Log & Level: User Manual

Author: Tony Abboud
Instructor: Bill Hawkins
University of Maryland, Fall 2013

Table of Contents

Abstract

This document describes the functionality and use of Log & Level.

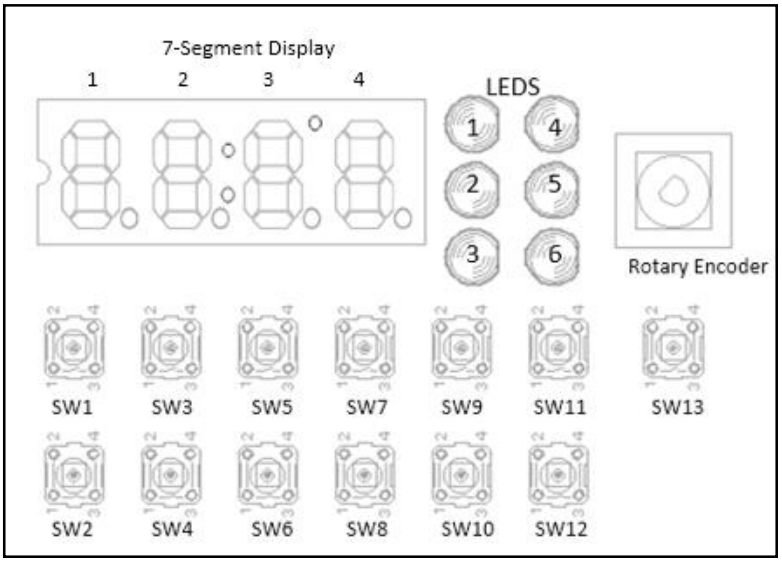
Target Audience

This user manual is tailored for an audience with a technical background who might use this software/hardware package in the field or lab setting.

Project Synopsis

This software/hardware project implements a logging voltage meter, using the onboard analog-to-digital converter (ADC), direct memory access (DMA), and universal asynchronous receiver/transmitter (USART), as well as an accelerometer interface, using the serial peripheral interface bus (SPI). In this software, the logging voltmeter and the accelerometer interface are treated as tasks or “apps”, which is controlled by a very simple operating system task scheduler. The interface for the software is through serial communication (USART), which allows the user to select settings, and control which tasks are run. While an app is running, the user has the ability to pause, continue, restart or switch tasks at any time using the P24v04 expansion board. The simplified OS is capable of switching tasks and restoring the correct state of the task upon return. The logging voltmeter utilizes the STM32F4 ADC, DMA, USART, and GPIO peripherals. The ADC performs the analog conversion, while the DMA transfers the converted values to a place in memory. The USART allows the voltage to be logged on the serial terminal in real time. The user is able to enter various settings such as a sample delay time, over/under voltage, etc. The accelerometer interface utilizes the STM32F4 SPI, USART, GPIO peripherals. The SPI provides communications between the accelerometer chip, LIS302DL, and the stm32f4 board. The display for the accelerometer app is solely on the P24v04 expansion board, and does not log any values to the serial terminal. More detailed descriptions of each peripheral app will be explained in further sections.

1. Board Layout



The P24 Expansion Board, external interface, contains 4 seven-segment displays, 6 red/green LEDs, 13 push-button switches, and 1 rotary encoder.

2. Hardware / Software Requirements

1. STM32F4-Discovery Evaluation Board
2. P24v04 Expansion Board
3. STM32 ST-LINK Utility or Keil uVision 4.xx
4. Serial Terminal (PuTTY, RealTerm, etc.)
5. USB mini to standard cable
6. (2) Jumper wires (one side jumper, other side exposed wire)
   *Connect to PC0 and GND of stm32f4 board

3. Build Instructions

3.1. Compile / Link Project

1. Navigate to the main project directory (i.e. Abboud_ENEE440_Project/)
2. Double click on clean.bat file to remove any previous execution files. Press "Enter" when prompted
3. Double click on make.bat file to compile and link all of the header and source files. Again, press "Enter" when prompted

