```
// FILE HEADER EXAMPLES:

//===========================================================================
//  File Name : ports.c
//
//  Description: This file contains the Initialization for all port pins
//
//  Author: John Doe
//  Date: Sept 2013
//  Compiler: Built with IAR Embedded Workbench Version: V4.10A/W32 (5.40.1)
//===========================================================================


//===========================================================================
// File Name : macros.h
//
//  Description: This file contains the macros used by all the other files
//
//  Author: John Doe
//  Date: Sept 2013
//  Compiler: Built with IAR Embedded Workbench Version: V4.10A/W32 (5.40.1)
//===========================================================================
// FUNCTION HEADER EXAMPLE
//===========================================================================
// Function name: Main
//
// Description: This function contains the while loop that runs continuously
// to act for the operating system. It also calls all the functions to
// initialize the system.
//
// Passed : no variables passed
// Locals: no variables declared
// Returned: no values returned
// Globals: char* display_1
//          char* display_2
//          slow_input_down
//          control_state[CNTL_STATE_INDEX]
//
//  Author: John Doe
//  Date: Sept 2013
//  Compiler: Built with IAR Embedded Workbench Version: V4.10A/W32 (5.40.1)
//===========================================================================
// SAMPLE FOR PORT1, WITH COMPLETE HEADERS AND COMMENTS
//=====================================================================
// Init_Port1
// Purpose: Initialize Port 1
//
// Various options for Pin 0
// SEL0 SEL1 DIR
//  0    0   I:0 O:1        - *General-purpose digital I/O with port interrupt and wake up from LPMx.5
//  0    1    0             - TA0 CCR1 capture: CCI1A input
//  0    1    1             - TA0 CCR1 compare: Out1
//  1    0    0             - External DMA trigger
//  1    0    1             - RTC clock calibration output
//  1    1    X             - Analog input A0 - ADC, Comparator_D input CD0, Vref- External applied reference
//
// Various options for Pin 1
// SEL0 SEL1 DIR
//  0    0   I:0 0:1       - *General-purpose digital I/O with port interrupt and wake up from LPMx.5
//  0    1    0            - TA0 CCR2 capture: CCI2A input,
//  0    1    1            - TA0 CCR2 compare: Out2
//  1    0    0            - TA1 input clock
//  1    0    1            - Comparator_D output
//  1    1    X            - Analog input A1 - ADC, Comparator_D input CD1, Input for an external reference voltage to
the ADC
//
// Various options for Pin 2
// SEL0 SEL1 DIR
//  0    0   I:0 O:1       - *General-purpose digital I/O with port interrupt and wake up from LPMx.5
//  0    1    0            - TA1 CCR1 capture: CCI1A input
//  0    1    1            - TA1 CCR1 compare: Out1
//  1    0    0            - TA0 input clock
//  1    0    1            - Comparator_D output
//  1    1    X            - Analog input A2 - ADC, Comparator_D input CD2
//
// Various options for Pin 3
// SEL0 SEL1 DIR
//  0    0   I:0 O:1       - *General-purpose digital I/O with port interrupt and wake up from LPMx.5
//  0    1    0            - TA1 CCR2 capture: CCI2A input
//  0    1    1            - TA1 CCR2 compare: Out2
//  1    0    X            - Slave transmit enable - eUSCI_B0 SPI mode
//  1    1    X            - Analog input A3 - ADC, Comparator_D input CD3
//
// Various options for Pin 4
// SEL0 SEL1 DIR
//  0    0   I:0 O:1       - *General-purpose digital I/O with port interrupt and wake up from LPMx.5
//  0    1    0            - TB0 CCR1 capture: CCI1A input
//  0    1    1            - TB0 CCR1 compare: Out1
//  1    0    X            - Slave transmit enable - eUSCI_A0 SPI mode
//  1    1    X            - Analog input A4 - ADC, Comparator_D input CD4
//
// Various options for Pin 5
// SEL0 SEL1 DIR
//  0    0   I:0 O:1       - *General-purpose digital I/O with port interrupt and wake up from LPMx.5
//  0    1    0            - TB0 CCR2 capture: CCI2A input
//  0    1    1            - TBO CCR2 compare: Out2
//  1    0    X            - Clock signal input - eUSCI_A0 SPI slave, Clock signal output - eUSCI_A0 SPI master
//  1    1    X            - Analog input A5 - ADC, Comparator_D input CD5
//
// Various options for Pin 6
// SEL0 SEL1 DIR
//  0    0   I:0 O:1       - General-purpose digital I/O with port interrupt and wake up from LPMx.5
//  0    1    0            - TB1 CCR1 capture: CCI1A input
//  0    1    1            - TB1 CCR1 compare: Out1
//  1    0    X            - *Slave in, master out - eUSCI_B0 SPI mode, I2C data - eUSCI_B0 I2C mode
//  1    1    0            - TA0 CCR0 capture: CCI0A input
//  1    1    1            - TA0 CCR0 compare: Out0
//
// