Using the LP8501 Evaluation Kit

User's Guide

TEXAS INSTRUMENTS

Literature Number: SNVU451
July 2014

Contents

List of Figures

User's Guide
SNVU451 – July 2014

LP8501 Evaluation Kit

1 General Description

This document describes how to get the LP8501 evaluation board up and running, how to use evaluation software, and how to get started with the programming of the lighting engines. The LP8501 provides flexibility and programmability for dimming and sequencing control. Each LED can be controlled directly and independently through the serial bus (in other words, without programming the engines), or LED drivers can be controlled by programming the lighting engines. The LP8501 has three independent lighting engines, so it is possible to form three independently programmable LED banks. LED drivers can be grouped based on their function so that, for example, the first group of drivers can be assigned to the keypad illumination, the second group to the “funlights”, and the third group to the indicator LED(s). Each group can contain 1 to 9 LED driver outputs. Instructions for lighting engines are stored in the internal program memory. The total amount of the program memory is 96 instructions, and the user can allocate the memory as required by the engines. The LP8501 is programmed using I^2C-compatible serial bus. It is possible to write programs for the LP8501 in the form of binary data, but the programming tools described in this document offer a more convenient way to write (and read) the registers and the SRAM memory and to program the device.

2 What Is Needed

To get started the user needs:

- A text editor
- LP8501 evaluation kit hardware
- LP8501 evaluation software (LP8501.exe)
- LP8501 compiler (MEASM.EXE)

A text editor is used to create source code for the compiler. In this case, PSPad is the text editor, but the user may use whichever editor most convenient. PSPad is a freeware editor, © 1991 - 2007 Jan Fiala. Please see the PSPad copyright notice at the end of this document. PSPad text editor is distributed with the LP8501 evaluation software distribution media or it can be downloaded from http://www.pspad.com/

3 Copying the Software

PSPad does not require installation, it can be simply unpacked into any directory. The archive contains subdirectories and must be unpacked with subdirectory preservation enabled. Also copy the Metka.ini–file from the evaluation software distribution media (CD-ROM or memory stick) into the Syntax-folder of PSPad (for example C:\Program Files\PSPad editor\Syntax), in case the file does not exist there already. Metka.ini–file contains customizing information for the text editor, and it must be saved into the Syntax-folder.

The LP8501 compiler (MEASM.exe) and LP8501 evaluation software can be copied to the PC’s hard disk and run without installation. The following files should be copied: LP8501.exe, regmap.ini, usblptio.dll, rtl60.bpl and MEASM.exe from the distribution media. All the files must lie in the same directory. Please avoid filenames or directory names containing a space character, since the compiler may fail when applied to filenames containing a space character. The evaluation software runs under Windows®.

4 PSPad Customization

Once the PSPad editor has run, the LP8501 editor should be customized as follows:

1. Select Settings menu > Highlighter settings.
2. Select Specification tab. See Figure 1.
3. On the left (highlighter) list, click on one of bolded highlighters (marked with **<not assigned>** -tab).
4. On the right side is list of user highlighters, select the METKA highlighter and click on it. See Figure 1

Also set the compiler search path and parameters for the MEASM.exe, as shown in Figure 2. The shown file path shown may need to replaced with the actual path. Remember that the path should not contain the *space character*. Tag the Capture Program Output Window as shown in Figure 2. Accept highlighter settings by clicking OK. Finally, in the main window show the LOG window by pressing CTRL + L. All software needed should be now ready for writing and compiling the first program. Note: The maximum length of a source code line is 140 characters. Lines that are longer than 140 are not assembled correctly.

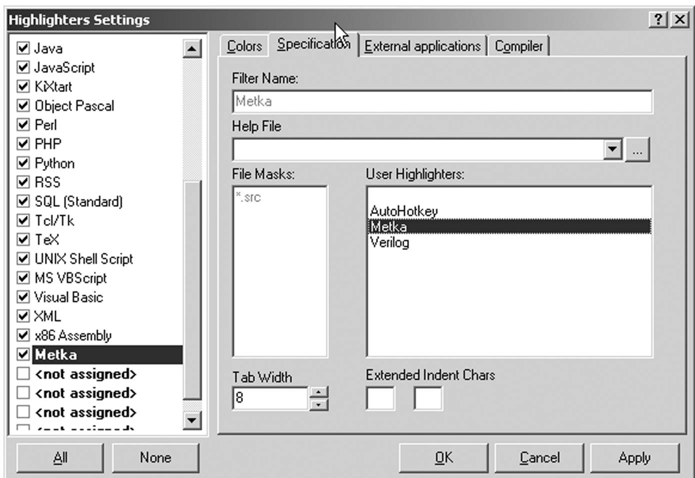

Figure 1. PSPad Highlighter Specification Settings for LP8501

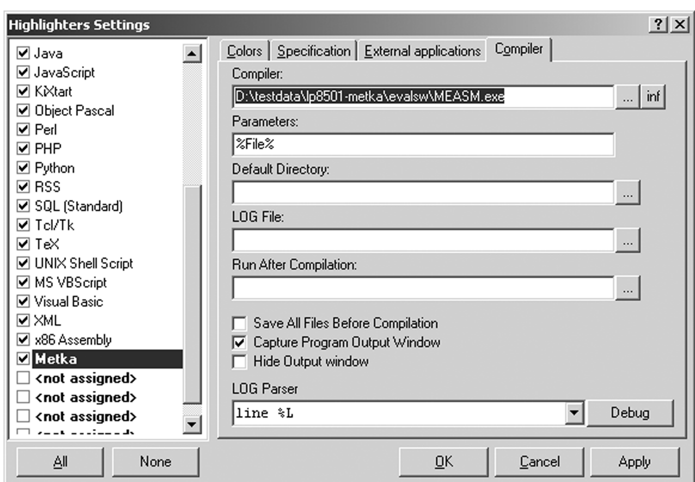

Figure 2. PSPad Highlighter Compiler Settings for LP8501

5 Hardware Set-up

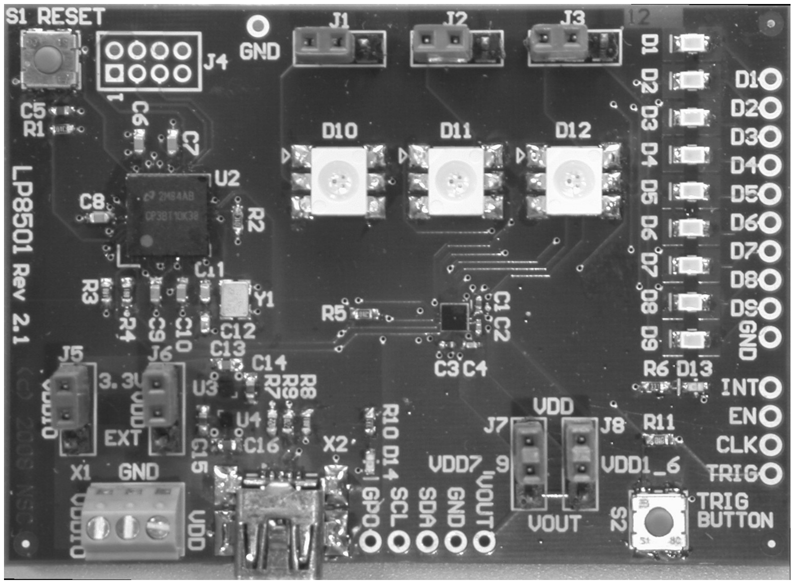

Figure 3. LP8501 Evaluation Board

The LP8501 evaluation kit is based on a one board system, where the USB communication and evaluation related components are assembled onto one board. (See Figure 3.) The evaluation board was designed specially for evaluation and therefore is not optimized for the smallest layout size. The components are physically large to make changing of the value easy if needed. The LP8501 input voltage V_{DD} is supplied from the USB board or from an external voltage applied to the X1 connector. The voltage supplied at the V_{DD} input of the device can be selected using jumper options for the J6 connector. Also the V_{DDIO} can be selected if it is powered from USB or from external voltage supply (X1 connector). Voltage feed for V_{DDIO} can be selected with J5. Voltage on the USB port is 5.0V and the maximum current is 500 mA. There is a regulator on the USB board, which can reduce the voltage to 3.0V or to 3.9V, depending on the evaluation program settings.