3.2. Program the STM32F4 board (using ST-LINK Utility)

1. Connect the USB mini to the STLINK/V2 (See Fig. 1) USB connection and the other side to the PC

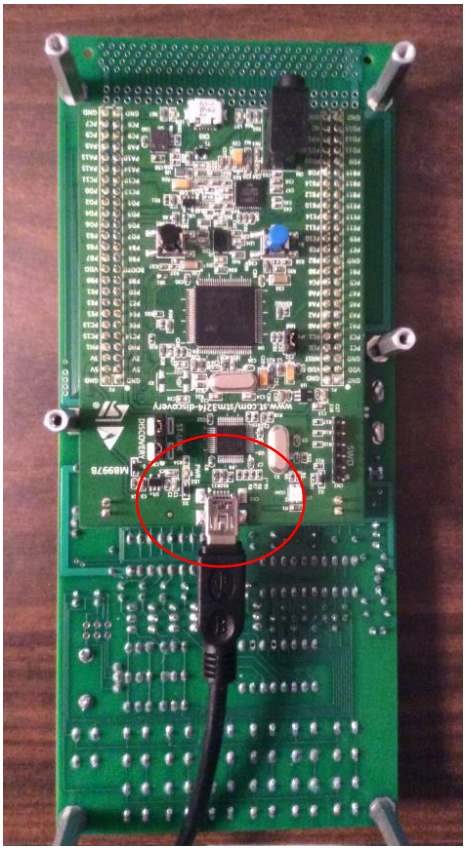


Figure 1: ST-LINK/V2

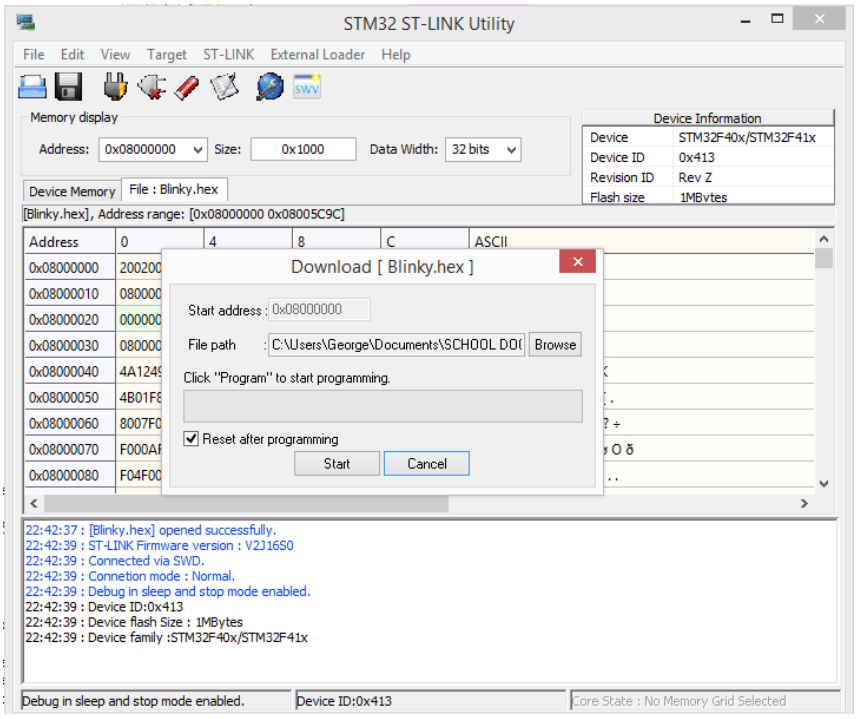


Figure 2: Programming Blinky.hex onto STM32F4

2. Open up STM32 ST-LINK Utility
3. Select File/Open_File, then navigate to the blinky.hex file created from the make.bat
4. Double click on the .hex file to open
5. Once the file is loaded into the ST-Link Utility, press Target/Program_&_Verify/Start (see Fig. 2 above)
6. The program will then be loaded onto the STM32F4 board.
7. The board is now programmed

