**LDC13xx/16xx EVM Protocol description**

**Version date: Jan 15, 2015**

**Connection**

LDC13xx/16xx communication is accomplished via driver-simulated COM protocols. Settings are:

Baud rate 115200

No termination CHAR

Data bits 8

Parity None

Stop bit 1

Flow control: None

**Write register command**

Write register command is used to set desired register settings. It is formed as follow:

4C 15 01 00 04 2A XX YY YY ZZ

Where

4C 15 01 00 04 2A = Write register command header.

XX is the register address to be written, expressed as HEX. For example, a register d12 is x0C, so XX = 0C

YY YY is the 2 bytes of data to be written, in HEX, MSB 1st.

ZZ is CRC-8 check for the command string, which is explained below.

Each HEX value represents corresponding ASCII character, which is written to COM port.

For example, if we need to write to register 11 (x1B) a value of x820D, the command is

4C 15 01 00 04 2A 1B 82 0D A3. In ASCII it appears as 

After writing a register it is required to read back 32 bytes – MCU response to a command. Convert ASCII string into array of 8-bit numbers. It consists of the command sent + 21 bytes of 00 + CRC value. In the above example, correct response is 4C 15 01 **00** 04 2A 1B 82 0D A3 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00A3.

4th byte (indicated in red above) is an error field. If this byte is 00, there is no error.

**Read register command**

Read register command is used to check LDC register settings. Read process consists of 2 steps:

1. **Set** register command, which is required to indicate which register is to be read. The command format is: 4C 15 01 00 02 2A XX ZZ, where XX is the register address to be read, and ZZ is CRC-8 check for the command string. For example, to read back register 1B the command is 4C 15 01 00 02 2A 1B 10. After sending this command it is required to read 32 bytes. The 4th byte will indicate an error, if any.
2. Read register command, in the format 4C 14 01 00 02 2A 02 ZZ, where ZZ is CRC-8 check. After sending this command it is required to read 32 bytes, as noted above. The 4th byte will indicate an error, if any. 7th and 8th bytes are MSB and LSB bytes of the register value. For example, if we read register x1B value, the response is 4C 14 01 **00** 03 2A **82 0D** C6 A3 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 C6

**Start streaming command**

To optimize data throughput a special command is implemented, which will continuously output data from LDC devices to COM port.

To start streaming, send 4C 05 01 00 06 01 29 04 04 30 2A C1. After sending this command it is required to read 32 bytes. 4th byte will indicate an error, if any.

After the stream is initialized, read 32 bytes at a time. An example read:

4C30 0100 092A **0151 42CB** **0150 5FFF** 1C00 0000 0000 0000 0000 0000 0000 0000 001C

Bytes 7-10 are the Ch0 data (marked in green), MSB 1st, bytes 11-14 are Ch1 data (marked in blue), MSB 1st. In this example, Ch0 data is x015142CB (22,102,731 in decimal), Ch1 is 01505FFF (22,044,671). Further conversion of the raw data to the frequency of oscillation is described in the LDC161x/131x datasheet

**Stop streaming command**

To stop the streaming, send the command 4C 06 01 00 01 01 D2. After sending this command it is required to read 32 bytes. 4th byte will indicate an error, if any.

**CRC-8 check**

CRC check is well described in <http://en.wikipedia.org/wiki/Computation_of_cyclic_redundancy_checks>

Please refer to the link for more detailed information.

The LDC 13xx/16xx EVMs use CRC-8 with polynomial coefficient indices of 8, 2, 1, and 0.

Outlined below is one of the possible implementations:

Convert each byte in the command to an array of Booleans (T = 1, F = 0), MSB being index 1. For example, xA7 becomes an array (T,F,T,F,F,T,T,T).

Concatenate the array of bytes, 1st byte being indices 1-8, 2nd – bytes 9-16 etc. For example, xA70F becomes (T,F,T,F,F,T,T,T,F,F,F,F,T,T,T,T). Record the array size N.

Form C = (T,F,F,F,F,F,T,T,T) – CRC array. Note that the indices of the non-zero coefficients are 8, 2, 1, 0.

Append to the Boolean command array an array of 8 False (F,F,F,F,F,F,F,F) at the end. Denote this array A.

For loop for i=1 to N

IF A[i] = True then

Get B = array A subset starting at index i, length 9

B’ = B XOR C

Replace B in A with B’

End IF

End loop

Take the last 8 Booleans of the updated A, convert it to byte (MSB 1st). For example, if last 8 elements in A are (T,F,T,T,F,F,T,T), it converts to xB3

This is desired CRC-8 check value (ZZ value in the document)