DC590 USB Controller Specifications and Technical Manual

Linear Technology Confidential – Customer Use Only

This information is provided as-is. The DC590 board is optimized for demonstrating the features of the products and is not intended for serious data collection or as a reference design for a real product. Please contact applications engineering for assistance in applying Linear Technology parts.

This board is a universal SPI and I2C controller board intended to GREATLY simplify demo board design for LTC products that have these interfaces. PC software development is also very easy, and the end result is extremely easy to use, due largely to the fact that there is a single software package on the LTC website that controls all modules that plug into DC590. When a new board is added to the family, the software package is updated, and customers that already have the software installed can simply update in the same way SwitcherCad updates.

Purpose –
The purpose of this document is to completely describe the hardware and firmware of Demo Circuit 590 LTC USB serial communications controller.

Host PC interface

All communications with the controller are done using ASCII characters. To avoid ambiguity, the characters 0 through 9 and A through F (capital only) are reserved for data. Information is sent to the controller through one of two interfaces to the USB port. The interface used depends on which driver is loaded on the PC the first time a controller is plugged into it.

The first is the "Virtual Com Port" driver which makes the USB interface look like a traditional serial port. This is extremely easy to use, as all buffering is taken care of and any terminal program can be used.

The second is the set of direct drivers, which are described in the FTD2xx Programmer's guide, Version 2.0.1, available at www.ftdichip.com.

Protocol Summary:

Command	Function	Notes
Q	Receive Byte *	I^2C only, send ACK
R	Receive Byte *	In I^2C mode, sends NACK
(lower case) r	Receive 7 bits	In SPI mode only.
Sxx	Send byte (2 hex chars) *	SPI and I^2C. **Returns 'N' if slave NACKs.**
Txx	Tranceive (send and receive) *	(SPI only)
X	CS=1	Use for GPIO in I^2C mode
(Lower case) x	CS=0	Use for GPIO in I^2C mode
(Lower case) s	Send start condition	(I^2C only)
(Lower case) p	Send stop condition	(I^2C only)
H	Wait for SDO=1	(SPI only) 'T' on timeout
L	Wait for SDO=0	(SPI only) 'T' on timeout
P	PingPause (echo P, wait 5 ms)	
Z	Send line feed to host (0x0A)	
I	Read demo board info.	
(Lower case) 'i'	Read contoller ID and firmware rev	
O	Turn isolated power on	
(Lower case) o	Turn isolated power off	
MS	Switch to isolated SPI mode	
MI	Switch to isolated I^2C mode	
MX	Switch to auxillary I^2C mode	
(Lower case) t	Begin recording loop	
(Lower case) u	Stop recording loop	
(Lower case) v	Echo loop	
(Lower case) w	Loop until another command rec'd	
0x80	Hang. Yup, crash.	
0x81	No operation	
New for DC590B!		
G, g	Pin 14 High, Low, respectively	
Kxy	Bang pin, x argument is '0' or '1' for low or high, y is pin address, see below.	
Uxxy	Write Port D bus, xx is ASCII Hex data, y is strobe pin address, strobed low.	

*Note: Commands Q, R, r, S, T will return an "X" if no daughter board is connected. This is done by looking for a pullup resistor on the Auxilliary SDA pin (pin 9).

Pin Addresses for U, K commands:
'0' – RB0
'1' – RB1
'2' – RB2
'3' – RC0
'4' – RC1
'5' – RC2

Example 1 - LTC1592
xSA0S40S00XxSA0S80S00X
(set to +/- 5 volt range, set output to ¼ scale, then same range ¾ scale.)

Example 2 – LTC2440
RxLT00RRRXx LT20RRRXx LT28RRRXx LTF8RRRXx
(Read result, program OSR256, Read result, program OSR512, Read result, program OSR1024, Read result, program OSR32768.)

Example 3 – LTC 1695 (SMBus fan controller)
MI sSE8 S3F S00 S00 p

Protocol Details –

The command protocol was designed to be extremely generic to allow communication with any SPI or I^2C device. In SPI mode, the clock line idles low; data changes on the falling edge of the clock and is read on the rising edge.

Important *– Notes on Postage Stamp Detection*
Several firmware features were added to facilitate rapid detection of a disconnected board and subsequent flushing of the command buffer. Board presence is determined by looking at the state of the auxillary SDA line, which is pulled high by the postage stamp. The commands affected are Receive, Send, Transcieve, Wait for Low, Wait for High, and Identify. Refer to the individual command descriptions for details.

Receive, send ACK (Q)
Receive 8 bits from an I^2C device with an ACK. After the data is read, it is sent as 2 ASCII hex characters to the host.

If the postage stamp is not connected, this command will be skipped and will return nothing to the host.

Receive (R)
Receive 8 bits from the SPI or I^2C device. After the data is read, it is sent as 2 ASCII hex characters to the host. In I^2C mode, the controller sends a NACK.

If the postage stamp is not connected, this command will be skipped and will return nothing to the host.

Send (Sxx)
Send 8 bits to the SPI or I^2C device. After the host sends the Send command, 2 ASCII hex characters are expected. After they are received, they are sent to the device. Sending characters other than 0-9, A-F will result in undefined data being sent to the device. In I2C mode, if the intended receiving device does not acknowledge or there is no device present, an ASCII ‘N’ will be returned.

If the postage stamp is not connected, this command will be skipped and return an ASCII ‘X’ to the host. This facilitates board detection for write-only devices such as DACs.