3.3. Run the Software (using PuTTY)

1. Disconnect the USB mini and connect it to the JNEX connector on the left side of the expansion board (see Fig 3)
2. Open up the PuTTY terminal and navigate to the serial connection

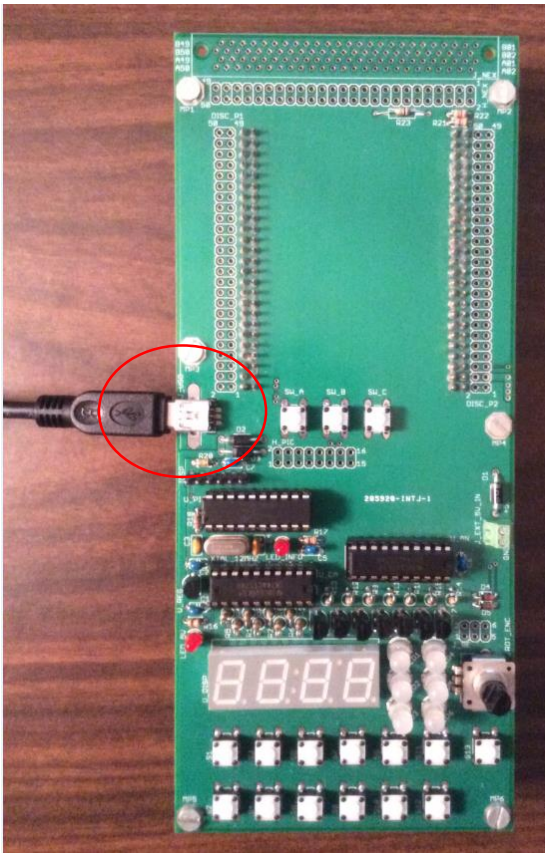


Figure 3: USB connected to JNEX on Expansion Board

3. On the PuTTY configuration choose "Serial" from the left side category
4. Change the "serial line to connect to" to the COM port that the JNEX has connected to
   **Look in the Device Manager to see which COM port is connected
5. Change the serial configurations to 96008N1 (see Fig 3)

Speed (baud) : 9600
Data bits : 8
Parity bits : None
Stop Bits : 1

6. Press "Open"
7. Once the serial terminal comes up, press the reset button the board
8. You will then be prompted with the display message
9. You are now running the program live (see Fig. 5)
10. To quit the program simply disconnect the USB

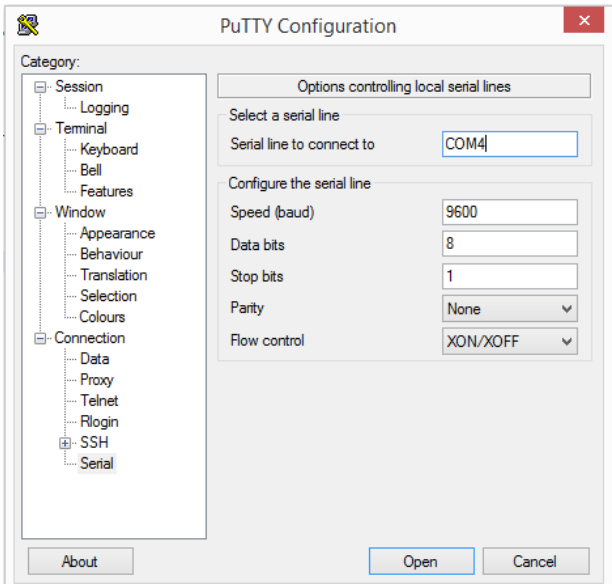


Figure 4: PuTTY Serial Configurations

Figure 5: Project Welcome Screen

4. Running the "Apps"

The following sections, 1 and 2, assume that the user enters these apps directly from the main menu. The process of switching apps will be discussed in section 4.3.

