is represented by the rows of green and red LEDs on the MSP-GANG430. For each channel, a green LED indicates a successful operation, and a red LED indicates a failed operation. If both red and green LEDs are off for a channel, the target device connected to that channel is not accessible or there is no target device.

To start the hardware self test of the MSP-GANG430 while in stand-alone mode, press and hold the Start key for more than three seconds. The progress of the test is indicated by alternately flashing green and red LEDs in groups of four. If the self test passes, all eight green LEDs are on. Any other combination of the LEDs indicates that the self test failed.

Note: If the GANG430.DLL is used for configuration of MSP-GANG430 in stand-alone mode, the MSP-GANG430 works with image #0 (default) after power on. Thus, the programming options and the program code should be downloaded into image #0. This is not relevant when using the GUI gang430.exe, because it uses only image #0.

2.3 Programming MSP430 Flash Devices With User-Configured UART Handler

The MSP-GANG430 programmer supports programming the MSP430 flash devices using generic firmware commands. The generic firmware commands and protocol are described in Chapter 3.

2.4 Programming the MSP430 Devices With GANG430.DLL

The MSP-GANG430 programmer supports programming the MSP430 flash devices by using DLL functions. The usage of the DLL functions and their prototypes (calling conventions) are described in Chapter 4.

Firmware

This chapter describes the firmware commands and protocol for controlling the MSP-GANG430 via its RS-232 serial communication interface.

Topic **Page**

3.1 Commands 24
3.2 Firmware Interface Protocol 24
3.3 Synchronization Sequence 24
3.4 Data Frame 24
3.5 Detailed Description of Commands 27

3.1 Commands

The MSP-GANG430 can be controlled via firmware commands received through its RS-232 serial port.

The following firmware commands are supported:

- Load Parameters
- Start
- Transmit Diagnostics
- Erase Image
- Load Image Block
- Load Image Checksum
- Erase Firmware
- Load Firmware Block
- Finalize Firmware
- Select Baud Rate
- Execute Self Test
- Set Signals (at target connector)
- Read Target Selective
- Write Target Selective
- Set Target V_{CC}
- Select Image

3.2 Firmware Interface Protocol

The MSP-GANG430 supports the following UART communication protocol:

- Baud rates from 9600 to 115200 baud in half duplex mode. The default baud rate at startup is 115200 baud, and is the recommended baud rate.
- One start bit, eight data bits, even parity bit, and one stop bit
- Software handshake by (not)acknowledge character

3.3 Synchronization Sequence

To synchronize with the MSP-GANG430, the host serial handler transmits a SYNC character (0x80) to the MSP-GANG430. The MSP-GANG430 acknowledges successful reception of the SYNC by responding with a DATA_ACK character (0x90). If the SYNC is not received correctly, a DATA_NAK character (0xA0) is sent back. This sequence is required to establish the communication channel and/or react immediately to line faults.

Note: The synchronization character is not part of the data frame described in Section 3.4.

3.4 Data Frame

The data frame format follows the TI MSP430 serial standard protocol (SSP) rules, extended with a preceding synchronization sequence (SS), as described in Section 3.3. The MSP-GANG430 is considered the receiver in Table 3-1.

The redundancy of some parameters results from the adaptation of the SSP or to save boot ROM space.

3.4.1 Frame Structure

The data frame format of the firmware commands is shown in Table 3-1.

- The first eight bytes (HDR through LH) are mandatory (– represents dummy data).
- Data bytes D1 to Dn are optional.
- Two bytes (CKL and CKH) for checksum are mandatory.
- Acknowledge done by the MSP-GANG430 is mandatory except with the Transmit Diagnostic command.

3.4.2 Checksum

The 16-bit (2-byte) checksum is calculated over all received/transmitted bytes, B1 to Bn, in the data frame except the checksum bytes themselves by XORing words (two successive bytes) and inverting the result, as shown in the following formulas.

$$CHECKSUM = INV\ [\ (B1 + 256 \times B2)\ XOR\ (B3 + 256 \times B4)\ XOR\ \ldots\ XOR\ ((Bn - 1) + 256 \times Bn)\ ]$$

or

$$CKL = INV\ [\ B1\ XOR\ B3\ XOR\ \ldots\ XOR\ Bn{-}1\ ]$$

$$CKH = INV\ [\ B2\ XOR\ B4\ XOR\ \ldots\ XOR\ Bn\ ]$$

The example of the frame Execute Self Test with checksum:

0x80 0x35 0x06 0x06 0x00 0x00 0x00 0x00 0x00 0x00 0x79 0xCC

Table 3-1. Data Frame of Firmware Commands[(1)(2)]

Received MSP-GANG430 Firmware Command	HDR	CMD	L1	L2	A1	A2	A3	A4	LL	LH	D1	D2...Dn	CKL	CKH	ACK
Load Parameters	80	36	08	08	D1	D2	D3	D4	D5	D6	00	00	CKL	CKH	ACK
Start	80	31	06	06	00	00	–	–	00	00	00	00	CKL	CKH	ACK
Transmit Diagnostic	80	32	04	04	00	00	–	–	00	00	–	–	CKL	CKH	–
GANG430 responds on Transmit Diagnostic	80	00	1E	1E	D1	D2	–	–	D3	D4	D5	D06...D1E	CKL	CKH	–
Erase Image	80	33	06	06	00	00	–	–	00	00	00	00	CKL	CKH	ACK
Load Image Block	80	34	n	n	A1	A2	A3	00	n–4	00	D1	D2...Dn–4	CKL	CKH	ACK
Load Image Checksum	80	37	n	n	D1	D2	–	–	D3	D4	D5	D6...Dn	CKL	CKH	ACK
Execute Self Test	80	35	06	06	00	00	–	–	00	00	00	00	CKL	CKH	ACK
Set Signals	80	30	04	04	D1	D2	–	–	D3	D4	–	–	CKL	CKH	ACK
Erase Firmware	80	39	04	04	D1	D2	–	–	D3	D4	–	–	CKL	CKH	ACK
Load Firmware Block	80	3A	n	n	AL	AH	–	–	n–4	00	D1	D2...Dn–4	CKL	CKH	ACK
Finalize Firmware	80	3B	04	04	00	00	–	–	00	00	–	–	CKL	CKH	ACK
Select Baud Rate	80	38	06	06	D1	00	–	–	00	00	00	00	CKL	CKH	ACK
Write Target Selective	80	3C	n	n	A1	A2	A3	00	n–6	00	D1	D2...Dn–6	CKL	CKH	ACK
Read Target Selective	80	3D	08	08	A1	A2	A3	00	n	00	D1	D2	CKL	CKH	–
GANG430 responds on Read Target Selective	80	00	n	n	D1	D2	–	–	D3	D4	D5	D6...Dn	CKL	CKH	–
Set Target V_{CC}	80	3E	04	04	D1	D2	–	–	D3	D4	–	–	CKL	CKH	ACK
Select Image	80	50	06	06	D1	00	–	–	00	00	00	00	CKL	CKH	ACK

(1) All numbers are bytes in hexadecimal notation. ACK is sent by the MSP-GANG430. The synchronization sequence (TX = 0x80, RX = 0x90) is not part of the data frame. It must precede every frame.

(2)

HDR:	DATA_FRAME = 0x80 means data frame expected in accordance with TI MSP430 serial standard protocol (SSP).
CMD:	Command identification
L1, L2:	Number of bytes consisting of AL through Dn. The valid values of these bytes are restricted as follows: L1 = L2, L1 < 255, L1 even.
A1, A2, A3:	Block start address or erase (check) address or jump address LO/HI byte. The bytes are combined to generate a 24-bit word as follows: Address = A3 × 0x10000 + A2 × 0x100 + A1
LL, LH:	Number of pure data bytes (maximum 250) or erase information LO/HI byte or block length of erase check (maximum is 0xFFFF)
D1...Dn:	Data bytes
CLK, CKH:	16-bit checksum LO/HI byte
xx:	Can be any data
–:	No character (data byte) received/transmitted
ACK:	The acknowledge character returned by the MSP-GANG430 can be either DATA_ACK = 0x90 (frame was received correctly, command was executed successfully) or DATA_NAK = 0xA0 (frame not valid (e.g., wrong checksum, L1 ≠ L2), command is not defined, is not allowed, or was executed unsuccessfully).
n:	Number of bytes

3.5 Detailed Description of Commands

3.5.1 General

After the header byte HDR (0x80) and the command identification byte CMD, the frame length bytes L1 and L2 (which must be equal) hold the number of bytes following L2, excluding the checksum bytes CKL and CKH.

Bytes A1, A2, A3, A4, LL, LH, and D1 to Dn are command specific. However, the checksum bytes CKL (low byte) and CKH (high byte) are mandatory.

If the data frame is received correctly and the command execution is successful, the acknowledge byte ACK is send back by the MSP-GANG430 set to DATA_ACK = 0x90. Incorrectly received data frames, unsuccessful operations, and commands that are not defined are confirmed with a DATA_NAK = 0xA0. The data frame should be sent to the MSP-GANG430 within one second after the synchronization sequence (TX = 0x80, RX = 0x90).

3.5.2 Load Parameters

The Load Parameters command downloads to the MSP-GANG430 the information that the main process needs for gang programming. It must be invoked before the first gang programming command. The parameters are stored in the system flash memory.

Data bytes D1 to D6 specify the parameters, as follows:

D1: Flags for process control, in any combination
- 0x01: Executes erasure of info memory
- 0x02: Executes erasure of main memory
- 0x04: Executes erasure of protected info memory
- 0x07: Executes erasure of main and info memory (including protected info)
- 0x08: Executes programming of info memory
- 0x10: Executes programming of main memory
- 0x18: Executes programming of info and main memory
- 0x20: Executes verification of info and main memory
- 0x40: Makes MSP430 devices secure
- 0x80: Executes programming of RAM memory

D2: Flags for target supply voltage VCC_MSP
- 0x3F: Voltage in 100-mV steps (valid values are 18 to 36, and 0 for external supply)
- 0x80: Switch VCC_MSP off after programming

D3
- 0x01: Executes erase check of info memory
- 0x02: Executes erase check of main memory
- 0x04: Executes erase check of protected info memory
- 0x07: Executes erase check whole memory
- 0x08: Use Spy-Bi-Wire JTAG interface
- 0x10: Use slower JTAG speed mode. Not applicable for Spy-Bi-Wire communication mode.
- 0x40: Verify targets additionally for marginal 1 and 0. For more information, see the *MSP430x2xx Family User's Guide* or *MSP430x4xx Family User's Guide*.
- 0x80: Restore DCO calibration data after erasing of protected INFO A

D4: Reserved, should be 0x00

D5, D6: VCC_MSP settle time

Delay in milliseconds between switching VCC_MSP on and starting programming cycle. Valid values are 1–0xFFFF (>65 s). D5 holds the LSB, and D6 holds the MSB.

Protected information memory segments can be treated in two ways:

- If the calibration data that was stored during factory testing is not needed, this segment can be used as normal information memory segments, and D1 bit 0x04 should be set.
- If calibration data should be preserved, D1 bit 0x04 should be cleared to avoid erasing of the protected information memory segment when normal information memory is erased.

3.5.3 Start

The Start command begins the gang programming cycle, using the operations defined with the Load Parameters command. The result of the command execution can be determined using the Transmit Diagnostic command described in Section 3.5.4.

3.5.4 Transmit Diagnostic

The Transmit Diagnostic command retrieves the result of the preceding gang programming command.

Data bytes D1 to D30 hold the parameters, as follows:

D1 to D5 give information about success of each target channel. The LSB represents channel 1, the MSB represents channel 8.

D1: Channels that are connected to the programmer (device detected)

D2: Channels that passed the erase check successfully

D3: Channels that passed the verification successfully

D4: Channels that passed the device securing successfully, or are already secured

D5: Channels that passed all operations successfully

D6: Miscellaneous errors (see Section 2.1.5.4)

D7 to D8 are reserved.

D9 to D10 hold the hardware version number: D9 (MSByte), D10 (LSByte).

D11 to D12 hold the firmware version number: D11 (MSByte), D12 (LSByte).

D13 to D30 hold a zero-terminated character string representing the firmware file name; e.g., GANG430_100.TXT.

3.5.5 Erase Image

The Erase Image command erases the entire image memory and verifies the erasure.

3.5.6 Load Image Block

The Load Image Block command loads the data bytes into the image buffer of the MSP-GANG430 and verifies them.

D1 to Dn–4 contain the data bytes.

3.5.7 Write Target Selective

The Write Target Selective command loads the data bytes into one exclusively selected target flash memory and verifies them.

