C:\Users\amlweems\Documents\GitHub\EE319k\Lab8\Lab8.uvproj - µVision4

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

SysTickIntHandler

Registers

Register	Value
Core	
R0	0x000001AB
R1	0x00000008
R2	0x40038000
R3	0x00000040
R4	0x0003079B
R5	0x000001D1
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x200003F8
R14 (LR)	0x000006B9
R15 (PC)	0x0000058C
xPSR	0x4100000F
Banked	
System	
Internal	
Mode	Handler
Privilege	Privileged
Stack	MSP
States	6106061
Sec	0.38678468

Project Registers

Logic Analyzer

Setup... Load... Save... | Min Time 0 s | Max Time 0.162828 s | Grid 2 ms | Zoom In Out All | Min/Max Auto Undo | Update Screen Stop Clear | Transition Prev Next | Jump to Code Trace | Signal Info | Show Cycles | Amplitude | Cursor

PORTF&0xF

...0010) >> 4

...0020) >> 5

...0002) >> 1

0, d: -2 2 | 0, d: 0 0 | 0, d: 0 0 | 0, d: 0 0

29.58765 ms | 46.50765 ms, d: -0.003 s ms | 55.58765 ms | 83.58765 ms

Disassembly Logic Analyzer

Lab8.c LCD.c ADCDriver.c LCD2.s Startup.s SysTick.c SysTick.h

```
129   ADC_ACTSS_R |= ADC_ACTSS_ASEN3;          // enable sample sequencer 3
130 }
131 // This function triggers an ADC conversion using sample
132 // sequencer 3, waits for the conversion to finish, and returns
133 // the result in the lower 10 bits of the return value.  It
134 // assumes that the hardware has already been initialized
135 // using ADC_InitSWTriggerSeq3().
136 unsigned long ADC_In(void){
137   unsigned long result;
138   ADC0_PSSI_R = ADC_PSSI_SS3;               // initiate SS3
139   while((ADC0_RIS_R&ADC_RIS_INR3)==0){};      // wait for conversion done
140   result = ADC0_SSFIFO3_R&ADC_SSFIFO3_DATA_M;
141   ADC0_ISC_R = ADC_ISC_IN3;                 // acknowledge completion of curren
142   return result;
143 }
144
145 //debug cohis program periodically samples ADC channel 0 and stores the
146 // result to a global variable that can be accessed with the JTAG
147 // debugger and viewed with the variable watch feature.
148 #define NVIC_EN0_INT19          0x00080000  // Interrupt 19 enable
```

```
1
2 #
3 #
4
5 #
6 #
7
8 e
9 #
10 #
11
12 e
13
14 #endif // __GLOBALS_H__
15
```

curred
the ADC

Analog to Digital Converter (ADC)

SEQ	ASEN	SSP	INR	MASK	IN	OV	UV	EM	MUX0	D0	END0	IE0	TS0
0	0	0	0	0	0	0	0	Controller	2	0	0	0	0
1	0	1	0	0	0	0	0	Controller	0	0	0	0	0
2	0	2	0	0	0	0	0	Controller	0	0	0	0	0
3	1	3	1	0	0	0	0	Controller	2	0	1	1	0

Selected Sequencer 0: ASEN0 SSP0 0 | EM0 Controller | INR0 MASK0 IN0 PSSI0 OV0 UV0

Sample 0 Sequence Settings: Sample: 0 | MUX0 ADC2 | TS0 IE0 END0 D0

FIFO

SSFIFO0:	0x000	SSFSTAT0:	0x0100	HPTR: 00	TPTR: 00	Full	Empty
SSFIFO1:	0x000	SSFSTAT1:	0x0100	HPTR: 00	TPTR: 00	Full	Empty
SSFIFO2:	0x000	SSFSTAT2:	0x0100	HPTR: 00	TPTR: 00	Full	Empty
SSFIFO3:	0x1AB	SSFSTAT3:	0x1000	HPTR: 00	TPTR: 00	Full	Empty

