

Austin Blackstone
 EE345M Lab1 PrePrep
 January 23, 2012

Question 1:

a) line 151 of uart_echo.c

b) baud rate:156

Parity: 158

Data Bits:157

Stop Bits:157

c) Port Pins: Port A

Transmit: PA0? (its hardware controlled, not set in software)

Receive: PA1? (its hardware controlled, not set in software)

d) UARTCharGet and UARTCharGetNonBlocking

e) UARTCharPut and UARTCharPutNonBlocking

f) Interrupt vector table in startup_rvmdk.s, it knows to use UARTIntHandler because it is declared in the interrupt vector table (line 111 of the file) this vector table is where you declare the function names for the other interrupt handlers as well.

g) Assembly Code:

```

      78:      UARTIntClear(UART0_BASE, ulStatus); n the receive FIFO. _OSC_MAIN |
0x00000A66 4620      MOV      r0,r4
0x00000A68 F7FFFFFF0 BL.W      UARTIntClear (0x00000A4C)

0x00000A4C 6441      STR      r1,[r0,#0x44]
0x00000A4E 4770      BX        lr

```

R0 is being used to transfer variables between function calls

LR is used as the return PC

R4 is... a temporary variable??? it gets loaded into r0, and used as an offset+#44 to store R1

h) The lm3s8962 is like a PC and has a 16x8 buffer as per pg322 of LM3S892-Datasheet.pdf

Question 2:

a) SSI(Synchronous Serial Interface) on busy wait

b) you give it a string of ascii characters to display (a-z,A-Z) in pcStR and a start point specified in rows from right (ulX) and top (ulY)

c)

PA5 / SSI0 - Oled Data In

PA6 / CCP1 - data / control select

PA7 - power enable switch (connects to pin24 on the OLED)

PA3 - OLED CSn

PA2 / SSI0 - OLED Clock

- d] Pins are shared with
SD Card: PA2, PA5

Question 3: Note: i couldnt find LCD_Blinky, so i found a copy from the interwebs, i hope this is the same

https://engineering.purdue.edu/ece477/Webs/F09-Grp07/nb/Shawn/10_29_notebook/Blinky.pdf

- a] SysTickPeriodSet(...)
- b] you use SysCtlClockSet() to set the clock divider,
for 50Mhz we would provide an argument of SYSCTL_SYSDIV_4
- c] when the counter reaches 0 and wraps around the systick interrupt is called (if enabled)
the COUNT flag is set (bit 16 of the NVIC_ST_CTRL_R) which is cleared when read

Question 4:

- a] The first assembly instruciton is : either a push to save extra registers, or it just starts executing,
the last instruction is either a ldm, Pop, or LDR that loads the LR into the PC, or a BX using LR.
- b] The System pushes R0-R3, R12, Lr, PC, and xPSR onto the stack to save the current processor state
- c] it loads the LR into the PC to restore the PC to it's previous values, and reloads R0-3, R12, and xPSR

See Lab1 Prep file for rest of lab 1 information.