Before applying this operation, stable supply voltage at the targets is assumed (if not, use Set Target V_{CC} command and optionally wait for capacitor's settle time).

D1: Target number (0 to 7)

D2: 0

D3 to Dn–4 contain the data bytes (start address and number of bytes must be even numbered).

3.5.8 Read Target Selective

The Read Target Selective command is used for any read access to the flash memory/RAM or peripheral module control registers at 0x0100 to 0x01FE of the selected target.

The 20-bit block start address is defined as:

Address = A3 × 0x10000 + A2 × 0x100 + A1

The 16-bit block length is defined in LL (low byte) and LH (high byte). Because pure data bytes are limited to a maximum of 244, LH is always 0.

For this operation, stable supply voltage at the targets is necessary. If not, use Set Target V_{CC} command and wait for capacitor's settle time, as required.

D1: Target number (0 to 7)

D2: 0

The GANG430 responds with the requested data block. No acknowledge character is necessary.

D1 to Dn contain the data bytes (start address and number of bytes must be even numbered).

3.5.9 Set Target V_{CC}

The Set Target V_{CC} command sets the VCC_MSP pin voltage of the programmer's target connector to the given value.

D1: Target supply voltage

- 0: VCC_MSP output voltage is equal to MSP_VCC_IN input voltage.
- 1: VCC_MSP output voltage is switched off.
- 2: VCC_MSP output voltage is same as loaded by the Load Parameters command.
- 18+: VCC_MSP is set in hundreds of millivolts (e.g., D1 = 36 for 3.6 V).

3.5.10 Load Image Checksum

The Load Image Checksum command writes the checksums of all non-empty memory segments (Info_x, main, RAM) and the memory model of the target device into the system flash memory. The gang programmer uses the checksums to verify the correct download of data to the image buffer and the correct programming of the target device.

On devices with protected information memory segments (e.g., SegA on F2xxx devices), it can be indicated if this segment should be excluded from erasure when the other information memory segments are to be erased.

The number of parameters depends on the memory model of the selected device.

Data bytes D1 to D4 hold the parameters, as follows:

D1:

- 0x80: VPP to TEST
- 0x00: VPP to TDI

D2: Combination of

- 0x01: Device is Spy-Bi-Wire capable
- 0x02: Device has X-CPU

D3: Device ID, corresponds to DevId in GANG430.ini file, LSByte

D4: Device ID, corresponds to DevId in GANG430.ini file, MSByte

D5: Byte 0x0D from ROM device ID, corresponds to DevIdxD

D6: Reserved, should be 0x00

For each memory segment of the device (main memory is treated as one segment), a set of 8 bytes indicating memory type, start address, length, and checksum follows:

D7: Memory type
- 0x06: Calibration data
- 0x05: Protected Info
- 0x03: Main
- 0x02: Info
- 0x01: RAM
- 0x00: Terminates the list of memory segment descriptions

D8
- 0x01: Indicates that the memory segment is empty (= 0xFFFF)
- 0x00: Indicates that the memory segment is not empty

D9: Segment start address, bits 23 to 16
D10: Segment start address, bits 15 to 8
D11: Segment start address, bits 7 to 0
D12: Segment length, bits 23 to 16
D13: Segment length, bits 15 to 8
D14: Segment length, bits 7 to 0
D15: Checksum (PSA) for memory segment, LSByte
D16: Checksum (PSA) for memory segment, MSByte

The list must be terminated by a single byte containing 0x00.

D7 to D14 correspond to the MemDefx=... entries in the GANG430.INI File.

The pseudo signature analysis (PSA) is computed using the following algorithm (using the C language):

```
for (PSA = StartAddr - 2, i = 0; i < Length; i++)
{
  if (PSA & 0x8000)
    PSA = ((PSA ^ 0x0805) << 1) | 1;
  else
    PSA <<= 1;
    PSA ^= Data[i];
}
```

Where StartAddr is the beginning address of the memory region, and length is the length of the memory region data, in words.

3.5.11 Select Baud Rate

The Select Baud Rate command sets the rate of the serial communications. The default is 115200 baud.

Baud rate index 0 to 4, representing the baud rate.

0: 9600 baud
1: 19200 baud
2: 38400 baud
3: 57600 baud
4: 115200 baud

The Select Baud Rate command takes effect (i.e., changes the baud rate) immediately.

3.5.12 Execute Self Test

The Execute Self Test command starts a self test on the MSP-GANG430 hardware. In the event of failure, the MSP-GANG430 Transmit Diagnostic command can be used to obtain detailed information about the failure. D6 (miscellaneous errors) holds the following error codes:

0x04: Self test failed – Control connections to image buffer devices invalid

0x05: Self test failed – No JTAG access to one or more target channels

0x06: Self test failed – Target voltage generator (VCCT) does not work properly

0x07: Self test failed – System voltage (V_{CC}) not in range

0x08: Self test failed – Blow fuse voltage (VPP) not in range

3.5.13 Set Signals

Data bytes D1 to D4 hold the parameters, as follows (also see Section 2.1.6):

D1: Control bits low byte

D2: Control bits high byte

D3: V_{CC} voltage in hundreds of mV (18 to 36 represent 1.8 V to 3.6 V)

D4: 0x00

3.5.14 Select Image

The Select Image command sets a number for the current image. After this command all operations that the GANG430 performs will use this image. The commands which use this image are: Load Parameters, Start, Erase Image, Load Image Block, Load Image Checksum. The GANG430 supports two images. Default image after power on is #0.

D1: holds a number of the image to set (0 or 1).

3.5.15 Firmware Commands

TI recommends against invoking commands that affect the firmware (such as Erase Firmware, Load Firmware Block, and Finalize Firmware), except from the MSP-GANG430 GUI.

Dynamic Link Library GANG430.DLL

This chapter describes the dynamic link library GANG430.DLL.

Topic **Page**

4.1 GANG430.DLL Description **34**
4.2 Return Values/Error Codes From GANG430.DLL **54**

4.1 GANG430.DLL Description

GANG430.DLL communicates with the MSP-GANG430 programmer unit and the connected MSP430 device(s). GANG430.INI must be in the same directory as GANG430.DLL.

The configuration of the MSP-GANG430 should be done with the following sequence.

```
InitCom                 // Open communication port
GangEraseImage          // Erase image flash memory
GangLoadImage           // Write target code into image flash memory
GangLoadParameters      // Determine process flow
ReleaseCom              // Close communication port
```

Once the configuration is completed, all process-specific parameters are stored in the flash memory of the MSP-GANG430 programmer unit.

```
InitCom                 // Open communication port
GangSelectBaudrate      // Optional: slow down communication speed
...GangMainProcess      // Execute gang programming process
...GangGetResult        // Receive result for diagnostic
...GangAccessTargetSFR  // Optional: stimulate device pins
...GangProgramTarget    // Optional: write serial numbers to targets
ReleaseCom              // Close communication port
```

The MSP-GANG430 is able to store two different program codes with two different parameters. This makes it possible to use an MSP-GANG430 both for production test and for final programming of the target systems. The following configuration will download “test code” to the target system, and after this, the “final code” will be downloaded to the target system.

```
InitCom                 // Open communication port
GangSelectImage(0)      // select image #0 (test image)
GangEraseImage          // Erase image flash memory #0
GangLoadImage           // Write target code into image flash memory #0
GangLoadParameters      // Determine process flow #0
GangSelectImage(1)      // select image #1 (final product image)
GangEraseImage          // Erase image flash memory #1
GangLoadImage           // Write target code into image flash memory #1
GangLoadParameters      // Determine process flow #1
...GangSelectImage(0)   // select image #0 (test image)
...GangMainProcess      // Execute gang programming process
...GangGetResult        // Receive result for diagnostic
...GangSelectImage(1)   // select image #1 (final product image)
...GangMainProcess      // Execute gang programming process
...GangGetResult        // Receive result for diagnostic
ReleaseCom              // Close communication port
```

Several examples showing how the DLL could be used are located in the DLL_Usage_Examples subdirectory of the GANG430 system. The return codes for all functions are listed in Section 4.2.

This DLL could be used separately using the following conventions.

InitCom

Declaration

```
long int InitCom(char* lpszComPort, long int lBaudRate)
```

Description

InitCom initializes (opens) the specified communications port, establishes communication with the MSP-GANG430 hardware, and sets the baud rate of the MSP-GANG430. If successful, the MSP-GANG430 is reset.

Arguments

lBaudRate

Valid baud rates are 9600, 19200, 38400, 57600, and 115200 baud. The default baud rate after initialization is 115200 baud.

lpszComPort

The name of the communication port: COM1, COM2, COM3, or COM4

Example

```
lFuncReturn = InitCom("COM1", 115200);
```

ReleaseCom

Declaration

```
long int ReleaseCom (void)
```

Description

This function is the counterpart to InitCom. It allows closing a communication with the MSP-GANG430 hardware.

Example

```
IFuncReturn = ReleaseCom();
```

GangSelectBaudrate

Declaration

```
long int GangSelectBaudrate (long int lBaudrateIndex)
```

Description

Sets the baud rate of the MSP-GANG430 programming unit. It offers the capability of transmissions at lower baud rates than the default 115200 baud.

Arguments

lBaudrateIndex

0: 9600 baud
1: 19200 baud
2: 38400 baud
3: 57600 baud
4: 115200 baud (default after power up)

Example

```
lFuncReturn = GangSelectBaudrate(2);   // selects 38400 baud
```

GangEraseImage

Declaration `long int GangEraseImage(void)`

Description Clears (presets with 0xFF) the image memory of the MSP-GANG430 programming unit. Performs a succeeding erase check over the address range 0x1000–0xFFFFF.

Example

```
lFuncReturn = GangEraseImage();
```

GangLoadImage

Declaration

```
long int GangLoadImage(char* lpszFileName, char* lpszDeviceName)
```

Description

This function writes data from a file either of type TI-txt or of type Intel-hex (auto-detect) into the MSP-GANG430 programming unit's image flash memory. It also writes the memory model parameters of the selected device type and the checksums for used memory segments into the programmer's flash system memory.

Arguments

lpszFileName

Name of the file to be loaded (full path) or NULL if function is only called to set up the target device.

lpszDeviceName

Name of the device in file Gang430.ini. Take care of the <space> between MSP430 and Fxxx.

Example

```
lFuncReturn = GangLoadImage(FileName, "MSP430 F1121A");
lFuncReturn = GangLoadImage(NULL, "MSP430 F1121A");
```

GangSelectImage

Declaration `long int GangSelectImage(long lImage);`

Description This function sets the active image. GANG430 supports two images.

Arguments lImage

0 or 1, the image number. Default image number is 0.

Example `lFuncReturn = GangSelectImage(1);`

GangLoadParameters

Declaration

```
long int GangLoadParameters(long int lFlags, long int lSupply, long int
lVccSettleTime)
```

Description

This function downloads the information that the main process needs for gang programming to the MSP-GANG430. It must be invoked prior to the first GangMainProcess call. The parameters are stored in the system flash memory.

Arguments

lFlags

F_ERASE_INFO	0x0001	// executes erasure of info memory
F_ERASE_MAIN	0x0002	// executes erasure of main memory
F_ERASE_PINFO	0x0004	// executes erasure of protected info memory
F_ERASE_MASS	0x0007	// executes erasure of info and main memory (including protected info)
F_PROGRAM_INFO	0x0008	// executes programming of info memory
F_PROGRAM_MAIN	0x0010	// executes programming of main memory
F_PROGRAM_MASS	0x0018	// executes programming of info and main memory
F_VERIFY	0x0020	// executes verification of info and main memory
F_SECURE_DEVICE	0x0040	// makes MSP430 devices secure
F_PROGRAM_RAM	0x0080	// executes programming of ram memory
F_ERASE_CHECK_INFO	0x0100	// executes erase check of info memory
F_ERASE_CHECK_MAIN	0x0200	// executes erase check of main memory
F_ERASE_CHECK_PINFO	0x0400	// executes erase check of protected info memory
F_ERASE_CHECK_MASS	0x0700	// executes erase check whole memory
F_USE_SBW	0x0800	// use SpyBiWire interface (2-wire JTAG)
F_JTAG_SPEED1	0x0000	// fastest JTAG speed mode
F_JTAG_SPEED2	0x1000	// slower JTAG speed mode
F_VERIFY_MARGINAL	0x4000	// verify target additionally for marginal 1 and 0
F_RESTORE_DCO	0x8000	// restore DCO calibration data after erase of INFO A

Note 1: F_ERASE_PINFO can be applied if F_ERASE_INFO is set.

Note 2: F_ERASE_CHECK_PINFO can be applied if F_ERASE_CHECK_INFO is set.