On the evaluation board itself there are 6 pin connectors for demonstrating some of the possible application options. As previously mentioned J6 is for selecting where from input voltage V_{DD} is fed. If looking at pin connectors in Figure 3, connecting upper and center pin with jumper selects that V_{DD} is fed from USB and connecting lower and center pin with jumper selects that V_{DD} is fed from connector X1. In connector X1 left connection point is for V_{DDIO}, center for Ground and right for V_{DD}. Pin connectors J1 J2, J3 at the top edge are for selecting whether LP8501 LED outputs are connected to WLEDs or RGBs. Pin connector J1 connects outputs 1 (Green), 2 (Blue) and 7 (Red) to RGB LED D10 (left and center pins are connected with jumper) or to WLEDs D1 (LED output 1), D2 (LED output 2) and D7 (LED output 7), when right and center pins are connected with jumper. Pin connector J2 connects outputs 3 (Green), 4 (Blue) and 8 (Red) to RGB LED D11 (left and center pins connected with jumper) or to WLEDs D3 (LED output 3), D4 (LED output 4) and D8 (LED output 8), when right and center pins are connected with jumper. Pin connector J3 connects outputs 5 (Green), 6 (Blue) and 9 (Red) to RGB LED D12 (left and center pins connected with jumper) or to WLEDs D5 (LED output 5), D6 (LED output 6) and D9 (LED output 9), when right and center pins are connected with jumper. Pin connectors J7 and J8 are for determining whether the LED output supply voltages VDD1_6 and VDD7_9 are fed from V_{OUT} or from V_{DD}. When upper and center pins of J7 are connected with jumper V_{DD} is supplied to VDD7_9 (LED outputs 7, 8 and 9 are powered from V_{DD}). When bottom and center pins of J7 are connected with jumper V_{OUT} is supplied to VDD7_9. When upper and center pins of J8 are connected with jumper V_{DD} is supplied to VDD1_6 (LED outputs 1–6 are powered from V_{DD}). When bottom and center pins of J8 are connected with jumper V_{OUT} is supplied to VDD1–6. See table below for reference.

Jumper #	Right Pin	Center Pin	Left Pin
J2	RGB LED D10 in use (LED1 = G, LED2 = B, LED7 = R)		
		WLEDs in use (LED1 = D1, LED2 = D2, LED7 = D7)	
J3	RGB LED D11 in use (LED3 = G, LED4 = B, LED8 = R)		
		WLEDs in use (LED3 = D3, LED4 = D4, LED8 = D8)	
J4	RGB LED D12 in use (LED5 = G, LED6 = B, LED9 = R)		
		WLEDs in use (LED5 = D5, LED6 = D6, LED9 = D9)	

Jumper	J5		J6		J7		J8	
Upper Pin	V_{DDIO} supplied from USB		V_{DD} supplied from USB		V_{DD} supplied to VDD7_9		V_{DD} supplied to VDD1_6	
Center Pin		V_{DDIO} supplied from external voltage supply		V_{DD} supplied from external voltage supply		V_{OUT} supplied to VDD7_9		V_{OUT} supplied to VDD1_6
Lower Pin								

Connecting the Evaluation Board to the Computer

1. Check that the jumpers on the evaluation board are on wanted positions.
2. Connect the USB board to your computer using a USB cable
3. Start the evaluation software *LP8501.exe*
4. Reset the LP8501 circuit by clicking the *Soft Reset* button on upper right corner of the window. In the message box that appears, click OK to confirm the register reading.
5. The evaluation kit is now fully up and running and the device can be controlled through the PC software. Figure 4 shows the evaluation software user interface (Control Panel).

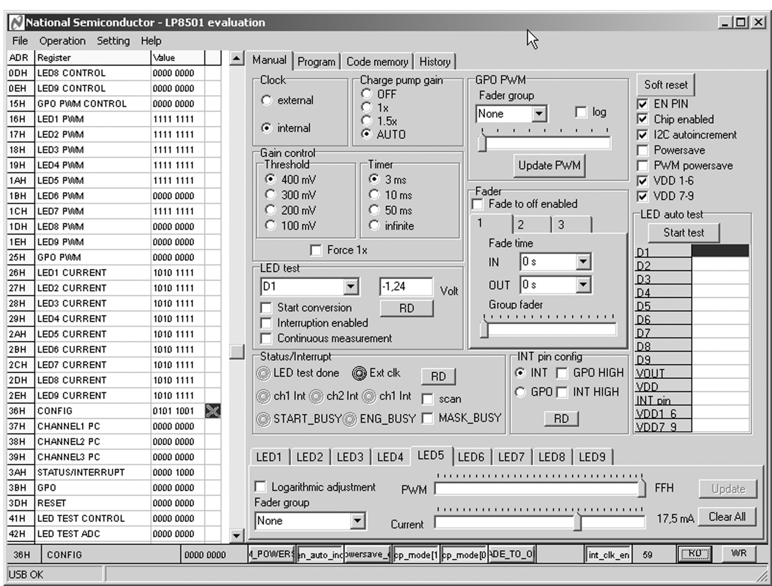

Figure 4. LP8501 Evaluation Software Control Panel

The USB OK message should be visible on the status bar (shown in the lower part of the window). If the USB communication is not working correctly, shut down the evaluation software and unplug the USB cable. Plug in the USB cable again and reset the USB interface board by pressing the RESET button on the USB board. Wait about 5 seconds and repeat steps 4 to 5 above.

6 Controlling the Evaluation Board

The evaluation software provides read-write control over the registers within the LP8501. Bits can be set from a logical '1' to a logical '0' or vice versa by a mouse click and for some settings there with a slider control. On the left of control panel all the registers are listed (see Figure 5) and once checking the appropriate register its value is stated on the bottom of the window (See Figure 6.) From this view one can see detailed description of the bits. The registers can also be written or read through clicking the bits and pushing WR-button or RD-button at the bottom bar.

Figure 5. Register View

Figure 6. Detailed Register View

The register views can be seen all the time. Otherwise controls are set into four tabs. Manual tab (open in Figure 4) includes the controls for almost all of the functions in the device. From the Program tab (See Figure 7) the user can control the lighting engines of the device. Code memory tab (see Figure 8) is used to read and write the device program memory. From History tab (see Figure 9), the user can see all the register reads and/or writes performed when controlling the device.