4.1. Logging Voltage Meter

Connect one jumper wire to GPIO port **PC0** (Analog Input) and connect the other wire to the GPIO port **GND**. These two wires will be the two leads to measure the voltages. To begin the logging voltage meter app, you select option A (press 'A', then hit Enter) from the main menu. You are then prompted to enter several setting selections (ADC Sample Rate and Sample Delay).

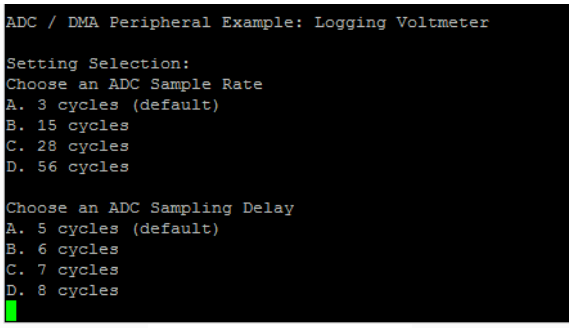


Figure 6: Setting Selection

The next settings are for the over/under voltage. To enter the various voltages, you use the P24v04 expansion board switches 1 through 8 and the rotary encoder to get more precision. At the beginning there will be all zeroes on the 7 segment display. With the precision set to 0 (default, no decimal point), you can only adjust switches 7 and 8 which signifies the ones digit. If you move the rotary encoder to the left one click, the decimal will shift in and you will get more precision on the value that you enter. For example, instead of only 3V over-voltage, you can set

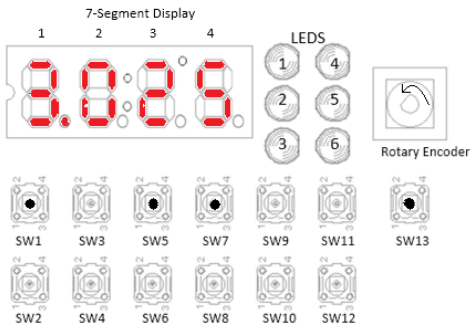


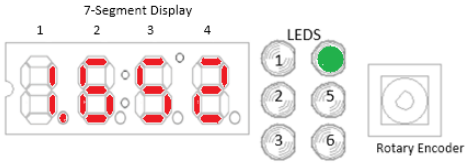
Figure 7: Over/Under Voltage

the over voltage to 3.025V, as seen above. The over/under voltage can only go up to 9.999V and down to 0.000V. Finally, to set the voltage you press switch 13.

```
Configuring GPIO...
Configuring ADC...
Configuring DMA...

Press sw 13 to run/stop logging

Voltage: 0.000V

Voltage: 3.251V

Voltage: 1.650V

Voltage: 1.652V

Voltage: 1.651V

Voltage: 1.653V

Voltage: 1.652V
```

Figure 8: Voltage Logging for AAA battery

The peripherals are then configured using the software functions (described in the maintainer document). To begin the logging of the voltage press switch 13. The voltage will then be logged on the screen via USART/DMA and also on the 7-segment display. LED4 will be blinking green when the voltage is logging. When the jumper wires are not measuring anything, the voltage reading is around 3.3V which is the operating voltage of the GPIO pins. Once the wires form a closed circuit, the measured voltage will appear. For example, to measure the voltage of a AAA battery, touch the wire lead of PC0 to the '+' side of the battery, and the GND lead to the '-' side of the battery. The voltage that is logged should be around 1.6V, as seen on the right.



