The structure of a multiple-sources C project

File	Mandatory	Contains	Examples

Definitions.h	No,
	Highly recommended
		Configuration bits (`#pragma config`)
		
		`#defines`
		
		`#include <xc.h>`
			`#pragma config OSC = XT  // CONFIG1H`
			`#pragma config FCMEN = OFF // CONFIG7H`
			`#pragma config EBTRB = OFF`
			`#define _XTAL_FREQ 8000000  // Fosc 8MHz`
			
			`#include <xc.h>`

ADC.h	No recommended as a match for ADC.c	Prototype functions from ADC.c `#defines`	`void ADC_Init(void);` `int ADC_Read(int);`

LCD.h	No,
	Recommended as a match for LCD.c
		Prototype functions from LCD.c
		`#defines`
			`void LCD_Init(void);`
			`void LCD_Command(char);`
			
			`#define RS LATD0 //PORTD 0 pin …`
			`#define EN LATD1 //PORTD 1 pin …`

Main.c	Yes	Main functions
		`#defines`, when needed
		Other functions, when needed
			`#include "Definitions.h"`
			`#include "LCD.h"`
			`#include "ADC.h"`
			`#define vref 5.00`
			`void main() {`
			`  char data[10];`
			`  while(1) {`
			`  digital = ADC_Read(0);}`
			`  return;`
			`}`

ADC.C	No
	Recommended for ADC functions
		`#includes` (Definitions.h, ADC.h, …)
		ADC related functions
			`#include "Definitions.h"`
			`#include "ADC.h"`
			
			`void LCD_Clear() {`
			`  LCD_Command(0x01);  // clear`
			`}`

LCD.c	No
	Recommended for ADC functions
		`#includes` (Definitions.h, LCD.h, …)
		LCD related functions
			`#include "Definitions.h"`
			`#include "LCD.h"`
			
			`void LCD_Init() {`
			`  __delay_ms(15);  // 15ms`
			`  LCD_Port = 0x00;`
			`}`

```
void ADC_Init(void);
int ADC_Read(int);
```

ADC.h

```
void LCD_Init(void);
void LCD_Command(char);
void LCD_String_xy(char, char, const char*);

#define RS LATD0           // PORTD 0 pin is used for RS
#define EN LATD1           // PORTD 1 pin is used for Enable
#define ldata LATB         // PORTB used for Tx data to LCD
#define LCD_Port TRISB     // define macro for PORTB Direction
#define LCD_Control TRISD  // define macro for PORTD Direction
```

LCD.h

```
// PIC18F4520 Configuration Bit Settings
// 'C' source line config statements

// CONFIG1H
#pragma config OSC = XT          // Oscillator Selection bits
#pragma config FCMEN = OFF       // Fail-Safe Clock Monitor
#pragma config IESO = OFF        // Internal/External Osc

// CONFIG7H
#pragma config EBTRB = OFF       // Boot Block Table Read
#define _XTAL_FREQ 8000000       // Fosc 8MHz XTAL frequency
                                 // used ONLY for _delay()
#include <xc.h>
```

Definitions.h

```c
#include "Definitions.h"
#include "ADC.h"

void ADC_Init() {
    TRISA = 0xFF;        //Set as input port
    ADCON2 = 0x92;       //Right Justified, 4Tad and Fosc/32
}

int ADC_Read(int channel) {
    int digital;
    ADCON0 =(ADCON0 & 0b11000011)|((channel<<2) & 0b00111100);
return(digital);
}
```

ADC.c

```c
#include "Definitions.h"
#include "LCD.h"

void LCD_Init() {
    __delay_ms(15);      // 15ms,16x2 LCD Power on delay
    LCD_Port = 0x00;     // Set PORTB as output (D0-D7) pins
}

void LCD_Clear() {
    LCD_Command(0x01);  // clear display screen*/
}

void LCD_Command(char cmd ) {
    ldata= cmd; //Send data to PORT as a command for LCD
}
```

LCD.c

```c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include "Definitions.h"
#include "LCD.h"
#include "ADC.h"

#define vref 5.00               // Reference Voltage is 5V

void main() {
    char data[10];
    int digital;
    float voltage;

    LCD_Init();                 // Initialize 16x2 LCD
    ADC_Init();                 // Initialize 10-bit ADC

    while(1) {
        digital = ADC_Read(0); // Convert analog voltage
                               // into a digital value
        voltage = digital*((float)vref/(float)1023);

        strcat(data," V");      // Concatenate result and
                               // unit to print*/
    }
return;
}
```

Main.c