FIFO Contents

	FIFO0
0	<->
1	<->
2	<->
3	<->
4	<->
5	<->
6	<->
7	<->

Sequencer Registers: SSMUX0: 0x00000002 SSMUX1: 0x0000 SSMUX2: 0x0000 SSMUX3: 0x02 | SSCTL0: 0x00000000 SSCTL1: 0x0000 SSCTL2: 0x0000 SSCTL3: 0x06

Global Sequencer Registers: ACTSS: 0x08 EMUX: 0x0000 OSTAT: 0x00 PSSI: 0x00 TMLB: 0x0000 | SSPRI: 0x3210 USTAT: 0x00 SAC: 0x00

Interrupt Con: RIS: 0x08 IM: 0x00 ISC: 0x00

Analog Inputs [V]: ADC0: 0.0000 ADC1: 0.0000 ADC2: 1.2540 ADC3: 0.0000 ADC4: 0.0000 ADC5: 0.0000 ADC6: 0.0000 ADC7: 0.0000

Temp [°C]: 25.00

Vref [V]: 3.0000

Settings: ADC Clock: 16.00 MHz, sampling frequency 125.00 kS/s, sampling time 8.00 us

Command

```
*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 3312 Bytes (10%)

LA (PORTF&0xF & 0xF) >> 0
LA ((PORTF & 0x00000010) >> 4 & 0x10) >> 4
LA ((PORTF & 0x00000020) >> 5 & 0x20) >> 5
LA ((gFlags & 0x00000002) >> 1 & 0x2) >> 1
```