Note 3: F_VERIFY_MARGINAL can be applied only if the F_VERIFY is set. For more information, see the *MSP430x2xx Family User's Guide* or *MSP430x4xx Family User's Guide*.

Note 4: F_RESTORE_DCO can be applied if F_ERASE_PINFO or F_ERASE_CHECK_PINFO is set.

lSupply

0x7F: Supply voltage VCC_MSP in hundreds of millivolts (for example, 30 = 3 V).

0x80: Switch VCC_MSP off after main process execution (symbolic F_VCC_OFF = 0x80).

lVccSettleTime

Capacitor settle time in milliseconds; maximum is 0xFFFF ≈ 65 s.

Example

```
lFuncReturn = GangLoadParameters(0x3F, 30, 100);
// executes Mass Erase, Erase Check, Mass Program, Verification;
// VCC_MSP = 3 V, VCC_MSP settle time is 100 ms.
```

GangMainProcess

Declaration

```
long int GangMainProcess(long int lTimeout)
```

Description

This function executes the main process determined by the parameters loaded through the Gang- LoadParameters function. Use a succeeding GangGetResult call to receive the result of the executed main process.

Arguments

lTimeout

The time out in hundreds of milliseconds until the programming unit must respond.

Example

```
lFuncReturn = GangMainProcess(120); // with 12s time out
```

GangGetResult

Declaration

```
long int GangGetResult(void *lpData)
```

Description

This function reads out the result of the precedent execution of the main process at the MSP-GANG430 programming unit.

It also can be used to get detailed error information about the precedent execution of some other functions like GangSelftest.

Arguments

lpData

Pointer points to a buffer that receives the result data.

Data bytes D00 to D29 hold the parameters, as follows:

D00 to D04 give information about success of each target channel. The LSB represents channel 1 the MSB represents channel 8.

D00: Channels that are connected to the programmer (device detected).

D01: Channels that passed the erase check successfully.

D02: Channels that passed the verification successfully.

D03: Channels that passed the device securing or the device was already secured.

D04: Channels that passed all operations successfully.

D05: Miscellaneous errors (see Section 2.1.5.4)

D06 to D07 are reserved.

D08 to D09 hold the hardware version number.

D08 = MSByte

D09 = LSByte

D10 to D11 hold the firmware version number.

D10 = MSByte

D11 = LSByte

D12 to D29 hold a zero-terminated character string representing the firmware file name (e.g., GANG430-120.TXT).

Example

```
lFuncReturn = GangGetResult(lpBuffer);
```

GangAccessTargetSFR

Declaration

```
long int GangAccessTargetSFR(long int lTarget, long int iFlags, long int
lAddress, void* lpData)
```

Description

Accesses one byte or word within the special function register (SFR) area of one selectable target. The user must take care about the memory attributes. Read access also can be used over the entire address range.

The first (last) access must additionally be set to the SFR_START (SFR_STOP) flag.

Before applying this operation, stable supply voltage at the targets is assumed (if not, use GangSetVccTarget operation and optionally wait for capacitor's settle time).

Arguments

lTarget

The target number minus 1 (indicating target 1 to 8 at programmer unit)

lFlags

SFR_READ	0x00	// read access
SFR_WRITE	0x01	// write access, all targets are affected
SFR_START	0x02	// start sequence (stop watchdog, halt CPU)
SFR_STOP	0x04	// stop sequence (release CPU)

lAddress

Address of memory to be accessed. If address is lower than 0x0100, byte access is performed; otherwise word access is performed.

lpData

Pointer points to a buffer that holds the data to be read/written.

Example

```
lFuncReturn = GangAccessTargetSFR(7, 0, 0x120, &lpData);
// reads WDTCTL register of target 8 wordwise to lpData
```

GangProgramTarget

Declaration

```
long int GangProgramTarget(long int lTarget, long int lStart, long int
lLength, void *lpData)
```

Description

Programs data exclusively into the flash memory of one selectable target. It is mainly used for applying serial numbers.

Before applying this operation, stable supply voltage at the targets is assumed (if not, use GangSetVccTarget operation and optionally wait for capacitor's settle time).

Arguments

lTarget

The target number minus 1 (indicating target 1 to 8 at programmer unit)

lStart

Start address of destination memory (must be even)

lLength

Length in number of bytes of destination memory (must be even)

lpData

Pointer points to a buffer that holds the data to be programmed (the source data buffer).

Example

```
lFuncReturn = GangProgramTarget(7, 0x1000, 0x10, lpData);
// programs 16 bytes exclusively into target 8 starting at address 0x1000
```

GangReadTarget

Declaration

```
long int GangReadTarget(long int lTarget, long int lStart, long int
lLength, void *lpData);
```

Description

Reads out data exclusively from one selectable target device into a buffer.

Before applying this operation stable supply voltage at the targets is assumed (if not, use GangSetVccTarget operation and optionally wait for capacitor's settle time).

Arguments

lTarget

The target number minus 1 (indicating target 1 to 8 at programmer unit)

lStart

Start address of the area to be read out (must be even). Allowed values are 0x0100–0x1FFFE (see memory map of the corresponding device).

lLength

Length of the area (must be even). Allowed values are 0x0000–0xFFFFE (see memory map of the corresponding device).

lpData

Pointer points to a buffer the data to be written to (the destination data buffer).

Example

```
lFuncReturn = GangReadTarget(0, 0xF000, 0x1000, lpDest);
// reads 4k bytes exclusively from target 1 starting at address 0xF000
into a buffer
```

GangReadTargetFile

Declaration

```
long int GangReadTargetFile(long int lTarget, long int lStart, long int
lLength, char* lpszFileName,
long int iFileType);
```

Description

Reads out data exclusively from one selectable target device into a file either of type TI-txt or Intel-hex.

Before applying this operation stable supply voltage at the targets is assumed (if not, use GangSetVccTarget operation and optionally wait for capacitor's settle time).

Arguments

lTarget

The target number minus 1 (indicating target 1 to 8 at programmer unit)

lStart

Start address of the area to be read out (must be even). Allowed values are 0x0100–0xFFFFE (see memory map of the corresponding device).

lLength

Length of the area (must be even). Allowed values are 0x0000–0xFFFFE (see memory map of the corresponding device).

lpszFileName

Name of the file (full path) to receive data. If the file does not exist, it will be created; if the file already exists, it will be overwritten.

iFileType

FILETYPE_TI_TXT	0x01	file type is TI-txt
FILETYPE_INTEL_HEX	0x02	file type is Intel-hex

Example

```
lFuncReturn = GangReadTargetFile(7, 0xF000, 0x1000, FileName, 1);
// reads 4k bytes exclusively from target 8 starting at address 0xF000
into a file with TI-txt format
```

GangSetVccTarget

Declaration

```
long int GangSetVccTarget(long int lVoltage)
```

Description

This function sets the VCC_MSP pin voltage of the programming adapter's target connector to the given value.

Arguments

lVoltage

VCC_EXT	0	VCC_MSP output voltage is equal to MSP_VCC_IN input voltage
VCC_OFF	1	VCC_MSP output voltage is switched off
VCC_NOM	2	VCC_MSP output voltage is same as loaded by the GangLoadParameters function.
Others	18+	VCC_MSP is set in hundreds of millivolts

Example

```
lFuncReturn = GangSetVccTarget(36);  // Set target VCC to 3.6 V
```

GangSelftest

Declaration

```
long int GangSelftest(long int lTimeout)
```

Description

GangSelftest performs a self test at the MSP-GANG430 programmer unit. Follow with a GangGetResult call to retrieve the results of the self test.

For information about possible errors, see Section 3.5.12.

Arguments

lTimeout

The time out in hundreds of milliseconds until the programming unit must respond.

Example

```
lFuncReturn = GangSelftest(60);  // with 6s time out
```

GangSetSignals

Declaration

```
long int GangSetSignals(long int lSignals, long int lVoltage)
```

Description

Sets specified signals at the MSP-GANG430 programmer unit's target connector.

For information about bit definitions, see Section 2.1.6.

Arguments

lSignals
: Determines output signals

lVoltage
: Determines logic high-level voltage of signals in hundreds of millivolts

Example

```
lFuncReturn = GangSetSignals(0x0400, 27);
// VPP at pin 17 (TEST/VPP); VCC = 2.7 V
```

InitProgress

Declaration

```
long int InitProgress(long int hStatusWnd, long int hProgBar)
```

Description

Passes the handles of the status window and the progress bar of the front-end application. It offers the capability of updating a status window and/or a progress bar during file I/O operations, for example.

Arguments

hStatusWnd
: Window handle of a status line

hProgBar
: Window handle, especially of a progress bar

Example

```
lFuncReturn = InitProgress((long int)lpStatus->GetSafeHwnd(), (long
int)lpProgress->GetSafeHwnd());
```

GetErrorString

Declaration

```
char* GetErrorString(long int lErrorNumber)
```

Description

Determines the string associated with the error number. For invalid error numbers, a pointer to "Invalid error number!" is returned.

Arguments

lErrorNumber

The error number

Example

```
lpszErrorString = GetErrorString(lFuncReturn);
```

4.2 Return Values/Error Codes From GANG430.DLL

Table 4-1. Return Values/Error Codes

Status	Return Value	Comment
ERR_NONE	0	Operation successful
ERR_COMM	1	Communication – Frame has errors
ERR_OPEN_COMM	2	Unable to open COM port – already in use?
ERR_CLOSE_COMM	3	Unable to close COM port
ERR_SET_COMM_STATE	4	Unable to modify COM port state
ERR_SYNC	5	Synchronization failed. Programmer connected?
ERR_RX_HDR_TIMEOUT	6	Timeout during operation – Correct COM port selected?
ERR_CMD_NOT_COMPLETED	7	Command did not complete correctly
ERR_CMD_FAILED	8	Command failed or not defined or Target not accessible
ERR_WRONG_BAUDRATE	9	Wrong baud rate specified
ERR_READ_INI	10	Could not read GANG430.ini
ERR_BAD_RECORD	11	File contains invalid record
ERR_FILE_END	12	Unexpected end of file
ERR_FILE_IO	13	Error during file I/O
ERR_FILE_DETECT	14	Selected file is of unrecognizable format
ERR_FILE_OPEN	15	Unable to open file
ERR_ARGUMENT	16	Function argument(s) out of range
ERR_IMAGE_CORRUPTED	31	Image Memory corrupted or erased
ERR_IMAGE_JTAGPORT	32	Self test – No JTAG access to Image Buffer device
ERR_IMAGE_DATAPORT	33	Self test – Data connections to Image Buffer device invalid
ERR_IMAGE_HANDSHAKE	34	Self test – No access to one ore more Image Buffer devices
ERR_TARGET_JTAGPORT	35	Self test – No JTAG access to one or more Target channels
ERR_TARGET_VOLTAGE	36	Self test – Target Voltage Generator (VCCT) does not work properly
ERR_SYSTEM_VOLTAGE	37	Self test – System Voltage (VCC) not in range
ERR_BLOWFUSE_VOLTAGE	38	Self test – Blow Fuse Voltage (VPP) not in range
ERR_TARGET_NOACCESS	39	Target not accessible
ERR_VERIFY_FAILED	40	Verification failed
ERR_NO_PARMS	41	Main Process Parameters not yet set
ERR_IMAGE_ERASE	42	Could not erase Image Buffer
ERR_IMAGE_LOAD	43	Could not load Image Buffer
ERR_PARMS_LOAD	44	Could not load Main Process Parameters
ERR_SEL_BAUDRATE	45	Could not select Baud Rate
ERR_SET_VCC	46	Could not set target voltage (VCCT) – Short circuit or settling time too short
ERR_WRONG_CMD	47	Invalid firmware command
ERR_LOW_SUPPLY_VOLTAGE	48	Power supply voltage too low
WNG_LOW_VCC_IN	49	Sense voltage out of range – Check pin MSP_VCC_IN
ERR_WRONG_DEVICE	50	Wrong target device connected (target doesn't match selected device)
ERR_DEVICE_MEM_MISMATCH	51	Image doesn't fit memory model of selected device.
ERR_NO_DEVICE	52	No target device connected

Hardware

This chapter presents the MSP-GANG430 specifications and describes the required interconnections between the MSP-GANG430 and the target devices.

Topic **Page**

5.1	Specifications	56
5.2	Programming Times vs Code Size for the Gang Programmer	56
5.3	Recommendations for Target Connections	57
5.4	MSP-GANG430 Target Connector Signals	57
5.5	MSP-GANG430 Schematics	61
5.6	MSP-GANG430 Component Locations	62
5.7	Gang_Exp Target Expansion Board Layout	63

5.1 Specifications

The specifications for the MSP-GANG430 hardware are shown in Table 5-1.