Transceive (Txx)
Send and receive data simultaneously with an SPI device only. 2 ASCII hex characters are expected after this command. After they are received, they are sent to the SPI device, while 8 bits are received from the device simultaneously. The received bits are sent to the host as 2 ASCII hex characters. This command is not available in I^2C mode.

If the postage stamp is not connected, this command will be skipped and will return nothing to the host.

Chip Select Control (X, x)
The chip select line is raised and lowered by sending a capital 'X' or lower case 'x', respectively.

I^2C start and stop conditions (s, p)
In I^2C mode, start and stop conditions are sent to the I^2C device with the 's' and 'p' command, respectively.

Wait for state on MISO line (H, L)
The 'H' and 'L' commands cause the controller to wait for a high level or low level on the MISO line, respectively. A timeout is set for 5 seconds – if the target state is not detected, a 'T' is sent out and the controller continues executing commands. (This may change – perhaps including the ability to set the timeout to some other value.)

If the postage stamp is not connected, these commands will terminate immediately. This includes unplugging a board while waiting.

Ping and Pause 5 mS (P)
This command echos a 'P' and pauses 5 ms. The pause is useful during programming of the serial EEPROM.

Send line feed (hex 0x0A) (Z)
This command sends a linefeed and is useful as a delimiter in the data stream to the host.

Identify postage stamp board (I)
Reads the contents of the postage stamp EEPROM until a hex 0x00 is reached. Format as follows:
LTCxxxx,cls,Dxxxx,xx,yy,DC,DCxxxx----------,\n\0

Example: LTC2440,cls,D2440,01,02,DC,DC570

Identify Controller Board and firmware revisioin (i)
This command sends the following string:
USBSC,PIC,xx,yy,DC,DC590----------,\n\0

Where USBSC is the board name, xx,yy is the firmware revision and DC590 is the demo circuit number, and \n is a line feed (Hex 0A.)

Example: USBSC,PIC,01,03,DC,DC590

Isolated power control (O, o)
The capital O command turns on the isolated switching power supply. Lower case o turns it off.

Communication mode switching (MS, MI, MX)
MS switches to SPI mode. MI switches to the main isolated I^2C mode. MX switches to the auxillary I^2C port for talking to the serial EEPROM.

Record / Playback operation –
This feature was included to allow continuous looping of a sequence of commands so that the host PC does not need to continuously send them. For example, to continuously read the LTC2440 in OSR256, program the appropriate sequence as follows:
Record: t xLT00RRRX u
Verify: v (Should echo "RxLT00RRRXx")
Playback: w

If the host's receive buffer fills up, the sequence will pause. Sending any command to the controller terminates playback.

The subset of commands available for recording are as follows:

Command	Function	
R	Receive Byte	Returns a or n in I^2C mode
Sxx	Send byte (2 hex chars)	Returns a or n in I^2C mode
Txx	Tranceive (send and receive)	(SPI only)
X	CS=1	Use for GPIO in I^2C mode
(Lower case) x	CS=0	Use for GPIO in I^2C mode
(Lower case) s	Send start condition	(I^2C only)
(Lower case) p	Send stop condition	(I^2C only)
H	Wait for SDO=1	(SPI only)
L	Wait for SDO=0	(SPI only)
P	PingPause (echo P, wait 5 ms)	
Z	Send carriage return to host	

Thus there can be no comm mode switching or turning power on and off in a recorded sequence.

Electrical Connections –
All connections are through a 14 pin, 2mm male header. Postage stamp board is connected to controller via a ribbon cable.

Top View:

1) 7V unreg	2) 5V reg
3) Isolated GND	4) SCK / SCL
5) MISO	6) Chip Select
7) MOSI / SDA	8) Isolated GND
9) AUX SDA	10) Main VCC
11) AUX SCL	12) Main GND
13) Isolated GND	14) N.C.

Pin 1 – Isolated, unregulated 7 – 8 volts @ 25 mA from controller board to stamp.

Pin 2 – Isolated, regulated 5 volts @ 100 mA from controller board to stamp.

Pin 3 – Isolated ground

Pin 4 – Serial clock to stamp and I^2C SCK. This is an open collector with 2k pull up)

Pin 5 – Master In, Slave Out (Data from stamp to controller) This input is pulled up to 5 volts through an LED, Schottky diode, and 5k resistor. The slave device must be able to sink 0.6 mA in the low state.

Pin 6 – Chip select to stamp (Open collector with 5k pullup)

Pin 7 – Master Out, Slave In (Data from stamp to controller in SPI mode) and I^2C SDA. This is an open collector in series with 170 ohms, with 2k pullup. Voh = 5V, Vol = 0.6V.

Pin 8 – Isolated ground

Pin 9 – Non-isolated SDA for EEPROM
Pin 10 – Non-isolated VCC (5 volts)
Pin 11 – Non-isolated SCL for EEPROM
Pin 12 – Non-Isolated Ground
Pin 13 – Isolated Ground
Pin 14 – No Connection

Pins 9-12 are intended to be connected to a 24LC01 / 02 I^2C EEPROM on the postage stamp board. Tie its address lines (A0, A1, A2) to non-isolated ground such that the I^2C address is 0xA0. Pull up the SDA and SCL to non-isolated VCC with 5k resistors. These lines are pulled down on the controller board, allowing fast detection of a disconnected stamp board.