If the measured voltage is above or below the over/under voltage, the chip will shut off, the 7 segment display will stop logging and show "----", and the terminal will prompt the user to press switch 12 to continue. Also, LEDs 2 and 5 will be lit red.

```
Voltage: 3.251V
Voltage has exceeded the over/under limit!!
The chip has turned off.

Press SW 12 to turn back on and continue.
```

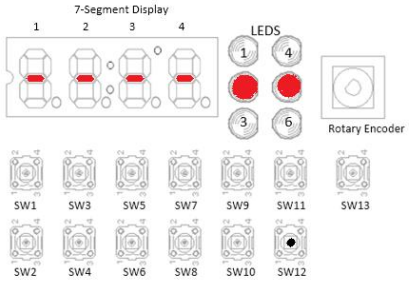


Figure 9: Over/Under voltage chip shutoff

If you want to pause, reset, or switch apps, while the voltage is logging, you can press switch 13 and you will be prompted with a menu. "Continue" will go right back to the voltage logging where you left off. "Reset" will take you back to the beginning of the voltage meter setting selection. "Change" will stop the voltage meter app, and transfer execution to the accelerometer app (more details in section 4.3).

```
To continue, select an option
A. Change to ACCEL/SPI Peripheral
B. Continue using the ADC/DMA
C. Restart the ADC/DMA App
```

Figure 10: ADC pause menu

4.2. Accelerometer Interface

Ensure that the board is in a static position. To begin the accelerometer interface, select B from the main menu. You will be prompted to press switch 13 to begin. After pressing switch 13, the configurations for the GPIO, MEMS (Accelerometer), and SPI will take place. The “app” will then start up. On the expansion board, you will see 3 zeroes on the 7-segment display, and all of the LEDs will be green, as seen below.

```
Accelerometer / SPI Peripheral Example

To begin press sw13
Configuring GPIO...
Configuring MEMS...
Configuring SPI...

To pause or switch peripherals, press sw 13...
```

Figure 11: Accelerometer Main Menu

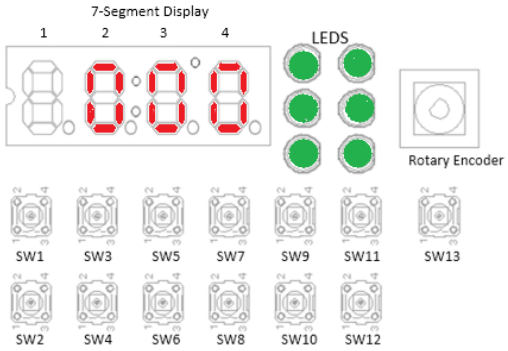


Figure 12: Accelerometer initial/reset state

To use the accelerometer, you tilt the board on the two axis show below.

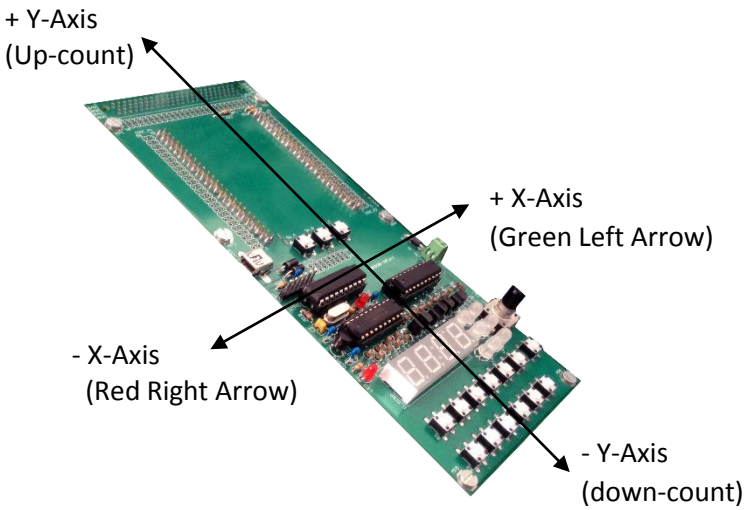


Figure 13: Accelerometer Axis

When the board is tilted forward, the values on the seven segment display will start to count up. When the board is tilted backwards, the values on the seven segment display will count down. The max and min values for the display are 999 and -999 respectively.