Table 5-1. MSP-GANG430 Hardware Specifications

Power supply	9 V to 15 V dc, 300 mA minimum
Output voltage (V_{CC})	1.8 V to 3.6 V ± 100 mV dc, 150 mA maximum
Output voltage (VPP)	6.5 V ± 400 mV dc, 150 mA maximum
Signal path length between MSP-GANG430 and each target. If cables between programmer and targets are longer then 20 cm, serial resistors of 100 Ω should be inserted on signals TCK and TMS on the expansion board. TCK is on pin 4, TMS on pin 3 of the 25-pin Sub-D connector.	<60 cm
Temperature range	10°C to 45°C (50°F to 113°F)
Humidity	40% to 70%
Dimensions	150 mm (W) × 30 mm (H) × 82 mm (D)

5.2 Programming Times vs Code Size for the Gang Programmer

T_{total} [ms] ≈ 210 + T_S + E × 220 + (EC + V) × 7.6 × $size_{mem}$ + P × 175 × $size_{code}$

Where

E = 1 if Erase checkbox is selected, 0 otherwise

EC = 1 if Erase Check checkbox is selected, 0 otherwise

P = 1 if Program checkbox is selected, 0 otherwise

V = 1 if Verify checkbox is selected, 0 otherwise

T_S: VCCT settle time in ms, defined in [User] section of GANG430.ini (e.g., VCCTSettleTime = 100)

$size_{mem}$: Flash memory size in KBytes of the selected device

$size_{code}$: Code size in KBytes

The following example is for MSP430F149, with all functions selected, T_S = 100 ms, and full memory programmed:

T_{total} [ms] = 210 + 100 + 220 + 2 × 7.6 × 60 + 175 × 60 = 11942 = 12 sec

The following equation is a simplified formula for all functions selected, T_S = 100 ms, and nearly the full memory of device programmed:

T_{total} [ms] ≈ 530 + 190 × $size_{code}$

Note: Using SpyBiWire mode requires approximately 3 seconds to program 2 KByte. As devices are programmed consecutively, the total time for eight devices is 24 seconds.

5.3 Recommendations for Target Connections

The following hardware connections are recommended when connecting the target MSP430 flash devices to the MSP-GANG430 without usage of the expansion board.

- The VCC pins of all the targets must be tied together and connected to the positive terminal of the supply.
- The VSS pins of all targets must be tied together and connected to the GND or negative terminal of the supply.
- For targets without Test/VPP, five interconnections are needed: TMS, TCK, TDI/VPP, TDO/TDI, and VSS.
- For targets with Test/VPP, six interconnections are needed: TMS, TCK, TDI, TDO/TDI, VSS, and TEST/VPP.
- Use short cables to connect the target to the MSP-GANG430–shorter than 60 cm is recommended. If cables between programmer and targets are longer then 20 cm, serial resistors of 100 Ω should be inserted on signals TCK and TMS on the expansion board. TCK is on pin 4 and TMS is on pin 3 of the 25-pin Sub-D connector. Also, a separate low impedance GND connection should be added between targets and GND terminal on the expansion board.
- Ensure low-impedance interconnections, especially for the path of the JTAG fuse blow voltage (TDI/VPP or TEST/VPP).

Note: If targets that are already powered are connected to the MSP-GANG430, an accidental fuse check can occur while connecting. This can falsely indicate a blown fuse. To avoid this accidental fuse check, the TMS signal (pin 5 on the 14-pin JTAG connector) should be connected to VCC_MSP (pin 2 on the 14-pin JTAG connector) via a 3.3-kΩ pullup resistor on the programmer side. VCCO should not be connected to the supply voltage of the target, as the target is powered externally. From the GUI or the DLL, VCC_MSP should be selected to be permanently on to drive TMS to 1 while the target is being connected to the programmer. Using the GUI, this can be achieved by unchecking the box Switch Vcc Off.

5.4 MSP-GANG430 Target Connector Signals

Figure 5-1, Figure 5-2, and Table 5-2 show the target connector signals for the MSP-GANG430. Chapter 6 is a schematic of the connections necessary to program multiple target devices.

CAUTION

When an external supply voltage is used for the target systems, it is important to disconnect VCC_MSP from the targets to avoid power supply conflicts that could potential damage the MSP-GANG430 and the targets. Connect the external supply voltage to MSP_VCC_IN.

It is not possible to use the MSP-GANG430 with targets powered by the MSP-GANG430 and with targets powered by an external supply simultaneously.

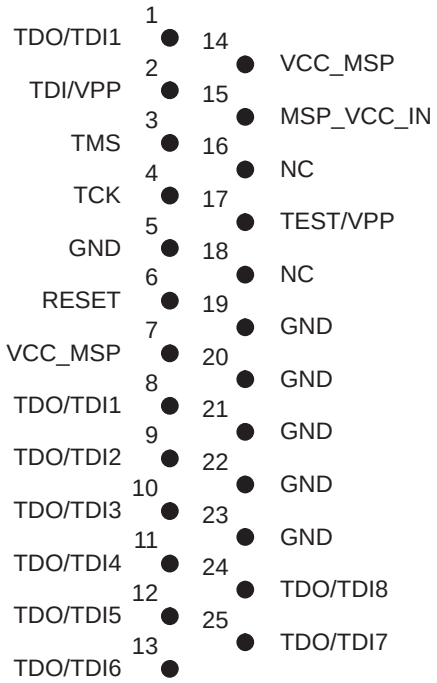


Figure 5-1. 25-Pin Sub-D at MSP-GANG430

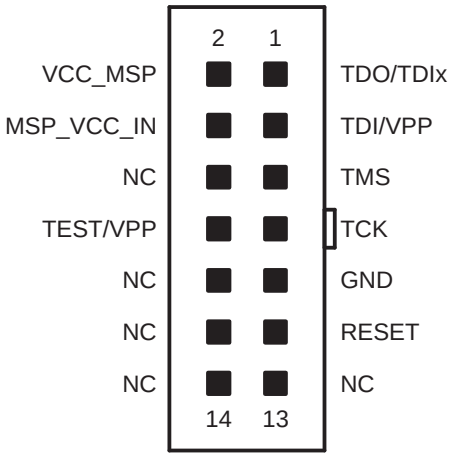


Figure 5-2. 14-Pin Connector at End of Interconnect Cable

Table 5-2. MSP-GANG430 Target Connector Signal Functions

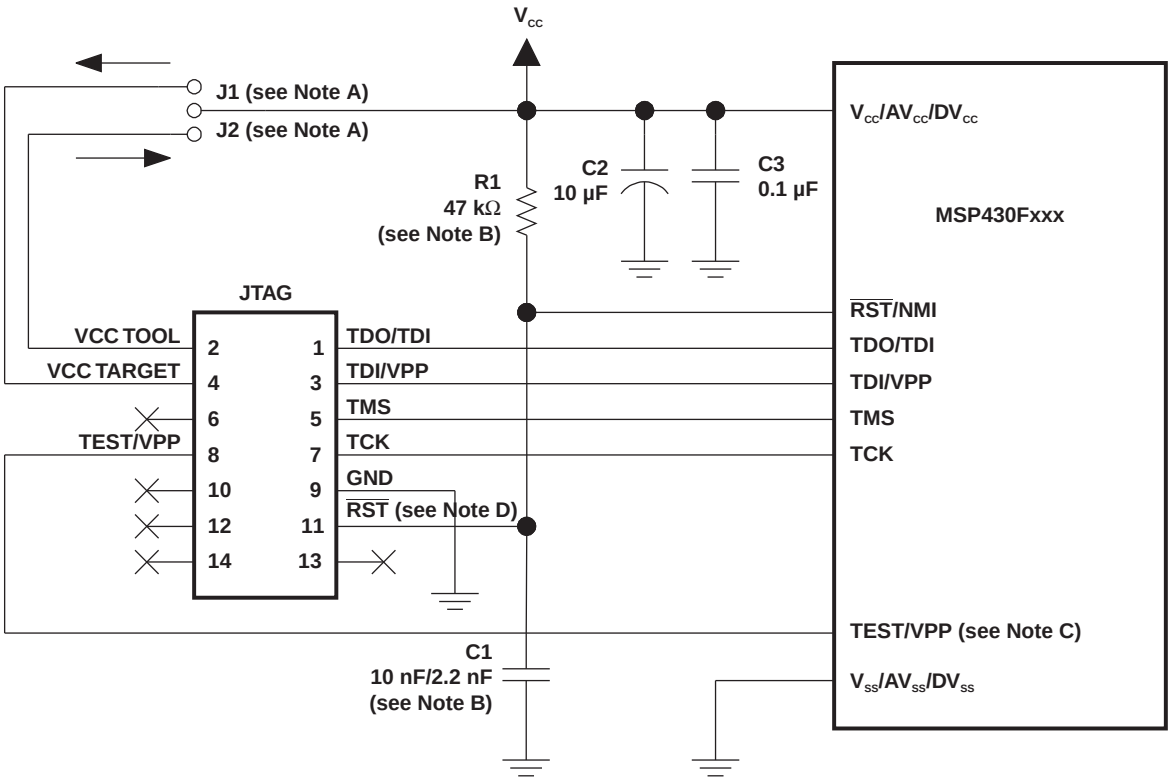
Signal Name	Required	Function/Comment
TMS	Yes	IEEE Std 1149.1 test mode select input
TCK	Yes	IEEE Std 1149.1 test clock input
TDI/VPP	Yes	IEEE Std 1149.1 test data input multiplexed with fuse blow voltage input
TDO/TDI1 to TDO/TDI8	Yes	IEEE Std 1149.1 test data output multiplexed with test data input for use during fuse-blow operation
GND	Yes	GND is the 0-V terminal
VCC_MSP	Yes (if internal supply voltage is used)	Software selectable supply voltage V_{CC} to power the targets from the programmer
MSP_VCC_IN	Yes (if internal supply voltage is used)	External supply voltage sense input, to adapt the internal level-shifter outputs
TEST/VPP	Yes (depending on device)	Signal used with MSP430 flash devices with TEST/VPP pin
RESET	Yes (depending on end application)	Provides the reset of target device. The duration and delay after reset is set by Settling Time.

The output signal levels of the MSP-GANG430 are near GND or VCC_MSP.

- The programming procedure (handling of the software) is described in Chapter 1 and Chapter 2.
- The connections from the MSP430 terminals must follow EMI rules, including short trace lengths and use of ground planes.

Table 5-3. MSP-GANG430 Signal Levels

Signal/Pin	Signal/Pin Levels
TMS	V_{SS} or VCC_MSP
TCK	V_{SS} or VCC_MSP
TDI/VPP	V_{SS} or VCC_MSP or VPP
TDO/TDI1 to TDO/TDI8	V_{SS} or VCC_MSP
GND	V_{SS}
VCC_MSP	V_{SS} or VCC_MSP
MSP_VCC_IN	Input: External V_{CC} of MSP430
TEST/VPP	V_{SS} or VCC_MSP or VPP
RESET	V_{SS} or Hi-Z (external pullup resistor required)



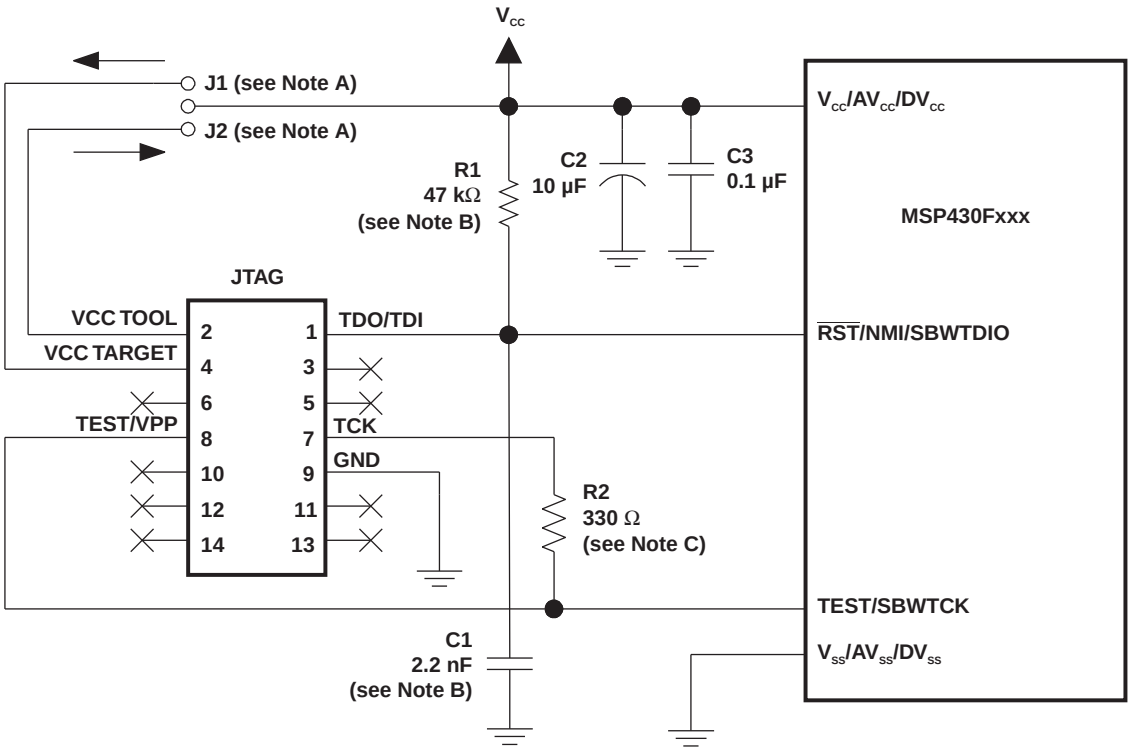
A Make either connection J1 (if a local target power supply is used) or connection J2 (if the target is powered from the debug/programming adapter).