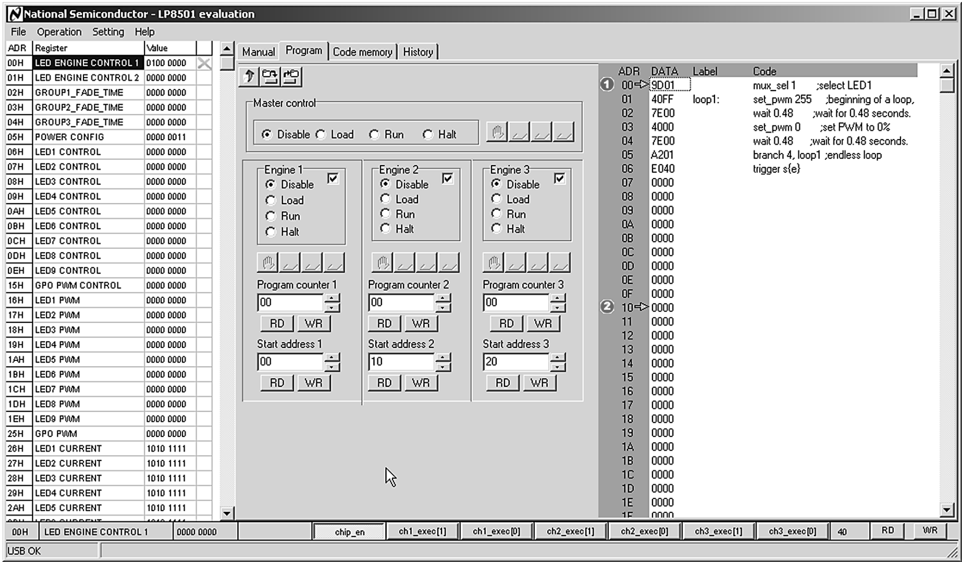

Figure 7. Program Tab of the LP8501 Evaluation Program

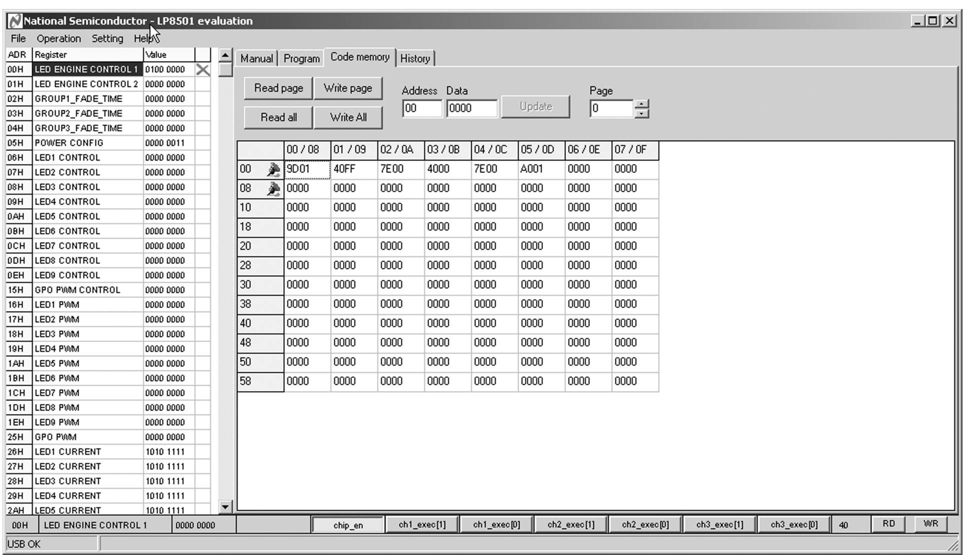

Figure 8. Code Memory Tab of the LP8501 Evaluation Program

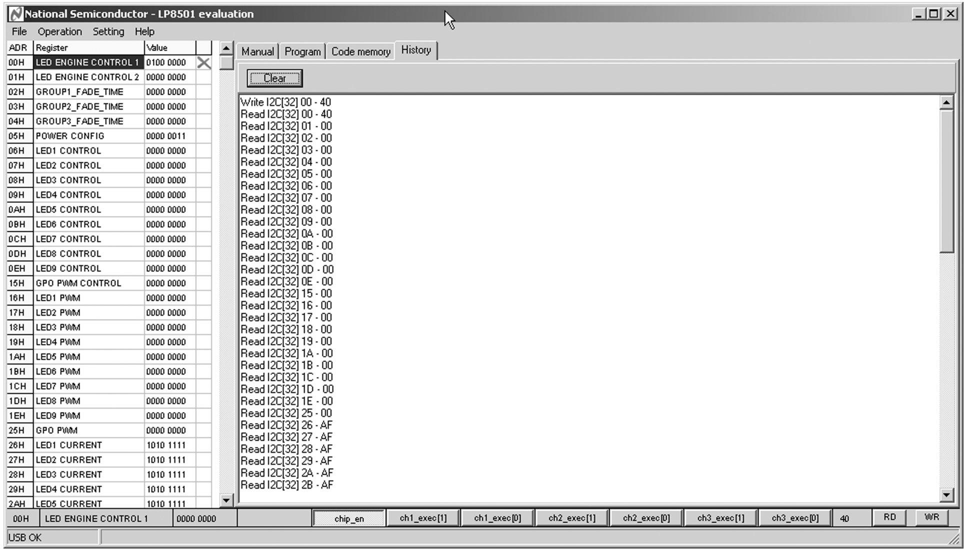

Figure 9. History Tab of the LP8501 Evaluation Program

Enabling, Powersave Modes, Supply Voltage Controls

From the upper rightmost corner of the Manual tab can be found general controls. From there one can find the Soft Reset-button, device enables, powersave settings, LED output supply voltage controls and I^2C autoincrement feature enable checkbox. Resetting device by pushing the Soft Reset-button writes FFh to Reset register 3Dh. Checking the Chip enabled writes '1' to bit 6 in register 00h, which enables the device itself. The EN checkbox should be checked by default. The EN-signal comes from the USB board and checking off EN-signal the voltage is disconnected and device is reset. Note that after checking EN off and back on, chip enable bit is set to '0'. I^2C autoincrement is set to '1' by default. This enables the autoincrement feature for I^2C writes. Powersave and PWM powersave check boxes enable the powersave modes when boxes are checked. LED output supply voltage controls VDD1_6 and VDD7_9 should be checked if the corresponding outputs are connected to VOUT (see Section 5 second paragraph). Leaving them unchecked means that the corresponding outputs are powered from VDD (or another external voltage supply).

Charge Pump, Clock, Gain Controls

On the upper leftmost corner of the Manual tab, beside the register view, can be found the controls for clock, charge pump and gain change. When clock is set to external (int_clk_en = '0'), user must connect external 32 kHz clock signal to CLK testpoint in evaluation board. Internal clock (int_clk_en = '1') selects the LP8501 internal clock. The charge pump can be disabled, set to 1x, 1.5x or AUTO mode. In auto mode, best possible gain is set. From Gain Control section gain Threshold and Timer values can be selected as well as forcing the charge pump to 1x testing mode. These are explained in detail in data sheet.

Controlling the LEDS

LEDs can be controlled from the lower part of the Manual tab. LED PWM and current values can be controlled, as well as grouping and logarithmic adjustment. Once LED is grouped, its PWM must be controlled from the Fader section in the Manual tab. GPO has its own PWM, logarithmic adjustment and grouping control in the GPO PWM section.

Controlling Groups

Groups can be controlled from the Fader section in Manual tab. Group fader controls the PWM value of LEDs belonging to that group. Different groups are on their own tabs. Fade time sets the times for fading in and out the LEDs in the group. In case *Fade to off* box is checked the LEDs will fade off, according to the OUT time set for them, when device is shut down (Chip enabled check box).

LED Test

LED test can be used for measuring voltages or detecting open and short circuits. LED test can be run from LED test section or from LED auto test. In LED test section LED or voltage to be measured is chosen from drop down menu. Before reading the value of the ADC (converted into voltage in software), user must check *Start conversion*. The selected LED or voltage is measured after *RD-button* is pushed. The value is read and updated only once. If the *Continuos conversion* is checked the value is updated every time *RD—button* is pushed. LED auto test goes through all the LED outputs and voltages and the results can be seen in a table.

Status and Interrupt

Various device statuses and interrupts can be seen from this section by pushing the RD-button. Scan checkbox enables the continuous scan of the statuses and interrupts.

INT Pin Config

INT pin can be configured as a GPO by selecting the GPO radial button in the INT pin config section. Selected as a GPO INT pin can be set high by checking the checkbox INT HIGH. Also GPO can be set high with the GPO HIGH checkbox.

7 LP8501 Programming

Programming Flow Chart

Figure 10 shows the typical programming flow of the LP8501. The program is first typed in with PSPad (or equivalent) text editor. Then the program is compiled into a hex and binary file. Finally the hex file is loaded into the LP8501's memory and tested.