>

ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE DEFINE DIR Display

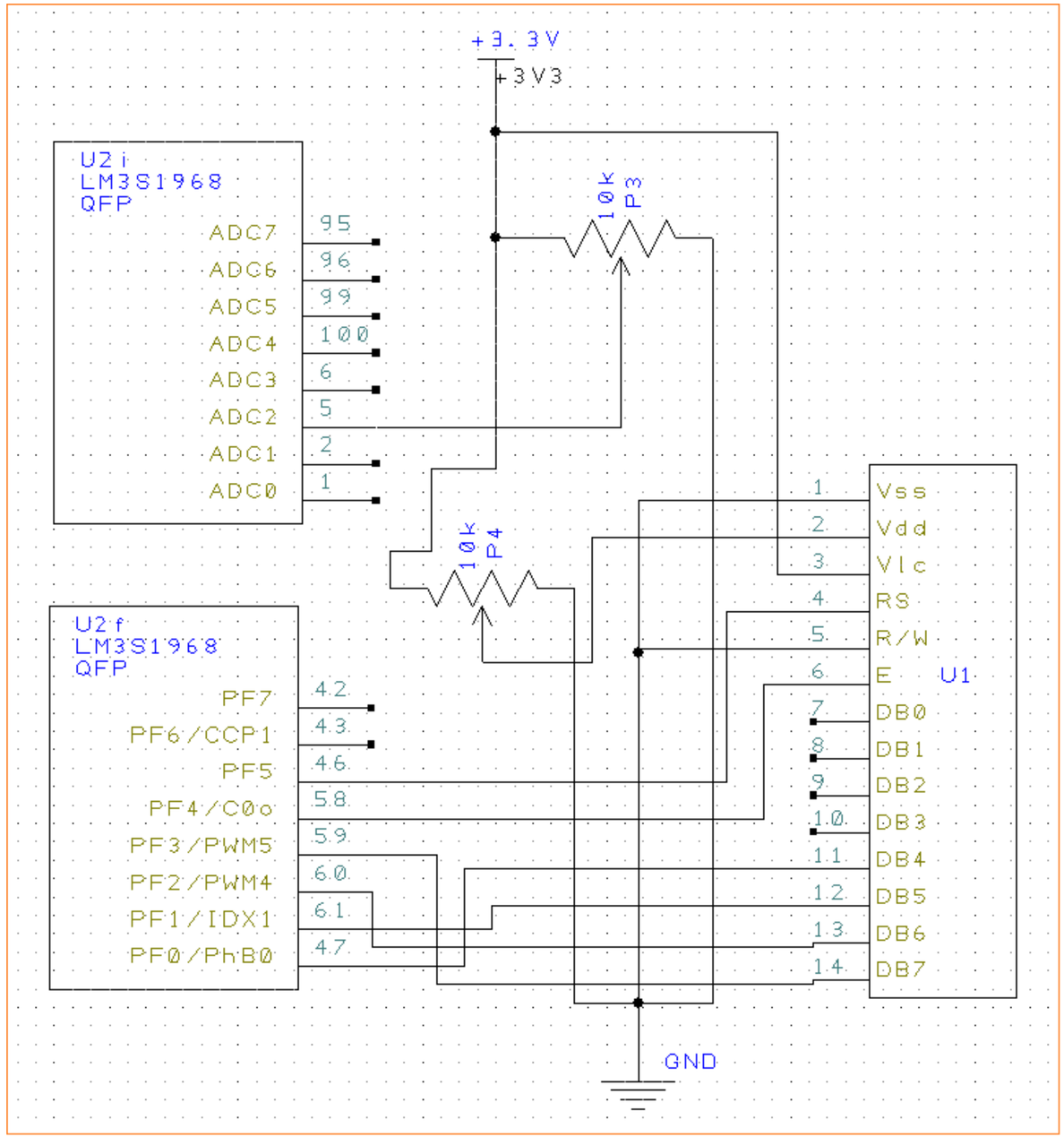
Call Stack + Locals

Name	Location/Value	Type
ADC_In	0x0000058C	unsigned long f()
result	<not in scope>	auto - unsigned long
SysTickIntHan...	0x000006B8	void f()
main	0x000002B4	int f()

Call Stack + Locals | UART #1 | Memory 1

Simulation | t1: 0.38678468 sec | L:139 C:1 | CAP NUM SCRL OVR R/W

1:56 PM 4/10/2013


+3.3V
+3V3
U2i
LM3S1968
QFP
ADC7
ADC6
ADC5
ADC4
ADC3
ADC2
ADC1
ADC0
95
96
99
100
6
5
2
1
10k
P3
10k
P4
U2f
LM3S1968
QFP
PF7
PF6/CCP1
PF5
PF4/C0o
PF3/PWM5
PF2/PWM4
PF1/IDX1
PF0/PhB0
42
43
46
58
59
60
61
47
Vss
Vdd
Vlc
RS
R/W
E
DB0
DB1
DB2
DB3
DB4
DB5
DB6
DB7
U1
1
2
3
4
5
6
7
8
9
10
11
12
13
14
GND

Position	Analog Input	ADC Sample	Correct Fixed-Point	Measured Fixed-Point
0	0	0.088	0	0
0.5	0.74	0.329	500	497
1	1.65	0.567	1000	1001
1.5	2.57	0.822	1500	1498
1.95	3.26	1.022	1950	1950

```c
int main(void){
    init(); // Bus clock is 50 MHz
    LCDInit();
    LCDClear();
    ADC_InitSWTriggerSeq3(2); // turn on ADC, set channel to 2, sequencer 3
    SysTickInit();
    while(1) {
        // wait for mailbox flag ADCStatus to be true
        while (HWREGBITW(&gFlags, FLAG_ADC_VALUE) == 0) { }
        // read the 10-bit ADC sample from the mailbox ADCMail
        Data = ADCvalue;
        // clear the mailbox flag ADCStatus to signify the mailbox is now empty
        HWREGBITW(&gFlags, FLAG_ADC_VALUE) = 0;
        // convert the sample into a fixed point number
        Convert(Data);
        // output the fixed point number on the LCD with units
        LCDCursor(0);
        LCDOutString(msg);
        LCDOutString("cm");
    }
}
```

True Position	False Position	Error
0.000	0.000	0.000
0.500	0.501	0.001
1.000	1.001	0.001
1.500	1.500	0.000
2.000	2.000	0.000
	Average Error	0.0004