B The $\overline{\text{RST}}$/NMI pin R1/C1 configuration is device-family dependent. See the respective MSP430 family user's guide for the recommended configuration. When using 2-wire JTAG communication on capable devices in 4-wire JTAG mode, the upper limit for C1 should not exceed 2.2 nF. This applies to both TI FET interface modules (LPT/USB FET).

C The TEST/VPP pin is available only on MSP430 family members with multiplexed JTAG pins. See the device data sheet to determine pin availability.

D The connection to the JTAG connector $\overline{\text{RST}}$ pin is optional when using devices that are capable of only 4-wire JTAG communication mode, and it is not required for device programming or debugging. However, this connection is required when using devices that are capable of 2-wire JTAG communication mode in 4-wire JTAG mode.

Figure 5-3. Signal Connections for 4-Wire JTAG Communication



A Make either connection J1 (if a local target power supply is used) or connection J2 (if the target is powered from the debug/programming adapter).

B The device $\overline{\text{RST}}$/NMI/SBWTDIO pin is used in 2-wire mode for bidirectional communication with the device during JTAG access, and any capacitance attached to this signal may affect the ability to establish a connection with the device. The upper limit for C1 is 2.2 nF when using current TI FET interface modules (USB FET).

C R2 protects the JTAG debug interface TCK signal against the JTAG security fuse blow voltage that is supplied by the TEST/VPP pin during the fuse blow process (securing device). If fuse blow functionality is not needed, R2 is not required (leave open), and the connection TEST/VPP must not be made.

Figure 5-4. Signal Connections for 2-Wire JTAG Communication (SpyBiWire)

5.5 MSP-GANG430 Schematics

The MSP-GANG430 schematics are presented in Chapter 6.

Note: Circuit U3 (TLC5921, LED-Driver) is operated at V_{CC} = 3.6 V, which is out of specification ($V_{CC(min)}$ = 4.5 V). This is acceptable here due to limited operating conditions.

5.6 MSP-GANG430 Component Locations

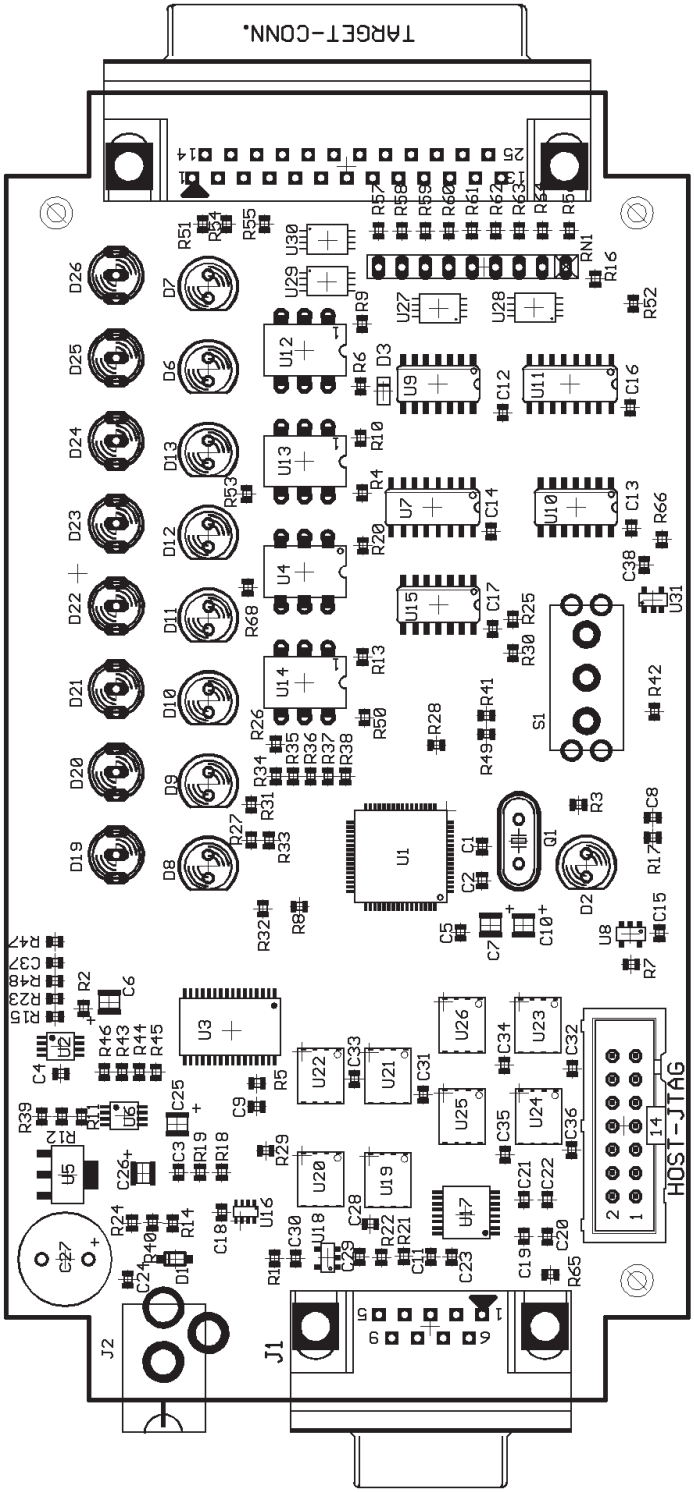


Figure 5-5. MSP-GANG430 Component Locations

5.7 Gang_Exp Target Expansion Board Layout

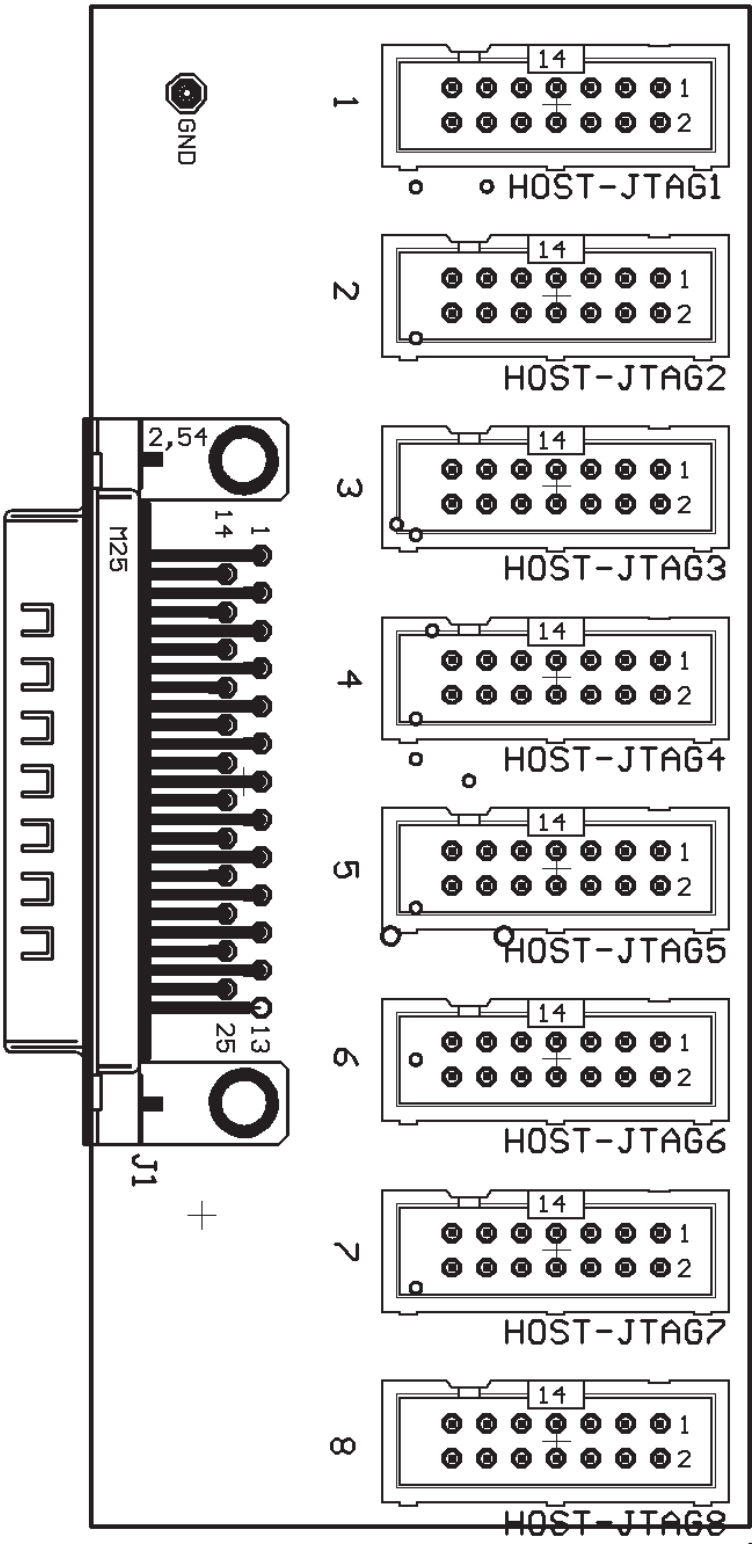


Figure 5-6. Gang_Exp Layout

Schematics

This chapter shows the schematics of the MSP-GANG430 and the connections necessary to program multiple target devices.