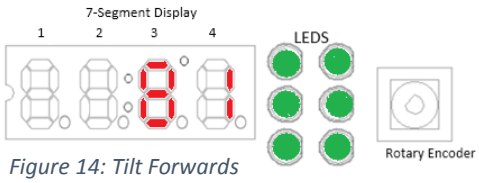


Figure 14: Tilt Forwards

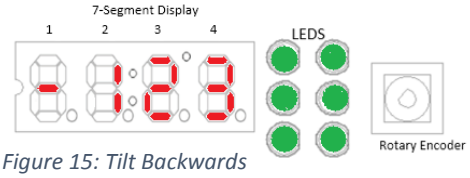


Figure 15: Tilt Backwards

When the board is tilted to the right, the LEDs will display a green arrow pointing to the left. The arrow represents the direction to turn the board to get it back to the original position on that axis. When the board is tilted to the left, the LEDs will display a red arrow pointing to the right. The board can be tilted in a combination of ways, to allow up/down counting along with the led arrows. For example, figure 18 shows the board tilted forward (counting up) while tilted to the right (green left arrow).

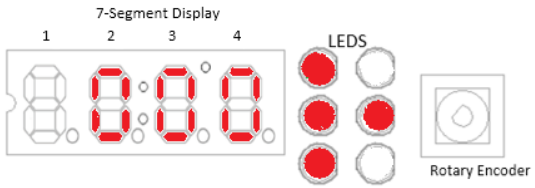


Figure 16: Tilt Left

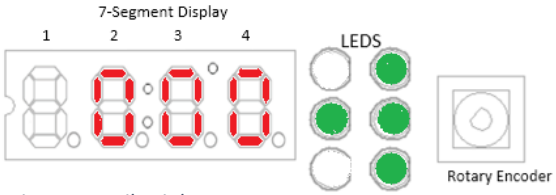


Figure 17: Tilt Right

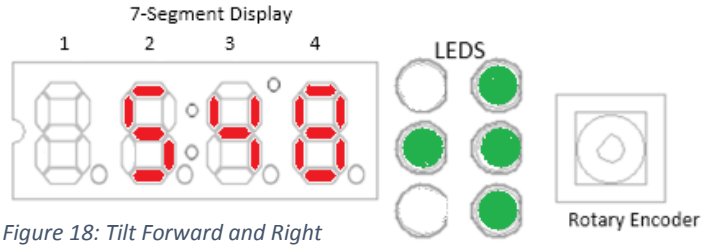


Figure 18: Tilt Forward and Right

If you want to pause, reset, or switch apps, while the accelerometer app is running, you can press switch 13 and you will be prompted with a menu. "Continue" will go right back where you left off. "Reset" will take you back to the beginning of the accelerometer app. "Change" will stop the accelerometer and transfer execution to the logging voltage meter (more details in section 4.3). The LEDs will turn off when this menu is shown.

```
To continue, select an option
A. Change to ADC/DMA Peripheral
B. Continue using the Accelerometer
C. Restart the Accel_SPI App
```

Figure 19: Accelerometer pause menu

4.3. Switching Apps

As you are running one app, you can press switch 13 to provide the option to switch to another app, as seen in figures 10 and 19. If you choose this option, the software will save the current state of the app that you are in and transfer you to the requested app. A message "Switching Apps…" will be displayed, which will then bring you to the main menu of the requested app. If you switch back to the original app, then the state before the transition will be loaded, and you will be presented with the message "Continuing the <app_name> app…". When you switch apps, the controls remain the same, i.e. you can still reset, continue or change apps again.

5. Resources

- *Maintainer.pdf*: Detailed description of the software functions that run the embedded system
- *STM32F4 Reference Manual* (*DM00031020.pdf):* Detailed descriptions for each peripheral used and the specific register and bit settings
- *LIS302DL.pdf*: Accelerometer Data Sheet
- *www.ST.com/stm32f4 :* Main website for the stm32f4 discovery board