**Lab 4 Prep**

**C) Modify lines 97,98,99**

#define SSID\_NAME "Caroline Phone"

#define SEC\_TYPE SL\_SEC\_TYPE\_OPEN

#define PASSKEY "bang133\*"

**D) Write a C function that extracts the Austin temperature from this buffer.**

char\* ParseBuffer(char\* recvbuff){

uint32\_t j = 0;

uint32\_t k = TEMP\_START\_INDEX;

for(uint32\_t i = 0; i < MAX\_RECV\_BUFF\_SIZE; ++i){

if(recvbuff[i] == 't'){

int result = CompareString(&recvbuff[i+1],"emp\":",5);

if(result){

for(j=i+6; j<TEMP\_STRING\_SIZE; ++j){

if(recvbuff[j] == ','){

recvbuff[j+1] = ' ';

recvbuff[j+2] = 'F';

ST7735\_OutString(Temp);

return Temp;

}

Temp[k] = recvbuff[i+6];

++k;

}

}

}

}

return "0";

}

**F) Write software to sample the ADC once using 64-point hardware averaging and calculate a physical parameter with units.**

void ReadVoltage(void){

uint32\_t ADCVal = ADC0\_InSeq3();

uint32\_t voltage = (ADCVal \* 3300 + 2048)/4096; //converts value to fixed point resolution .001

ST7735\_OutString("Voltage~");

ST7735\_OutUDec(voltage/1000);

ST7735\_OutChar('.');

ST7735\_OutUDec(voltage%1000);

}

**ADC Sampling Functions**

void ADC0\_InitSWTriggerSeq3\_Ch9(void){

SYSCTL\_RCGCADC\_R |= 0x0001; // 7) activate ADC0

// 1) activate clock for Port E

SYSCTL\_RCGCGPIO\_R |= 0x10;

while((SYSCTL\_PRGPIO\_R&0x10) != 0x10){};

GPIO\_PORTE\_DIR\_R &= ~0x10; // 2) make PE4 input

GPIO\_PORTE\_AFSEL\_R |= 0x10; // 3) enable alternate function on PE4

GPIO\_PORTE\_DEN\_R &= ~0x10; // 4) disable digital I/O on PE4

GPIO\_PORTE\_AMSEL\_R |= 0x10; // 5) enable analog functionality on PE4

// while((SYSCTL\_PRADC\_R&0x0001) != 0x0001){}; // good code, but not yet implemented in simulator

ADC0\_PC\_R &= ~0xF; // 7) clear max sample rate field

ADC0\_PC\_R |= 0x1; // configure for 125K samples/sec

ADC0\_SSPRI\_R = 0x0123; // 8) Sequencer 3 is highest priority

ADC0\_ACTSS\_R &= ~0x0008; // 9) disable sample sequencer 3

ADC0\_EMUX\_R &= ~0xF000; // 10) seq3 is software trigger

ADC0\_SSMUX3\_R &= ~0x000F; // 11) clear SS3 field

ADC0\_SSMUX3\_R += 9; // set channel

ADC0\_SSCTL3\_R = 0x0006; // 12) no TS0 D0, yes IE0 END0

ADC0\_IM\_R &= ~0x0008; // 13) disable SS3 interrupts

ADC0\_SAC\_R = 0x06; //64x hardware averaging

ADC0\_ACTSS\_R |= 0x0008; // 14) enable sample sequencer 3

}

//------------ADC0\_InSeq3------------

// Busy-wait Analog to digital conversion

// Input: none

// Output: 12-bit result of ADC conversion

uint32\_t ADC0\_InSeq3(void){ uint32\_t result;

ADC0\_PSSI\_R = 0x0008; // 1) initiate SS3

while((ADC0\_RIS\_R&0x08)==0){}; // 2) wait for conversion done

// if you have an A0-A3 revision number, you need to add an 8 usec wait here

result = ADC0\_SSFIFO3\_R&0xFFF; // 3) read result

ADC0\_ISC\_R = 0x0008; // 4) acknowledge completion

return result;

}