Topic **Page**

6.1 Schematics **66**
6.2 Revision History **72**

6.1 Schematics

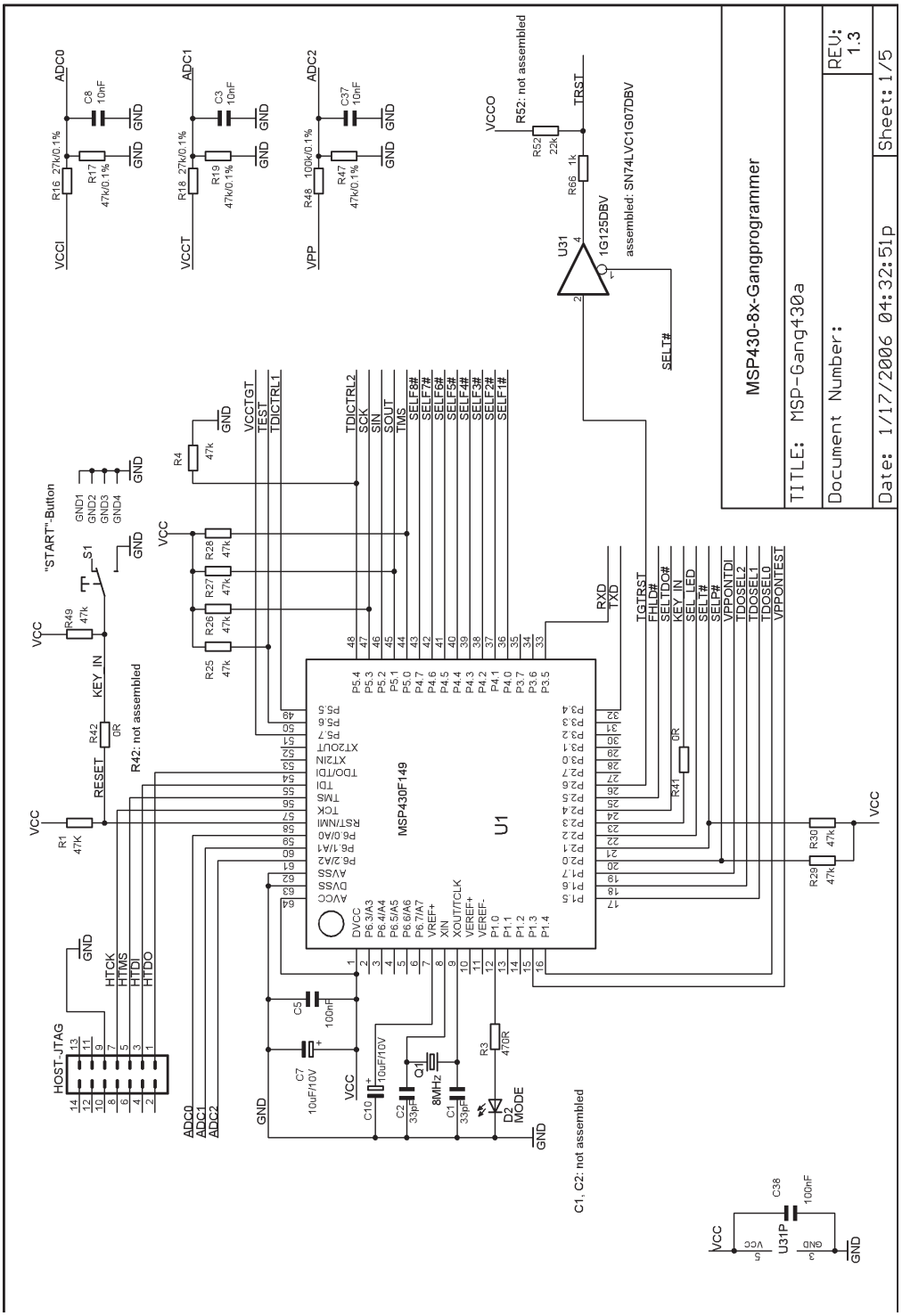


Figure 6-1. MSP-GANG430 Schematic (1 of 5)

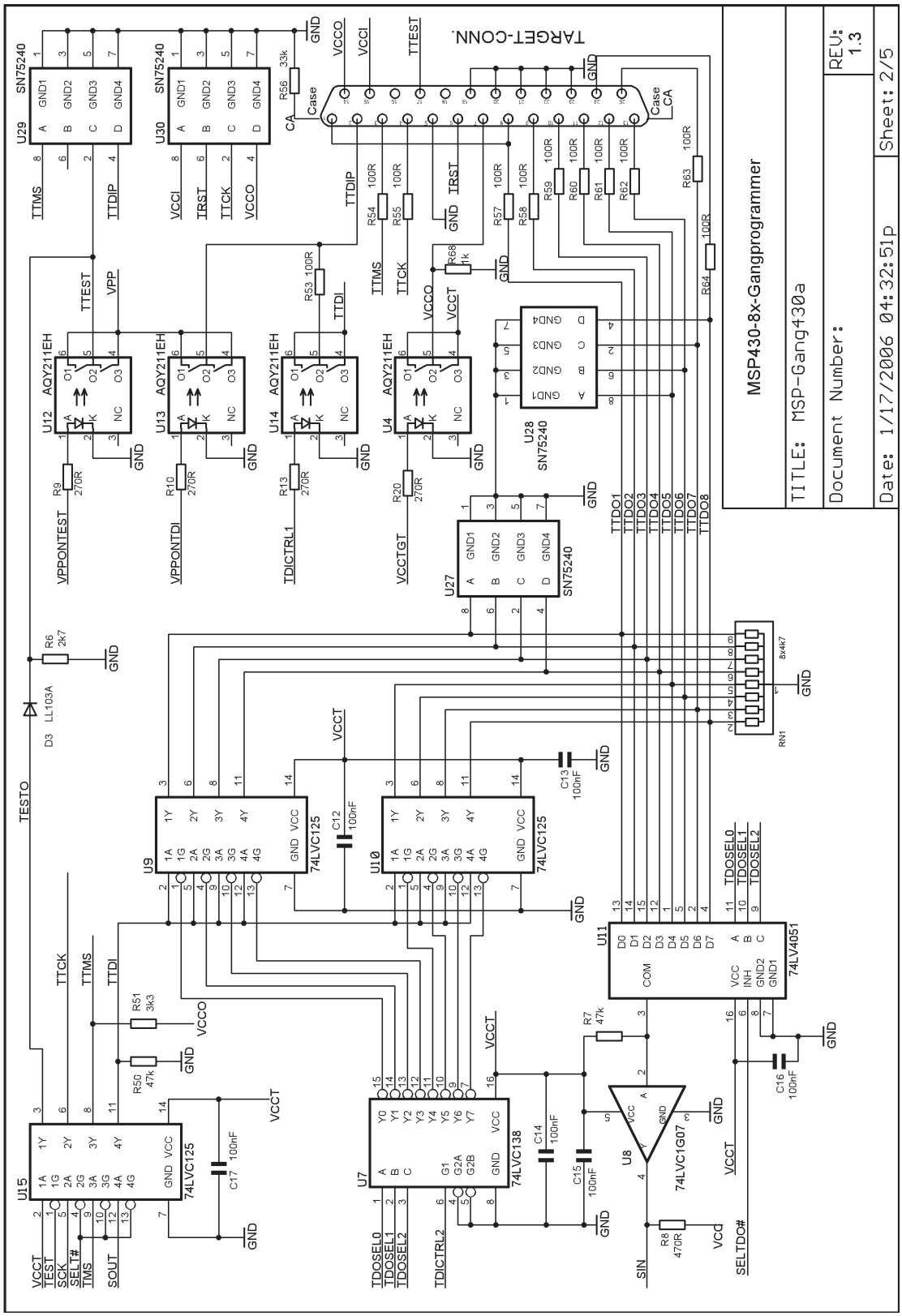


Figure 6-2. MSP-GANG430 Schematic (2 of 5)

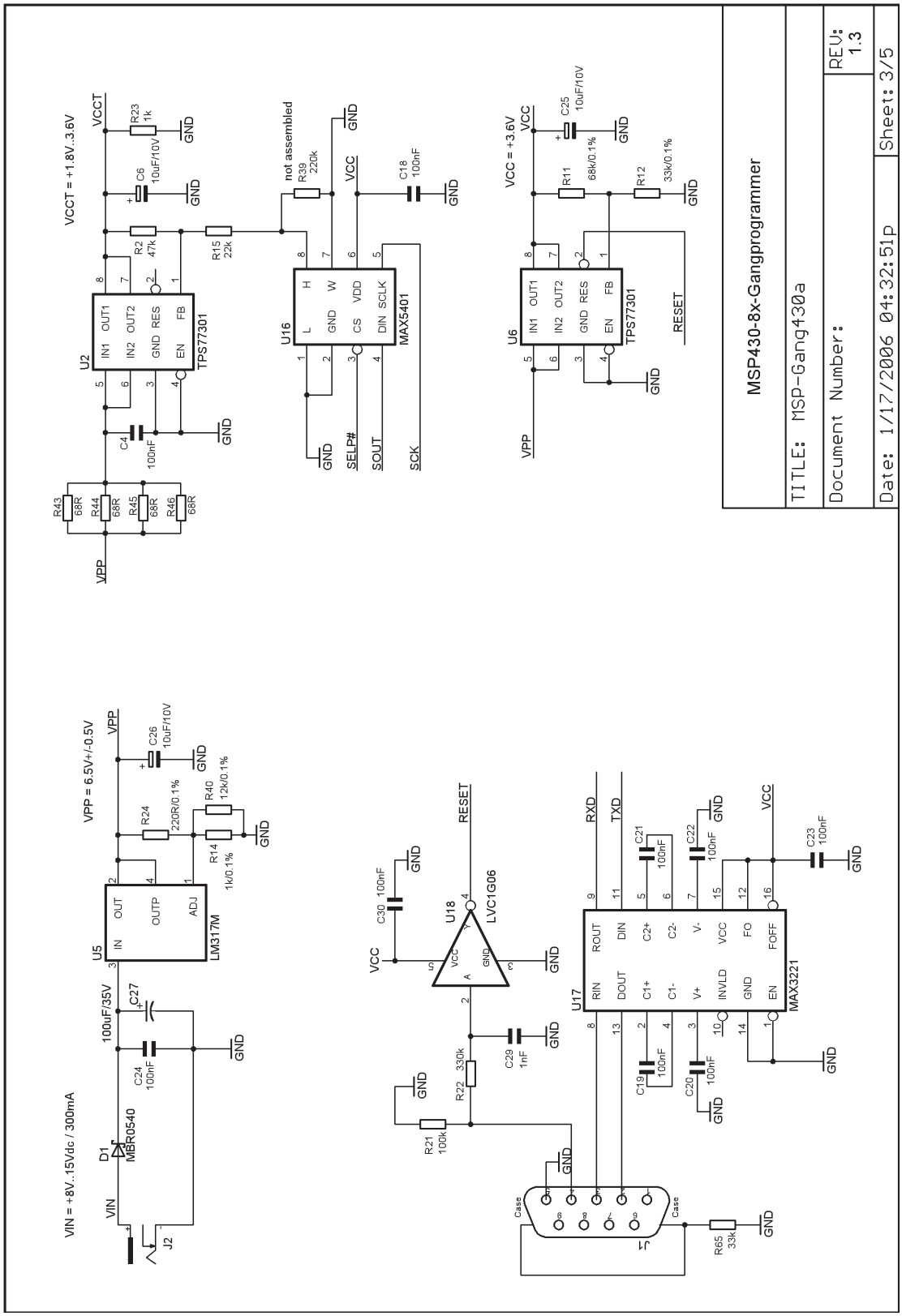


Figure 6-3. MSP-GANG430 Schematic (3 of 5)

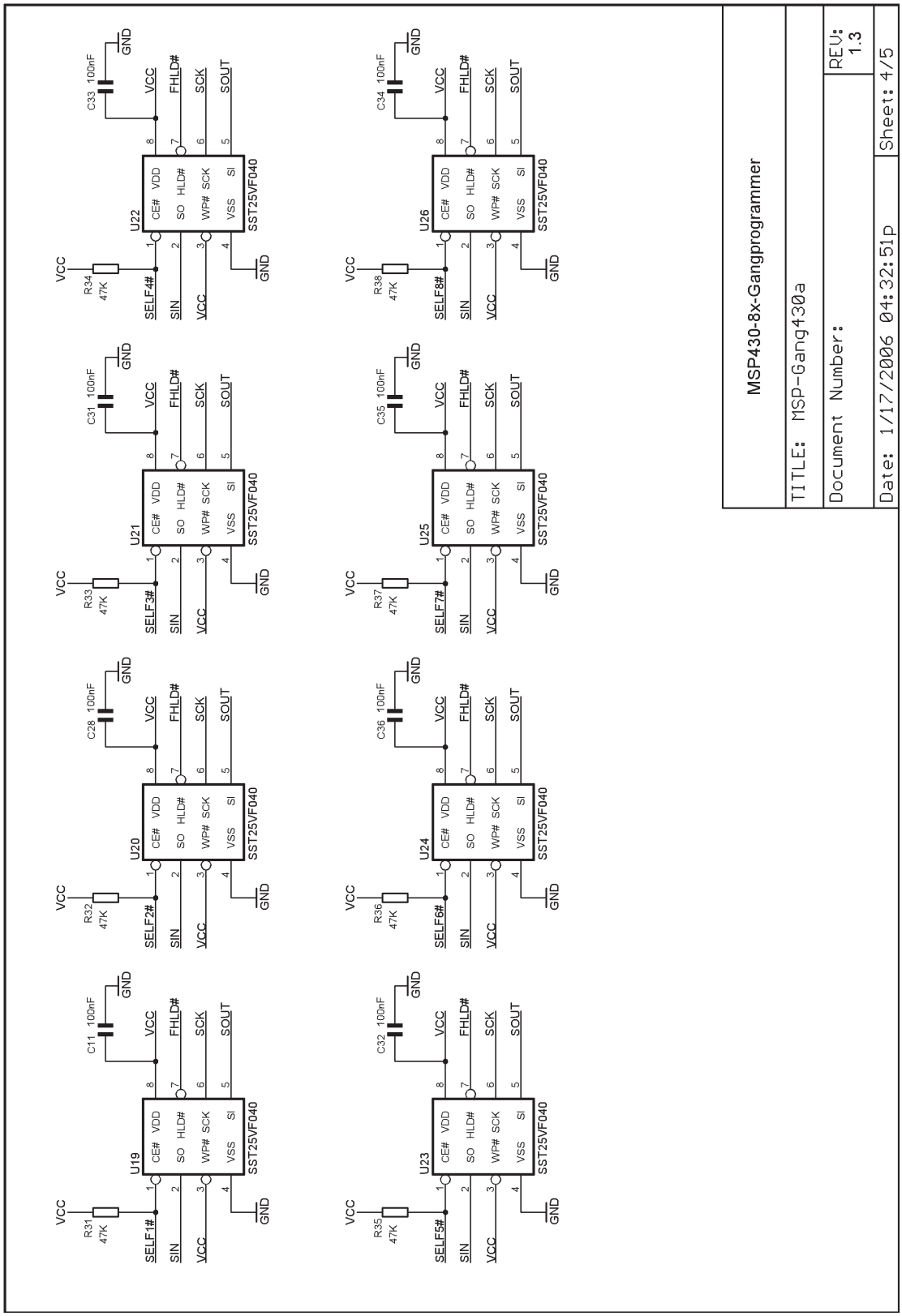


Figure 6-4. MSP-GANG430 Schematic (4 of 5)