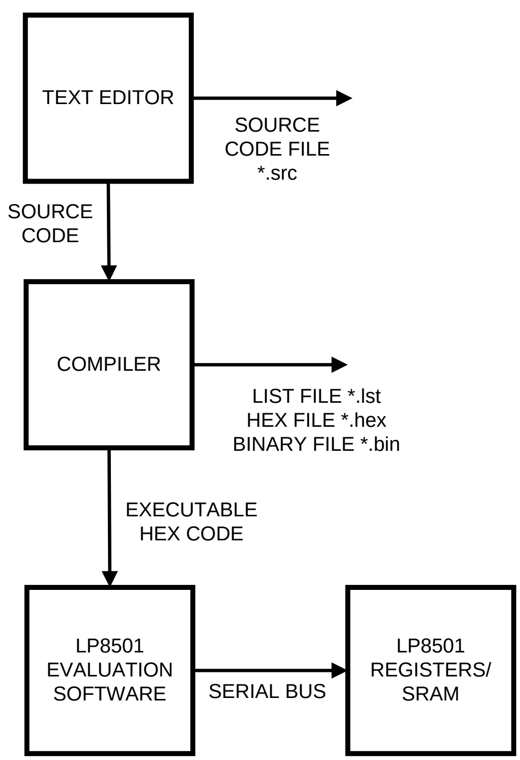

Figure 10. Programming Flow Chart

The Structure of a LP8501 Program

Figure 11 shows an example of an LP8501 program. When this program is run, the program will flash a red LED once per second.

```
.segment program1                   ;Beginning of a segment.
                mux_sel 1           ;select LED1
loop1:          set_pwm 255         ;beginning of a loop, set PWM full scale.
                wait 0.48           ;wait for 0.48 seconds.
                set_pwm 0           ;set PWM to 0%
                wait 0.48           ;wait for 0.48 seconds.
                branch 0, loop1     ;endless loop
```

Figure 11. Example Code of LP8501 Program

Although this program is short and simple it shows all the main parts of a typical LP8501 program.

Commenting

Commenting starts with a semicolon (;). The compiler will ignore all characters after a semicolon. If you are using PSPad editor and you have customized the editor according to the instructions on first pages of this document, the editor recognizes comments, directives, labels and instructions automatically and uses different highlighter colors for different datatypes.

Directives

The directives are not translated directly into LP8501. Instead, directives are instructions for the MEASM.exe compiler. Directives are used to adjust the location of the engine 1, 2, and 3 programs in memory and reserve memory resources in the LP8501 SRAM. For example **.segment** program1 is a directive which tells to the compiler that whatever follows is the program for the program execution engine 1. An overview of the directives is given in the following table.

Directive	Description	Example source code
.segment	Adjust the location of the programs in SRAM. Note the leading dot	**.segment** program1 **.segment** flashing_light
ds	**D**efine **S**torage; The directive reserves memory resources in the SRAM. The ds directive takes one parameter, which is the number of words to reserve. The number of bits in a word (word length) is 16. The allocated words are initialized with zeros	**ds** 3 **ds** 17
dw	**D**efine constant **W**ord. Inserts a binary word to the SRAM.	**dw** 0000000011111111b **dw** FFABh **dw** 3

Labels

A label is a symbolic address. Labels are used to mark program line(s), like in branch instruction and labeling mapping table rows. Labels must have the colon (:) suffix. In the Figure 12 code *loop1:* is a label which indicates the starting address of the loop, *row1:* is a label which indicates mapping table row1.

Instructions

Instructions are executable statements. MEASM-compiler translates text-based language source instructions into hex- and binary-based executable codes. In the example code presented in Figure 12 for example **set_pwm** is an instruction. Almost all the instructions take parameters, which may be constants (like FFh). All instructions are explained in a list below. Note that with the mux instructions engines will not push new PWM value to the output pointed with this instruction until set_pwm or ramp instruction is executed. Current PWM value will remain until new one is updated.

- **branch**
  - Branch instruction is part of a loop structure. Loop starting point is marked with label ending with the colon suffix. Branch instruction has two parameters: number of times loop is executed (loopcount) and the starting point of the loop (address). Instruction in text editor: **branch**

parameter(loopcount), parameter(address). For example if loop is labeled as *loop1:* and loop is executed 5 times, the branch instruction would look like *branch 4, loop1*. An endless loop would be defined with 0.

- **end**
  - End instructions ends the program execution. All programs should have end instruction. End instruction can have two parameters i (meaning interrupt) and r (meaning reset). Parameter i sends interruption from the program. Parameter r resets the program counters.
- **int**
  - Int instruction sends interruption from the program. Can be read from the register 3Ah, which engine has sent the interruption. Also interruption can be seen from the evaluation program manual tab Status/Interrupt section. This instruction does not stop the engine execution.
- **ramp**
  - Ramp instruction generates a PWM ramp for the mapped output(s). Ramp instruction has two parameters: *ramp time* and *PWM step value*. In machine code the ramp time consists in PWM step value (steps taken from the current PWM value to wanted PWM value), step time and prescale. When using text editor and compiler, user needs only to set the *PWM step value* and *ramp time*, compiler will calculate suitable values for step time and prescale needed by LP8501. Compiler can round the time if needed. Instruction in text editor: **ramp** parameter(ramp_time), parameter(PWM _step_value). For example *ramp 0.5, 255* ramps the mapped output to full PWM value in 0.5 seconds. Note that when maximum PWM value is reached before the whole ramp is executed the rest of the ramp time will saturate to wait. For example if starting PWM value is 127 and we have an instruction *ramp 2, 255*, the maximum PWM value is reached in 128 steps and the rest of the steps are wait. PWM step value can vary from 0 to 255 and can have minus sign prefix to indicate falling ramp. Maximum ramp time is 127 seconds.
  - Note that if the step time is set to 0, **ramp** instruction will be considered as a **set_PWM** instruction.
- **rst**
  - This instruction resets Program Counter register and continues executing the program from the program start address defined in registers 4Ch — 4Eh. In LP8501 datasheet this is the *Go to Start* instruction.
- **set_PWM**
  - Set_PWM instruction sets the PWM value of the mapped output(s). Instruction has one parameter which is the output(s) wanted PWM value. Instruction in text editor: **set_pwm** parameter(PWM value). For example *set_pwm 255* sets the mapped output(s) PWM to full-scale. Parameter can be in decimal or hexadecimal. Range is from 0 to 255 or 0 to FF.
- **mux_clr**
  - Mux_clr instruction clears the mapping of output(s) to engines.
- **mux_dec**
  - Mux_dec instruction sets the previous row active in the mapping table.
- **mux_inc**
  - Mux_inc instruction sets the next row active in the mapping table.
- **mux_ld_end**
  - Mux_ld_end instruction defines the end address of the mapping table. Instruction has one parameter which is the end address of the mapping table. Instruction in text editor: **mux_ld_end** parameter(address). Here it is supposed that the address is labeled previously in the code. For example: *row_end: dw 0000000100000000b* (this defines that to the address labeled row_end is mapped output 9).*mux_ld_end row_end* defines that the mapping table ends to the address labeled as row_end. Note that if one or more outputs are mapped to multiple engines engine 1 is strongest and takes control. In case where one or more outputs are mapped into engines 2 and 3, engine 2 is stronger and takes the control of the outputs.
- **mux_ld_start**
  - Mux_ld_start instruction defines the start address of the mapping table. Instruction has one parameter which is the start address of the mapping table. Instruction in text editor: **mux_ld_start** parameter(address). Here it is supposed that the address is labeled previously in the code. For

example: *row_start: dw 0000000000000001b* (this defines that to the address labeled row_start is mapped output 1). *mux_ld_start row_start* defines that the mapping table starts from the address labeled as row_start Note that if one or more outputs are mapped to multiple engines engine 1 is strongest and takes control. In case where one or more outputs are mapped into engines 2 and 3, engine 2 is stronger and takes the control of the outputs.

- **mux_sel**
  – This instruction maps one and only one output to an engine. Instruction has one parameter which is the output being mapped. Instruction in text editor: **mux_sel** parameter(output). For example *mux_sel 2* maps output 2 to an engine.
- **mux_set**
  – This instruction sets index pointer to point to the mapping table row. Instruction has one parameter which is the address defined in mapping table. Instruction in text editor: **mux_set** parameter(address). Here it is supposed that the address is labeled previously in the code. For example *mux_set row_middle* defines that the pointer is now pointing to the address labeled row_middle in the mapping table.
- **trigger**
  – Trigger instruction is used between engines and with external trigger to start the engines at wanted point and to synchronize operation between them. All engines can send to and wait trigger from other engines and from external trigger. Instruction in text editor for sending a trigger is **trigger** s{parameter}. For example *trigger s{2}*, where engine sends a trigger to engine two, supposing the sending engine is different than engine two. Parameter can be from 1–3 and e, meaning the external trigger. Parameter can consist of one value or many separated by point. Similarly instruction for waiting a trigger in text editor is **trigger** w{parameter}. For example *trigger w{e}* , where engine waits for external trigger.
- **wait**
  – Wait instruction sets the program to pause for defined time. Parameter for this instruction is time. Instruction in text editor: **wait** parameter(time). For example *wait 0.48* sets the program wait for 0.48 seconds. Wait time parameter can be set from 0 to 0.484 seconds. This parameter can be rounded by the compiler.

