Lab 2

ECE2534 CRN:12927

Jacob Abel

February 26, 2018

Description The lab required that we implement a basic terminal relay over UART. This requires that the terminal be capable of receiving input and both echoing it back to the transmitter and displaying it on the terminal screen. The terminal had the commands #fN and #bN where N is a number 0 through 7 representing a colour code. These changed the foreground and background of the terminal display. Additionally the top button cleared the display and printed out a status report with the current baud rate, the foreground and background color codes and the number of characters that have been sent to the terminal. With that, the bottom button changed the baud rate from 9600 to 19200 to 38400 to 57600 and back to 9600 with one change on each press. This button needed to be debounced. Finally as a bonus, the LED lights up corresponding to certain characters. White for the # symbol. Red for digits. Blue for characters in the alphabet. Green for everything else.

Finite State Machine The debouncer uses a rudimentary finite state machine to smooth out the button presses by limiting the amount of triggers until some amount of time expired and only after the button is released. Similarly the main parser uses a finite state machine to keep track of which stage of command parsing it is on with state 0 being normal operation, state 1 being a command character being recieved (#), state 2 being a option being recieved(f or b), and state 3 being the final argument being recieved(the color code). If the command was incorrectly structured, the FSM reset to stage 1 and printed any retained characters.

HAL Design The following functions make up my HAL

set_uart_bd(UARTBaudRate_t t): Sets a specific baud rate

cycle_uart_bd(): Cycles the baud rates

InitTimer(): Initialises the timers

void Timer222msStartOneShot(): Starts the timer

int Timer222msExpiredOneShot(): Returns the completion state of the timer

void TimerDebounceStartOneShot(): Starts the timer

int TimerDebounceExpiredOneShot(): Returns the completion state of the

bool BounceFSM(bool b): Debounces the button with output b

void InitButtonS2(): Initialises the button

void InitButtonS1(): Initialises the button

int ButtonS2Pressed(): Returns the button state

int ButtonS1Pressed(): Returns the button state

void init_led(): Configures the LED

void set_led(color_t c): Sets the LED with the color c (black is off)

void trigger_222led(color_t c): Starts the LED and timer

void disable_222led(): Disables the LED

void lcd_clear(): Resets the text stream for the display.

int8_t conv_color(color_t c): Converts a color_t into a GraphicsLib Color

void lcd_set_fg(color_t c): Sets the foreground color

void lcd_set_bg(color_t c): Sets the foreground color

void lcd_out(int8_t c): A basic text stream for the display.