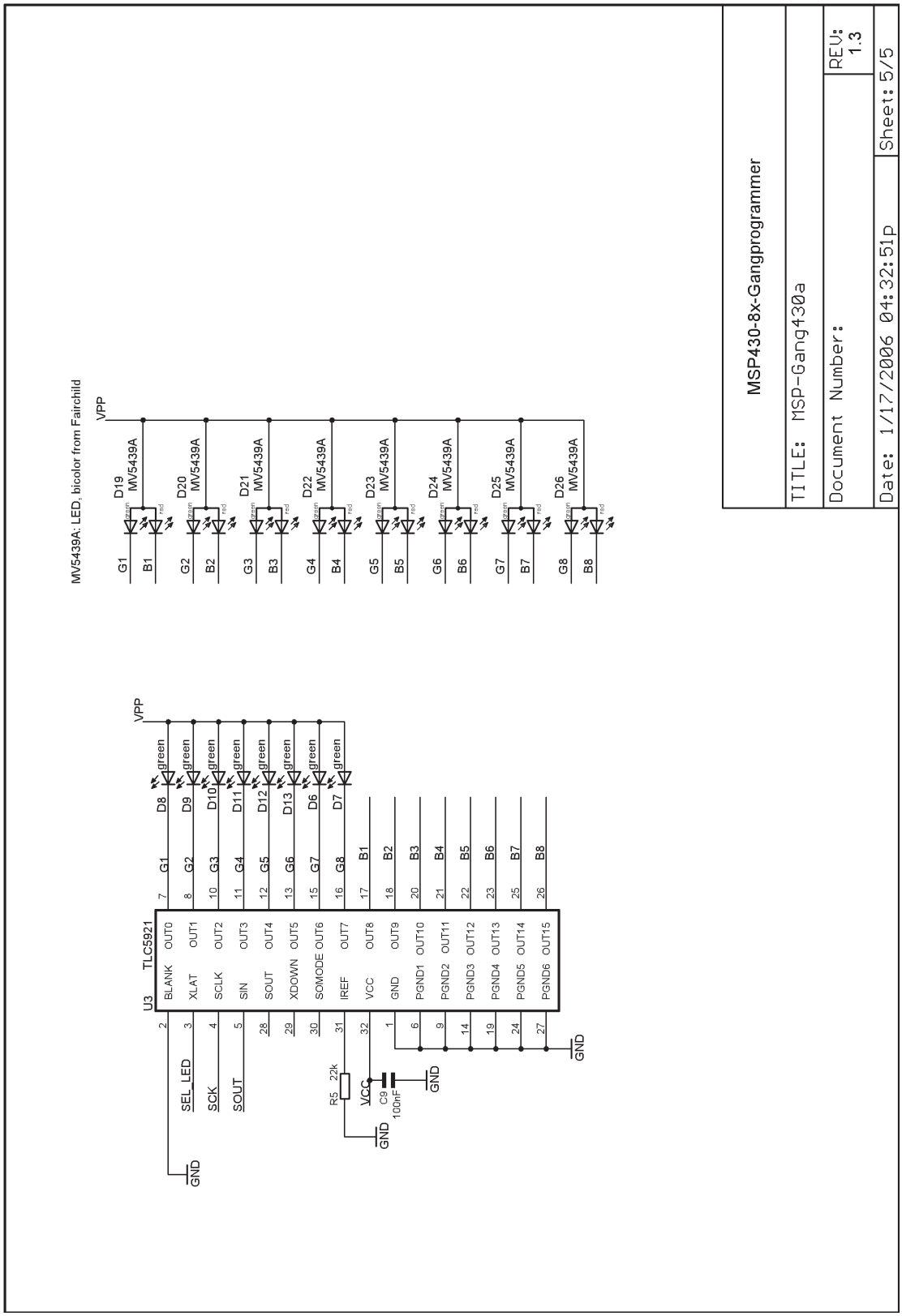


Figure 6-5. MSP-GANG430 Schematic (5 of 5)

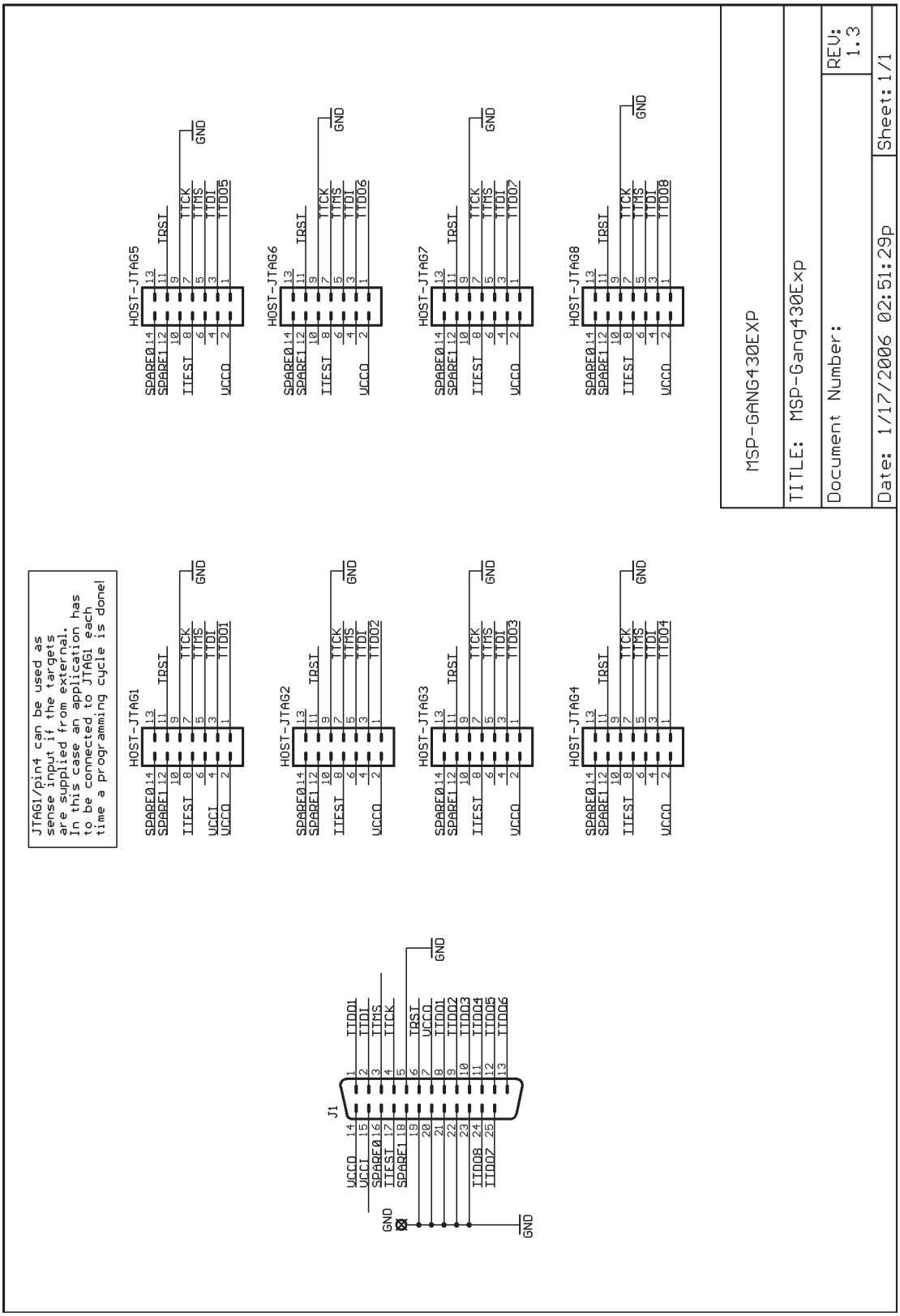


Figure 6-6. MSP-GANG430 Expansion Board Schematic

6.2 Revision History

Revision 1.2

- Initial released hardware version
- Only the 2-wire JTAG mode is supported on SpyBiWire capable devices

Revision 1.3 (December 2005)

- Serial resistors of 100R (R53–R55 and R57–R64) are added on JTAG signals (TCK, TMS, TDI, and TDO1 to TDO8) to improve stability if target connections are longer than 20 cm and/or bed-of-nails is used
- Resistors (R56, R65) of 33 kΩ are added on case-to-GND connections of DSUB9 and DSUB25 connectors
- New R68 on VCCO: discharges targets
- New U31, R66, R52, C38: adds Reset capability
- U31 assembled SN74LVC1G07DBV instead of SN74LVC1G125DBV (open-drain Reset)
- C27 changed from 1 µF tantalum to 100 µF / 35 V aluminum
- R51 (3k3) added between TTMS and VCCO; should avoid bouncing on TMS if already supplied targets are connected (irregular fuse check)
- Both 2-wire and 4-wire JTAG modes are supported on SpyBiWire capable devices

Assembly change on 1.3 (January 2005)

- R52 changed from 22 kΩ to DNP

Hex File Formats

This appendix describes the Intel hex object format and the TI-txt file format.

Topic **Page**

A.1 Intel Hex Object Format **74**
A.2 TI-txt File Format **75**

A.1 Intel Hex Object Format

The Intel hex object format supports 16-bit addresses and consists of a 9-character (4-field) prefix that defines the start of record, byte count, load address, record type, and a 2-character sumcheck suffix.

The record types, which are represented in the 9-character prefix, are:

00 = Data record [begins with the colon (:) start character].

01 = End-of-file record

Record type 00, the data record, begins with the colon (:) start character and is followed by the byte count, the address of the first data byte, the record type (00), and the sumcheck. The sumcheck is the twos complement (in binary) of the preceding bytes in the record including the byte count, address, and data bytes.

Record type 01, the end-of-file record, also begins with the colon (:) start character. The colon is followed by the byte count, the address, the record type (01), and the sumcheck.

To represent an address bus of more than 16 bits, other record types are defined:

02 = Extended Segment Address Record, segment-base address. Used when 16 bits is insufficient, this mode is identical to 80x86 real-mode addressing. The address specified by the 02 record is multiplied by 16 (shifted 4 bits left) and added to the subsequent 00 record address. This allows addressing of up to one megabyte of address space. The address field of this record must be 0000, the byte count is 02 (the segment is 16 bit). The least-significant hex digit of the segment address is always 0.

04 = Extended Linear Address Record, allowing for fully 32-bit addressing. The address field is 0000, the byte count is 02. The two data bytes represent the upper 16 bits of the 32-bit address, when combined with the address of the 00 type record.

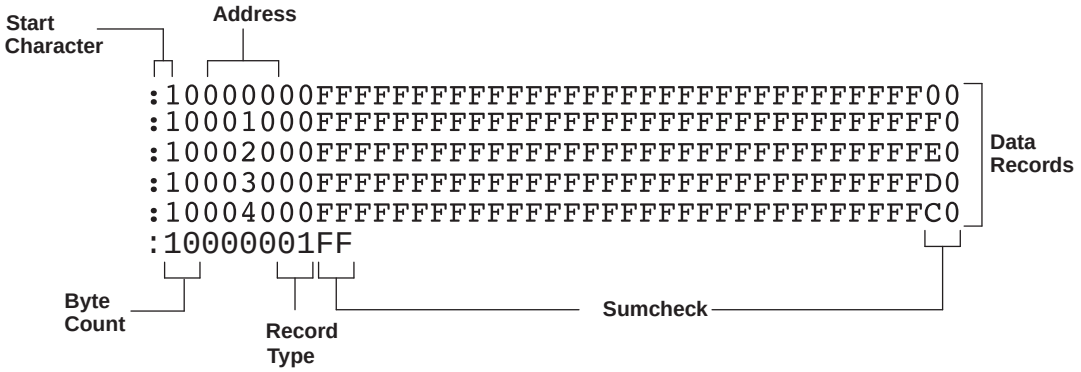


Figure A-1. Intel Hex Object Format

A.2 TI-txt File Format

The TI-txt file format used by the MSP-GANG430 is:

```
@ADDR1
DATA01 DATA02 ........ DATA16
DATA17 DATA18 ........ DATA32
........
DATAm ........ DATAn
@ADDR2
DATA01 ........ DATAn
q
```

Where:

@ADDR is the start address of a section (hexadecimal).
DATAn represents a data byte (hexadecimal).
q is the termination of the file.