Producing an Executable File

Once the text-based source file is typed in and saved using the text editor (PSPad), the source code window should look like the one in Figure 12. To call the compiler routine, select *File >> Compile*. The PSPad LOG window shows the progress of compiling. If the compiler generates error messages, LOG window is necessary for locating these errors.

A listing file, a hex file and a binary file is produced by the MEASM.exe compiler. The name of the files are the same name that you have given to the source code file, with the *.lst, *.hex or *.bin extension. *.hex and *.bin files contain the machine code.

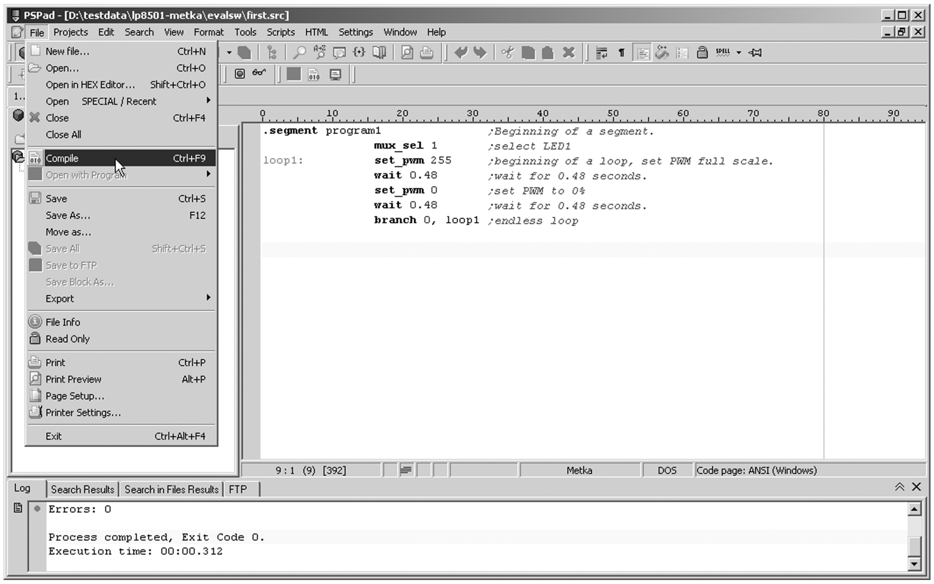

Figure 12. Example of Compiling Source Code

Loading a program to the LP8501'S SRAM

To upload code into the LP8501 SRAM start the LP8501.exe evaluation software. Select the Program tab (see Figure 7) The Program tab is divided into two parts: the right contains the program's source code and the compiled version of the code; the left part contains program execution engine controls. Load the generated *.hex file into the evaluation software view: Click on Open Source File (Figure 13), browse the file and click on Open. To download the machine code into the LP8501, click on download button Figure 14. Pressing this button sets all the engine modes to *Load*.

Figure 13. Open Source File Button In Program Tab

Figure 14. Download Opened Source File Into LP8501

Running the Program

The program is run by checking the Run from the Master control and clicking on Free run-button. See Figure 15. This way all the engines will start at the same time. If you have less than three engines in use, extra engines must be disabled by checking off the box in each engine section. Like in Figure 16 program has only two engines in use and the engine three is checked off. If this is not done the programs may not work correctly. Engines can also be run from individual engine control (see Figure 16) This is one way of debugging the programs by running the programs in steps. Individual engine control can be used also with multiple engines, but then they wont start at the same time.

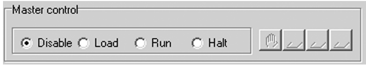

Figure 15. Master Control of the Program Tab

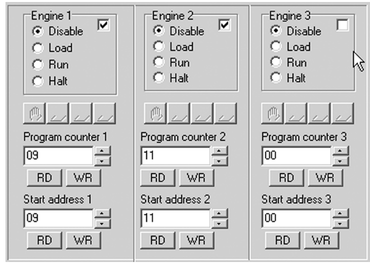

Figure 16. Individual Engine Control

For operating the programs following four modes exist (see Figure 17):

Operation modes

- **Disable** — Engine operation is disabled and they cannot be run.
- **Load** — In this mode writing to program memory is allowed. All the three engines are in hold while one or more engines are in load program mode. PWM values are frozen, also. Program execution continues when all the engines are out of load program mode. Load program mode resets the program counter of the respective engine. Load program mode can be entered from the disabled mode only. Entering load program mode from the run program mode is not allowed. Note that this does not automatically load the program opened with Figure 13. When using this operation mode, one must write the program through the Code memory tab Figure 8.
- **Run** — This mode executes the instructions stored in the program memory. Execution register (ENG1_EXEC etc.) bits define how the program is executed (hold, step, free run or execute once). Program start address can be programmed to Program Counter (PC) register. The Program Counter is reset to zero when the PC's upper limit value is reached.
- **Halt** — In this mode instruction execution aborts immediately and engine operation halts. Execution can be continued if operation mode is set to Run again.

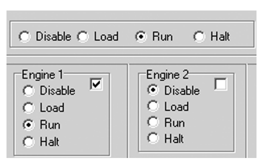

Figure 17. Operation Modes

For executing the programs following four modes exist (see Figure 18):

Executions modes

- **Stop** — Engine execution is stopped. The current instruction is executed and then execution stopped.
- **Step** — Execute the instruction at the location pointed by the Program Counter, increment the Program Counter by one and then reset ENG1_EXEC bits to 00 (enter Stop-mode).
- **Execute Command** — Execute the instruction pointed by the current Program Counter value and reset ENG1_EXEC to 00 (i.e. enter Stop-mode). The difference between Step and Execute Command is that Execute Command does not increment the Program Counter.
- **Free Run** — Start program execution from the instruction pointed by the Program Counter.

Figure 18. Execution Modes

Program can be loaded also from the *Code memory* tab (see Figure 8). In this case one must set the operation mode to *Load*. In *Code memory* tab one must first select the address where data is written. Once the address is active Data can be written in hexadecimal to the Data entry field and push the Update button. Once all the data is updated to the wanted addresses, it can be written to the SRAM memory by pushing the Write Page, which writes two lines in code memory table (first two lines refer to page 0, next two lines to page 1, etc. Pages can be changed with the page selector) or by pushing Write All, which updates the whole memory. Once the data is written to SRAM, operation mode can be set to *Run* and Execution mode for example to *Free Run*. Note that Program Counter(s) must be set accordingly. This is not done automatically like loading program by clicking the Download-button. Note also that the program code does not show up in the program view (in *Program* tab).

Debugging Considerations

There are a few ways to see how the compiler translates the instructions to machine code. Listing file (*.lst) may be used for locating assembling errors. The listing file contains the source code along with the compiled machine code. You can examine the files in any text editor. This is helpful for debugging and seeing how source code is translated into machine code. In Figure 19 is shown an example of listing file. The first column is the row number, second column indicates the SRAM memory address, third shows the machine code data and fourth column includes the source code. Note that the .segment directives show the start address of the program i.e. where to the Program Counters should point.

