C++ / OOP
File Management

Last updated 3/4/19

C++/OOP File Management

- Workspace
  - Eclipse uses the concept of a workspace to manage projects
  - Create a single workspace
    - No Spaces in the path/names
    - Located in a place you can find it (outside of the eclipse install)
    - I use “workspace_class#_eclipse

      e.g workspace_ee2510_eclipse

      workspace_ee2920_ccstudio

C++/OOP File Management

- .cpp files and .h files
  - .cpp files are used to store C++ code
    - Project code
    - Library code (collected functions)
    - Class definition code
  - .h files are used to store prototypes and constants
    - Function prototypes
    - Class prototypes
      - Member variables
      - Member function prototypes
    - Constants

C++/OOP File Management

- Header Files
  - xxxx.h
  - Store prototypes and constants
    - Constants
      - Pin / Bit numbers and names (msp.h)
    - Structure definitions
    - Enumerated types
    - Function declarations (prototypes)
    - Class declarations (class specification file)
      - Member variable declarations
      - Member function declarations (prototypes)
    - Wrapped in an “include guard” to prevent including the code multiple times

C++/OOP File Management

- Header File - Include guard
  - Prevents the same code from being included multiple times

```
#ifndef MYFILENAME_H
#define MYFILENAME_H
...
declarations
...
#endif
```

Check to see if the constant MYFILENAME_H has not been defined – #ifndef

If it is not defined,
create the constant - #define
execute the commands between #define and #endif

If it has been defined
skip to #endif

All caps used for the constant
Based on .h file name with dot replaced by _

Constant is not initialized or set

C++/OOP File Management

- Header File - Inclusion
  - Header files are #included into the .cpp file using the module
  - Optionally they can be included into the related module .cpp file

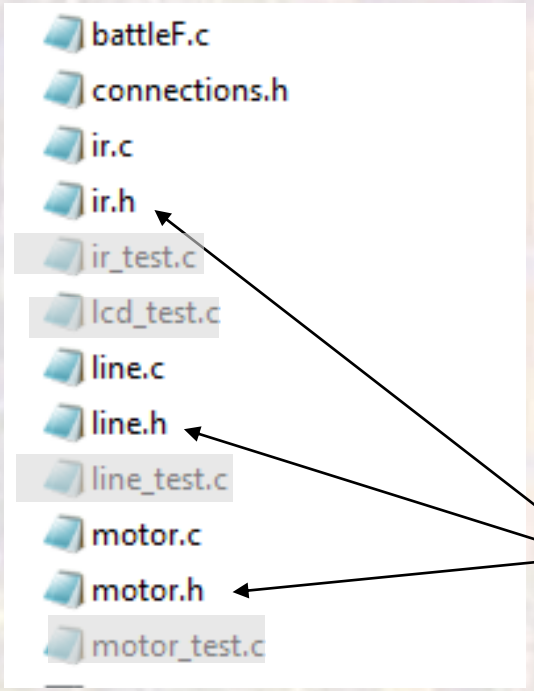
Note – c++ system header files are enclosed in angled brackets < >

user defined header files are enclosed in double quotes " "

C++/OOP File Management

- Header File – Inclusion
  - Sumo bot example

Project Files



Top level program
battleF.c

battleF.c - Notepad

File Edit Format View Help

```
/*
 * battleF.c
 *
 *  Created on: Feb 3, 2018
 *      Author: Tim
 */

#include <stdio.h>
#include "msp432.h"
#include "msoe_lib_all.h"
#include "motor.h"
#include "line.h"
#include "ir.h"
```

C++/OOP File Management

- Header File – Inclusion
  - Sumo bot example

IR .h file

ir.h - Notepad

File Edit Format View Help

```
 * ir.h
 *
 *  Created on: Jan 17, 2018
 *      Author: Tim
 */

#ifndef IR_H_
#define IR_H_

//////////////////////////////////////
//
// IR_setup()
//    Sets up the pins for the IR rx/tx
//    Uses 2 IR transmitters and 2 IR receivers
//
// Transmitters are IR diodes and require one pin each
// L tx - P10.5
// R tx - P10.5     -  common output
//
// Sensors require Vcc, gnd, and 1 output
// L rx - P10.2
// R rx - P10.3
//
//////////////////////////////////////
void ir_setup(void);

//////////////////////////////////////
//
// check_ir(l_ptr, r_ptr)
//    modifies the pointers based on sensor output values
//
//////////////////////////////////////
void check_ir(int * left, int * right);

#endif /* IR_H_ */
```

IR .c file

```
/*
 * ir.c
 *
 *  Created on: Dec 28, 2017
 *      Author: johnsontimoj
 */

#include "msp432.h"
#include "msoe_lib_all.h"
#include <stdio.h>

//////////////////////////////////////
//
// IR sensor routines
//
// 1) IR_setup
//    Sets up the pins for the IR rx/tx
//    Sets up the 38KHz PWM signal - TimerA3
//    Sets up the PWM envelope signal - TimerA2
//
// 2) check_IR
//    check the 2 IR sensors
//
// 3)
//    IR PWM generation interrupt service routine
//
//////////////////////////////////////
//
// IR_setup()
//    Sets up the pins for the IR rx/tx
//    Uses 2 IR transmitters and 2 IR receivers
//
```

• • •

```
//////////////////////////////////////
void ir_setup(void){
    //
    // setup pins
    //
    // tx outputs          P10.5
    P10->SEL0 |= 0x20;
    P10->SEL1 &= ~0x20;
    P10->DIR |= 0x20;

    // rx inputs          P10.2, P10.3
```

C++/OOP File Management

- Project Build
  - The IDE (Eclipse)
    - Includes all the files “included” in the top level file (the one containing main), and all files “included” in those files
      - This gives the compiler a complete set of function/object prototypes
    - Compiles all the .cpp files in the project that have not been excluded from the build
    - Builds the overall solution