For example:

```
@F000
31 40 00 03 B2 40 80 5A 20 01 D2 D3 22 00 D2 E3
21 00 3F 40 E8 FD 1F 83 FE 23 F9 3F
@FFFE
00 F0
Q
```

Restrictions:

- The number of sections is unlimited.
- Each line can have a maximum of 16 data bytes, except the last line of a section.
- Data bytes are separated by a single space.
- The termination tag q indicates end-of-file is mandatory.

Frequently Asked Questions

Topic	Page
B.1 FAQs 78	

B.1 FAQs

1. **Q. How can I verify which software version is installed?**

   A. Click the small icon in the top left corner of the GUI, then click "About Gang Programmer ..." or press F1. The versions of the installed software and hardware are displayed.

2. **Q. What software version should I use?**

   A. In general, it is recommended to use always the latest version, which can be found on this web page: http://focus.ti.com/docs/toolsw/folders/print/msp-gang430.html under Support Software. This web page is more frequently updated than the CD-ROM that is supplied with the MSP430 Development Tools. In particular, new members of the MSP430 family are supported only by the latest version of the GANG430 software. Older devices can still be programmed with older software versions.

3. **Q. How can I install a new software version?**

   A. After downloading the .zip file from above link, unpack it, and start the .exe file. Follow the on-screen instructions during installation. Older versions can coexist with newer ones.

   The Gang Programmer hardware contains firmware. **This firmware must always match with the version installed on the PC.** When the new software version on the PC is started the first time with Gang Programmer hardware connected, a window opens, recommending to upgrade the firmware. This should be followed. Next, a file dialog opens to select the file containing the firmware. This file resides in the directory where the Gang Programmer software was installed. The filename is similar to GANG430-xxx.txt, where xxx represents the revision number. **If the request to upgrade the firmware is denied, the programmer might not work correctly and might show unexpected behavior. The download process of the firmware should not be interrupted.**

4. **Q. Can I upgrade the firmware manually?**

   A. If, for any reason, the previously described firmware upgrade fails, the upgrade can be performed manually: From the GUI, click Maintenance → Category → Upgrade Firmware, then click Execute. Follow the file selection dialog as described in the previous question.

5. **Q. Where can I find examples of how to use the GANG430.DLL?**

   A. Examples can be found in C:\ADT430\GANG\DLL_usage_Examples (the root directory might be different, depending on the chosen installation path).

6. **Q. What is the maximum allowable length of the connections between the Gang Programmer hardware and the targets?**

   A. The length of the wires that connect the Programmer to the targets should not exceed 60 cm to ensure sufficient quality of the JTAG signals. In case of problems, try these solutions:

   - Reduce length of wires as much as possible.
   - Use an oscilloscope to check the TCK and TMS signals at the targets. There should be no overshoot or undershoot exceeding 0.3 V, and there should be no glitches on TCK. To correct overshoot/undershoot or glitches, serial resistors of 100 Ω up to approximately 200 Ω can be inserted in the respective signal. See the note in Section 1.3, Step 3.
   - The maximum frequency of TCK can be reduced by a factor of five by changing the value of JtagSpeed = 0 to JtagSpeed = 1 in the file Gang430.ini. This has no affect if SpyBiWire-Mode is used.

7. **Q. Can the Programmer hardware supply the targets with power?**

   A. Yes, the Gang Programmer can supply 150 mA (max) to the targets. The supply voltage can be adjusted between 1.8 V and 3.6 V in steps of 0.1 V. See also Section 1.3, Step 4.

8. **Q. Why do I get the error message 'Fuse is blown' when in fact it is not?**

   A. See the note in Section 5.3.

9. **Q. How can devices be programmed with unique data (e.g., a serial number)?**

   A. Each of the eight connected devices can be programmed with individual data. This could be required if the target application is the same for all devices, but a serial number or calibration data varies. This can be done using the function GangProgramTarget(). See the examples in C:\ADT430\GANG\DLL_usage_Examples (the root directory might be different depending on the chosen installation path).

10. **Q. Does the Gang Programmer reset the targets?**

    A. The reset signal (pin 11 on the 14-pin header) is pulled low for the interval set in Settling Time. This is an open-drain output, therefore, a pullup resistor is required on the targets. See also Section 2.1.1.

11. **Q. How can connected targets be reset using GANG430.DLL?**

    A. Use the DLL function GangSetSignals() as described in Section 4.1.

12. **Q. The GANG430 hardware appears nonfunctional: no LED lighting, not possible to connect from PC application.**

    A. This can happen if the firmware upgrade was interrupted or not successful. A manual firmware upgrade can solve this:

    a. Connect GANG430 via RS-232 to PC and to a power supply.
    b. Start the GUI (gang430.exe).
    c. The GUI may show an error message such as "Synchronization failed" or "The voltage supply can not be set". These error messages can be ignored.
    d. Click Maintenance → Category → Upgrade Firmware and click Execute.
    e. Select the appropriate firmware file (e.g., GANG430-143.txt) and click Open. This file is located in the directory where the GANG430 software is installed.
    f. The firmware upgrade should be performed. If this does not solve the issue then the GANG430 should be returned for repairs.

13. **Q. Why do devices programmed in 2-wire (Spy-Bi-Wire) mode not run after programming, even when they are supplied with power?**

    A. Target devices do not run after programming due to limitation of the GANG430 hardware. It is possible to run every target separately using the GUI.

    a. Open the Target Connector window.
    b. Select the desired device to run, e.g., TDO/TDI1 for device #1.
    c. Check TDO/TDIx 1 checkbox.
    d. Click Set.

    Alternatively, the DLL function GangSetSignals() can be used. This function is described in Section 4.1.

14. **Q. The target stops responding to the GANG430. What is wrong?**

    A. The MSP-GANG430 software in JTAG mode releases the CPU for short periods of time to normal operation and, within this time, the CPU might stop responding due to loading incorrect DCO calibration data (0xFF) into the DCO registers. Under these circumstances, the MSP-GANG430 cannot access the MSP430 through JTAG anymore. The Spy-Bi-Wire mode operates differently, and the CPU is never released to normal operation. This allows the MSP-GANG430 to control the MSP430F2xx under any circumstance. To avoid this problem:

    Before the DCO calibration values are written into the DCO registers, they should be checked for erased data.

```
if ((CALDCO_16MHZ != 0xFF) && (CALBC1_16MHZ != 0xFF))
{
   DCOCTL = 0;                // Select lowest DCOx and MODx settings
   BCSCTL1 = CALBC1_16MHZ;    // DCO = 16MHz calibrated
   DCOCTL = CALDCO_16MHZ;     // DCO = 16MHz calibrated
}
```

    This method ensures that the CPU never stops operation because of an erased information memory segment A. The MSP-GANG430 always should be able to access the device through JTAG (4 wire) and Spy-Bi-Wire (2 wire).

    The next release of the MSP-GANG430 software will have the modified JTAG programming mode and the CPU will not be released for normal operation during programming. Until this time, the options are to either change the MSP430 software according to above code example or to do the programming access through Spy-Bi-Wire instead of 4-wire JTAG.

Preproduction Checklist

Table C-1. Preproduction Checklist for the Gang Programmer

Item No.		Check	Remarks
1	Is the latest software version installed on the programmer?	☐	
2	Are connections between programmer and targets <60 cm long?	☐	
3	Is the programmer supplied with 9 to 15 VDC and 300 mA?	☐	
4	Are the correct images loaded to the programmer?	☐	
5	Are the targets supplied with power by the programmer?	☐	
	Is the total power consumption of all targets less than 150 mA?	☐	
	Are pins 7 and 14 of the 25-pin Sub-D connector (or pin 2 on the 14-pin connectors connected to the targets?	☐	
6	Or are the targets supplied with power locally?	☐	
	Is the local supply voltage connected to pin 14 of the 25-pin Sub-D connector (or pin 4 on the 14-pin connector)?	☐	
	Is the local supply voltage in the correct range?	☐	

Document Revision History

Revision	Comments
SLAU101M	Updated Figure 2-1. Changed menu item Blow Fuse to Secure Device in Section 2.1.2. Changed FuseBlown to SecureDevice in Section 2.1.5.4. Added Select Image command in Section 3.1. Updated Write Target Selective in Table 3-1. Changed terminology from "blow fuse" to "secure device" in Section 3.5.2, Section 3.5.4, Section 4.1 (see GangLoadParameters, GangGetResult), Section 5.4. Updated FAQ #1.
SLAU101L	Updated Table 3-1. Changes to Section 1.3, Section 2.2, Section 3.5.10, Section 3.5.11, and Section B.1.
SLAU101K	Added DLL function GangSelectImage() in Section 4.1. Updated FAQs in Section B.1. Added preproduction checklist in Appendix C.
SLAU101J	Updated Table 2-3 and Table 3-1. Updated Section 2.2 and Section 4.1. Added Section 3.5.14. Updated FAQs in Section B.1.
SLAU101I	Added FAQ on how to determine current software version in Section B.1.
SLAU101H	Updated Section 2.1.1, Section 2.1.5.1, Section 3.5.2, "GangLoadParameters" in Section 4.1

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

PROGRAMMER IMPORTANT NOTICE

Texas Instruments (TI) provides the enclosed product(s) under the following conditions:

This Programmer is intended for use for ENGINEERING DEVELOPMENT and PRODUCTION DEVICE PROGRAMMING PURPOSES and is not considered by TI to be a finished end-product fit for general consumer use. Persons handling the product(s) must have electronics training and observe good engineering practice standards. As such, the goods being provided are not intended to be complete in terms of required design-, marketing-, and/or manufacturing-related protective considerations, including product safety and environmental measures typically found in end products that incorporate such semiconductor components or circuit boards. This Programmer does not fall within the scope of the European Union directives regarding electromagnetic compatibility, restricted substances (RoHS), recycling (WEEE), FCC, CE or UL, and therefore may not meet the technical requirements of these directives or other related directives.

Should this Programmer not meet the specifications indicated in the User's Guide, the Programmer may be returned within 30 days from the date of delivery for a full refund. THE FOREGOING WARRANTY IS THE EXCLUSIVE WARRANTY MADE BY SELLER TO BUYER AND IS IN LIEU OF ALL OTHER WARRANTIES, EXPRESSED, IMPLIED, OR STATUTORY, INCLUDING ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE.

The user assumes all responsibility and liability for proper and safe handling of the goods. Further, the user indemnifies TI from all claims arising from the user's handling or use of the goods. Due to the open construction of the product, it is the user's responsibility to take any and all appropriate precautions with regard to electrostatic discharge.

EXCEPT TO THE EXTENT OF THE INDEMNITY SET FORTH ABOVE, NEITHER PARTY SHALL BE LIABLE TO THE OTHER FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES.

TI currently deals with a variety of customers for products, and therefore our arrangement with the user is not exclusive.

TI assumes **no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein.**

Please read the User's Guide and, specifically, the Warnings and Restrictions notice in the User's Guide prior to handling the product. This notice contains important safety information about temperatures and voltages. For additional information on TI's environmental and/or safety programs, please contact the TI application engineer or visit www.ti.com/esh.

No license is granted under any patent right or other intellectual property right of TI covering or relating to any machine, process, or combination in which such TI products or services might be or are used.

FCC WARNING

This Programmer is intended for use for **ENGINEERING DEVELOPMENT, PRODUCTION DEVICE PROGRAMMING PURPOSES** and is not considered by TI to be a finished end-product fit for general consumer use. It generates, uses, and can radiate radio frequency energy and has not been tested for compliance with the limits of computing devices pursuant to part 15 of FCC rules, which are designed to provide reasonable protection against radio frequency interference. Operation of this equipment in other environments may cause interference with radio communications, in which case the user at his own expense will be required to take whatever measures may be required to correct this interference.

Mailing Address

Texas Instruments

Post Office Box 655303

Dallas, Texas 75265