From the produced two hex-files user can see the pure machine code represented in hexadecimal in two different ways. In *.he2 —file representation of data is 0xYY (YY being the changing data information). In *.hex file the data is represented YY, where YY is the changing data information. In *.he2 file the 8–bit long data elements are separated with comma whereas in *.hex file they are separated with tab. In both files data is represented like in *Code memory* tab memory table. First two columns correspond to first column in *Code memory* tab memory table, third and fourth columns correspond to second column etc. For example if the user would have mux_inc instruction (9D80), in he2 it would be 0x9D, 0x80 and in hex file it would be 9D [tab] 80. In hex-file the start addresses of the programs are at the bottom whereas in *.he2 file they are on the first row engine 1 start address being first, engine 2 second and engine 3 start address third. Also in the bin-file user can see the pure machine code represented in binary. First three rows represent the start addresses of the programs. After the start addresses the program code follows.

Programs can be debugged in the evaluation software *Program* tab by running the program in steps using *Step* or *Execute command* execution modes. Also one way to see what is written to the LP8501 is to look at the evaluation program *History* tab (see Figure 9).

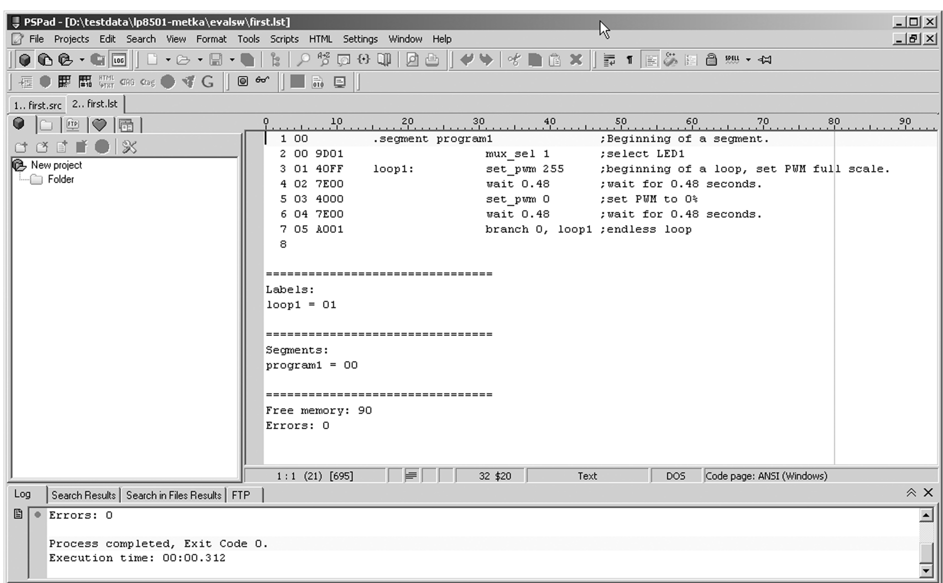

Figure 19. Listing File of the Example Program First

8 Schematic and Layout

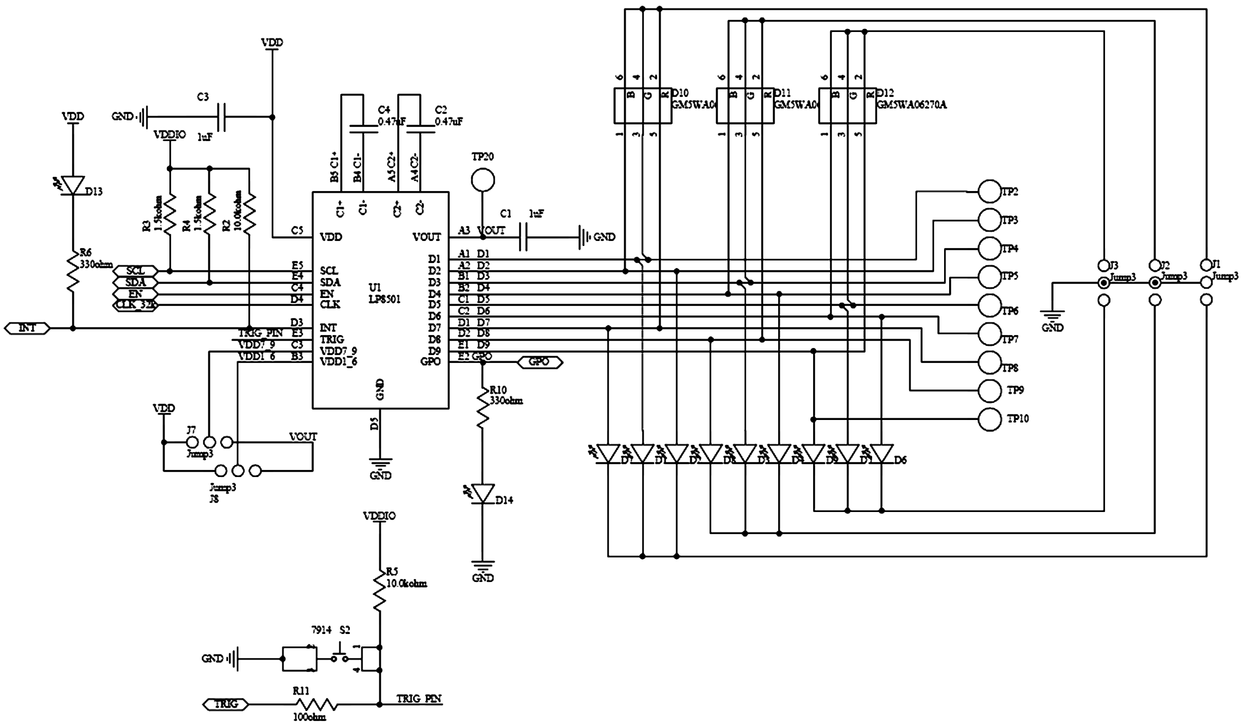

Figure 20. Schematics of the Evaluation Board Application Side

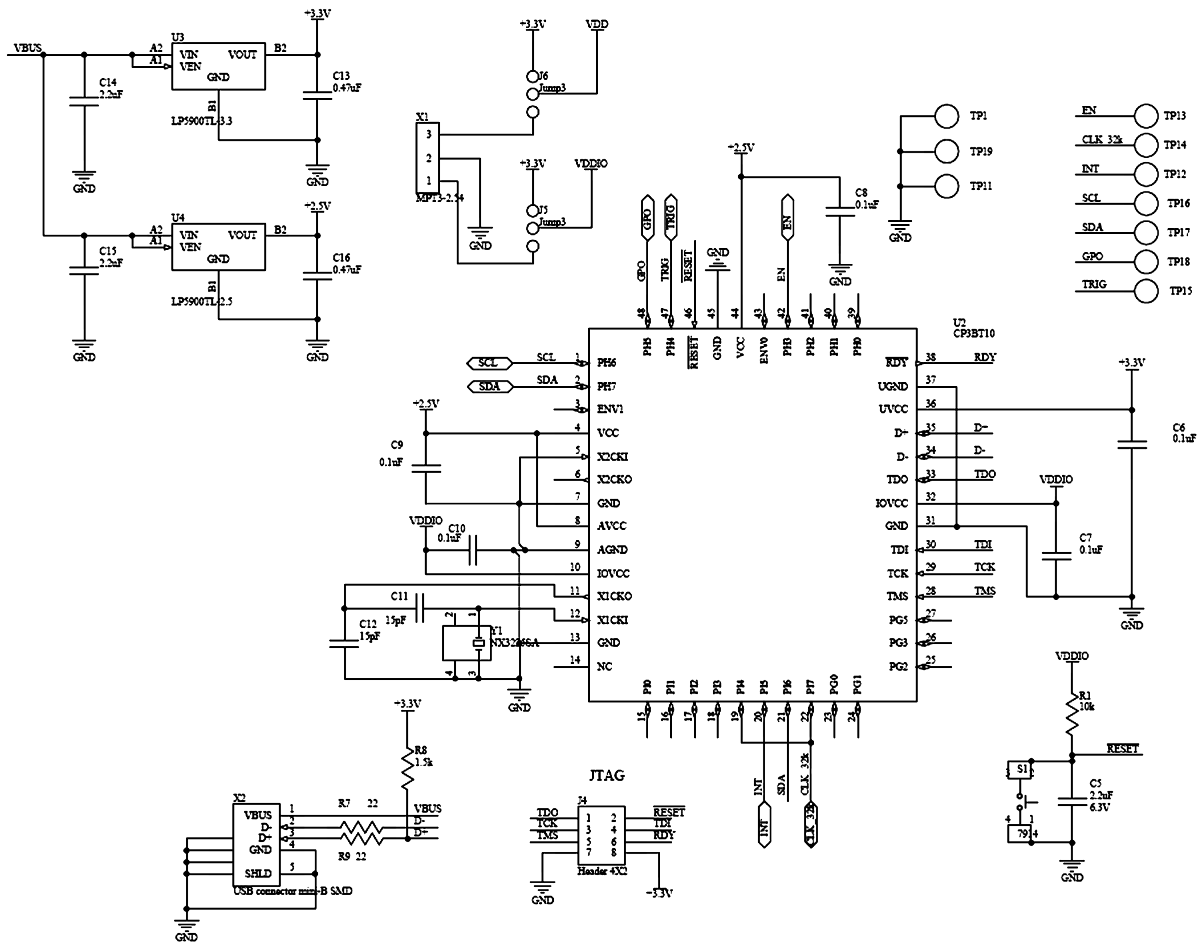

Figure 21. Schematics of the Evaluation Board USB Side

9 Bill of Materials

Designator	Quantity		Comment	Description	Footprint
C1, C3	2	1.0uF 10% 10V 0402 X5R	TDK C1005X5R1A105K	Capacitor	0402
C2, C4	2	470nF 10% 10V 0402 X5R	TDK C1005X5R1A474K	Capacitor	0402
C5, C14, C15	3	2.2uF 10V 0603	LMK107BJ225KA-T. TAIYO YUDEN	Capacitor	0603L
C11, C12	2	15pF 5% 50V 0603	C1608C0G1H150J	Capacitor	0603L
C6, C7, C8, C9, C10	5	100nF 10% 16V 0603	C1608X7R1C104K	Capacitor	0603
C13, C16	2	470nF 10% 10V 0603	C1608X5R1A474K	Capacitor	0603
D1, D4, D7	3	LW M67C	Osram LW M67C	White LED	LW M67C WLED
D10, D11, D12	3	GM5WA06270A	Sharp GM5WA06270A	RGB LED	GM5WA
D13, D14	2	LSL296-P2Q2-1-Z	Osram LS L296	Red LED	0603
J1, J2, J3, J5, J6, J7, J8	7	Pinheader 3x1, 2.54 mm pitch	AMP 5-146280-3	Jumper3	J3
R1, R2, R5	3	10k 5% 0603	Resistor, thick film	Resistor	0603L
R7, R9	2	22R 1% 0603	Resistor, thick film	Resistor	0603L
R3, R4, R8	3	1k5 1% 0603	Resistor, thick film	Resistor	0603
R6, R10	2	330R 5% 0603	Resistor, thick film	Resistor	0603
R11	1	100R 5% 0603	Resistor, thick film	Resistor	0603
S1, S2	2	PUSHBUTTON-SPNO-SMD	BOURNS 7914J-001-000	Push button	7914J
U1	1	LP8501	TI LP8501	Multi-Purpose 9-Output LED Driver	YFQ0025
U2	1	CP3BT10	TI CP3BT10	Connectivity Processor with USB Interface	NPB0048B
U3	1	LP5900TL-3.3	TI LP5900TL-3.3	Ultra Low Noise, 100mA Linear Regulator for RF/Analog Circuits Requires No Bypass Capacitor	YZR0004BCA
U4	1	LP5900TL-2.5	TI LP5900TL-2.5	Ultra Low Noise, 100mA Linear Regulator for RF/Analog Circuits Requires No Bypass Capacitor	YZR0004BCA
X1	1	MPT3-2.54	Phoenix Contact 1725669	Terminal block, 3x1 2.54mm pitch	MPT3-2.54
X2	1	USB mini B-type, SMD	Molex 675031020	USB 2.0, Right Angle, SMT, A Type, Receptacle, 5 Position, Black	440478
Y1	1	Crystal 12MHz, NX3225SA	NDK NX3225SA-12.000000MHZ	NDK crystal oscillator	NX3225A

ADDITIONAL TERMS AND CONDITIONS, WARNINGS, RESTRICTIONS, AND DISCLAIMERS FOR EVALUATION MODULES

Texas Instruments Incorporated (TI) markets, sells, and loans all evaluation boards, kits, and/or modules (EVMs) pursuant to, and user expressly acknowledges, represents, and agrees, and takes sole responsibility and risk with respect to, the following:

1. User agrees and acknowledges that EVMs are intended to be handled and used for feasibility evaluation only in laboratory and/or development environments. Notwithstanding the foregoing, in certain instances, TI makes certain EVMs available to users that do not handle and use EVMs solely for feasibility evaluation only in laboratory and/or development environments, but may use EVMs in a hobbyist environment. All EVMs made available to hobbyist users are FCC certified, as applicable. Hobbyist users acknowledge, agree, and shall comply with all applicable terms, conditions, warnings, and restrictions in this document and are subject to the disclaimer and indemnity provisions included in this document.
2. Unless otherwise indicated, EVMs are not finished products and not intended for consumer use. EVMs are intended solely for use by technically qualified electronics experts who are familiar with the dangers and application risks associated with handling electrical mechanical components, systems, and subsystems.
3. User agrees that EVMs shall not be used as, or incorporated into, all or any part of a finished product.
4. User agrees and acknowledges that certain EVMs may not be designed or manufactured by TI.
5. User must read the user's guide and all other documentation accompanying EVMs, including without limitation any warning or restriction notices, prior to handling and/or using EVMs. Such notices contain important safety information related to, for example, temperatures and voltages. For additional information on TI's environmental and/or safety programs, please visit www.ti.com/esh or contact TI.
6. User assumes all responsibility, obligation, and any corresponding liability for proper and safe handling and use of EVMs.
7. Should any EVM not meet the specifications indicated in the user's guide or other documentation accompanying such EVM, the EVM may be returned to TI within 30 days from the date of delivery for a full refund. THE FOREGOING LIMITED WARRANTY IS THE EXCLUSIVE WARRANTY MADE BY TI TO USER AND IS IN LIEU OF ALL OTHER WARRANTIES, EXPRESSED, IMPLIED, OR STATUTORY, INCLUDING ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE. TI SHALL NOT BE LIABLE TO USER FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RELATED TO THE HANDLING OR USE OF ANY EVM.
8. 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 EVMs might be or are used. TI currently deals with a variety of customers, and therefore TI's 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 with respect to the handling or use of EVMs.
9. User assumes sole responsibility to determine whether EVMs may be subject to any applicable federal, state, or local laws and regulatory requirements (including but not limited to U.S. Food and Drug Administration regulations, if applicable) related to its handling and use of EVMs and, if applicable, compliance in all respects with such laws and regulations.
10. User has sole responsibility to ensure the safety of any activities to be conducted by it and its employees, affiliates, contractors or designees, with respect to handling and using EVMs. Further, user is responsible to ensure that any interfaces (electronic and/or mechanical) between EVMs and any human body are designed with suitable isolation and means to safely limit accessible leakage currents to minimize the risk of electrical shock hazard.
11. User shall employ reasonable safeguards to ensure that user's use of EVMs will not result in any property damage, injury or death, even if EVMs should fail to perform as described or expected.
12. User shall be solely responsible for proper disposal and recycling of EVMs consistent with all applicable federal, state, and local requirements.

Certain Instructions. User shall operate EVMs within TI's recommended specifications and environmental considerations per the user's guide, accompanying documentation, and any other applicable requirements. Exceeding the specified ratings (including but not limited to input and output voltage, current, power, and environmental ranges) for EVMs may cause property damage, personal injury or death. If there are questions concerning these ratings, user should contact a TI field representative prior to connecting interface electronics including input power and intended loads. Any loads applied outside of the specified output range may result in unintended and/or inaccurate operation and/or possible permanent damage to the EVM and/or interface electronics. Please consult the applicable EVM user's guide prior to connecting any load to the EVM output. If there is uncertainty as to the load specification, please contact a TI field representative. During normal operation, some circuit components may have case temperatures greater than 60°C as long as the input and output are maintained at a normal ambient operating temperature. These components include but are not limited to linear regulators, switching transistors, pass transistors, and current sense resistors which can be identified using EVMs' schematics located in the applicable EVM user's guide. When placing measurement probes near EVMs during normal operation, please be aware that EVMs may become very warm. As with all electronic evaluation tools, only qualified personnel knowledgeable in electronic measurement and diagnostics normally found in development environments should use EVMs.

Agreement to Defend, Indemnify and Hold Harmless. User agrees to defend, indemnify, and hold TI, its directors, officers, employees, agents, representatives, affiliates, licensors and their representatives harmless from and against any and all claims, damages, losses, expenses, costs and liabilities (collectively, "Claims") arising out of, or in connection with, any handling and/or use of EVMs. User's indemnity shall apply whether Claims arise under law of tort or contract or any other legal theory, and even if EVMs fail to perform as described or expected.

Safety-Critical or Life-Critical Applications. If user intends to use EVMs in evaluations of safety critical applications (such as life support), and a failure of a TI product considered for purchase by user for use in user's product would reasonably be expected to cause severe personal injury or death such as devices which are classified as FDA Class III or similar classification, then user must specifically notify TI of such intent and enter into a separate Assurance and Indemnity Agreement.

RADIO FREQUENCY REGULATORY COMPLIANCE INFORMATION FOR EVALUATION MODULES

Texas Instruments Incorporated (TI) evaluation boards, kits, and/or modules (EVMs) and/or accompanying hardware that is marketed, sold, or loaned to users may or may not be subject to radio frequency regulations in specific countries.

General Statement for EVMs Not Including a Radio

For EVMs not including a radio and not subject to the U.S. Federal Communications Commission (FCC) or Industry Canada (IC) regulations, TI intends EVMs to be used only for engineering development, demonstration, or evaluation purposes. EVMs are not finished products typically fit for general consumer use. EVMs may nonetheless generate, use, or radiate radio frequency energy, but have not been tested for compliance with the limits of computing devices pursuant to part 15 of FCC or the ICES-003 rules. Operation of such EVMs 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.

General Statement for EVMs including a radio

User Power/Frequency Use Obligations: For EVMs including a radio, the radio included in such EVMs is intended for development and/or professional use only in legally allocated frequency and power limits. Any use of radio frequencies and/or power availability in such EVMs and their development application(s) must comply with local laws governing radio spectrum allocation and power limits for such EVMs. It is the user's sole responsibility to only operate this radio in legally acceptable frequency space and within legally mandated power limitations. Any exceptions to this are strictly prohibited and unauthorized by TI unless user has obtained appropriate experimental and/or development licenses from local regulatory authorities, which is the sole responsibility of the user, including its acceptable authorization.

U.S. Federal Communications Commission Compliance

For EVMs Annotated as FCC – FEDERAL COMMUNICATIONS COMMISSION Part 15 Compliant

Caution

This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation. Changes or modifications could void the user's authority to operate the equipment.

FCC Interference Statement for Class A EVM devices

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at its own expense.

FCC Interference Statement for Class B EVM devices

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

Industry Canada Compliance (English)

For EVMs Annotated as IC – INDUSTRY CANADA Compliant:

This Class A or B digital apparatus complies with Canadian ICES-003.

Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

Concerning EVMs Including Radio Transmitters

This device complies with Industry Canada licence-exempt RSS standard(s). Operation is subject to the following two conditions: (1) this device may not cause interference, and (2) this device must accept any interference, including interference that may cause undesired operation of the device.

Concerning EVMs Including Detachable Antennas

Under Industry Canada regulations, this radio transmitter may only operate using an antenna of a type and maximum (or lesser) gain approved for the transmitter by Industry Canada. To reduce potential radio interference to other users, the antenna type and its gain should be so chosen that the equivalent isotropically radiated power (e.i.r.p.) is not more than that necessary for successful communication.

This radio transmitter has been approved by Industry Canada to operate with the antenna types listed in the user guide with the maximum permissible gain and required antenna impedance for each antenna type indicated. Antenna types not included in this list, having a gain greater than the maximum gain indicated for that type, are strictly prohibited for use with this device.

Canada Industry Canada Compliance (French)

Cet appareil numérique de la classe A ou B est conforme à la norme NMB-003 du Canada

Les changements ou les modifications pas expressément approuvés par la partie responsable de la conformité ont pu vider l'autorité de l'utilisateur pour actionner l'équipement.

Concernant les EVMs avec appareils radio

Le présent appareil est conforme aux CNR d'Industrie Canada applicables aux appareils radio exempts de licence. L'exploitation est autorisée aux deux conditions suivantes : (1) l'appareil ne doit pas produire de brouillage, et (2) l'utilisateur de l'appareil doit accepter tout brouillage radioélectrique subi, même si le brouillage est susceptible d'en compromettre le fonctionnement.

Concernant les EVMs avec antennes détachables

Conformément à la réglementation d'Industrie Canada, le présent émetteur radio peut fonctionner avec une antenne d'un type et d'un gain maximal (ou inférieur) approuvé pour l'émetteur par Industrie Canada. Dans le but de réduire les risques de brouillage radioélectrique à l'intention des autres utilisateurs, il faut choisir le type d'antenne et son gain de sorte que la puissance isotrope rayonnée équivalente (p.i.r.e.) ne dépasse pas l'intensité nécessaire à l'établissement d'une communication satisfaisante.

Le présent émetteur radio a été approuvé par Industrie Canada pour fonctionner avec les types d'antenne énumérés dans le manuel d'usage et ayant un gain admissible maximal et l'impédance requise pour chaque type d'antenne. Les types d'antenne non inclus dans cette liste, ou dont le gain est supérieur au gain maximal indiqué, sont strictement interdits pour l'exploitation de l'émetteur.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265

Important Notice for Users of EVMs Considered “Radio Frequency Products” in Japan

EVMs entering Japan are NOT certified by TI as conforming to Technical Regulations of Radio Law of Japan.

If user uses EVMs in Japan, user is required by Radio Law of Japan to follow the instructions below with respect to EVMs:

1. Use EVMs in a shielded room or any other test facility as defined in the notification #173 issued by Ministry of Internal Affairs and Communications on March 28, 2006, based on Sub-section 1.1 of Article 6 of the Ministry’s Rule for Enforcement of Radio Law of Japan,
2. Use EVMs only after user obtains the license of Test Radio Station as provided in Radio Law of Japan with respect to EVMs, or
3. Use of EVMs only after user obtains the Technical Regulations Conformity Certification as provided in Radio Law of Japan with respect to EVMs. Also, do not transfer EVMs, unless user gives the same notice above to the transferee. Please note that if user does not follow the instructions above, user will be subject to penalties of Radio Law of Japan.

http://www.tij.co.jp

【無線電波を送信する製品の開発キットをお使いになる際の注意事項】 本開発キットは技術基準適合証明を受けておりません。 本製品のご使用に際しては、電波法遵守のため、以下のいずれかの措置を取っていただく必要がありますのでご注意ください。

1. 電波法施行規則第6条第1項第1号に基づく平成18年3月28日総務省告示第173号で定められた電波暗室等の試験設備でご使用いただく。
2. 実験局の免許を取得後ご使用いただく。
3. 技術基準適合証明を取得後ご使用いただく。。

なお、本製品は、上記の「ご使用にあたっての注意」を譲渡先、移転先に通知しない限り、譲渡、移転できないものとします

上記を遵守頂けない場合は、電波法の罰則が適用される可能性があることをご留意ください。

日本テキサス・インスツルメンツ株式会社

東京都新宿区西新宿６丁目２４番１号

西新宿三井ビル

http://www.tij.co.jp

Texas Instruments Japan Limited

(address) 24-1, Nishi-Shinjuku 6 chome, Shinjuku-ku, Tokyo, Japan

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as “components”) are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI’s terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers’ products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers’ products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI’s goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or “enhanced plastic” are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have ***not*** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265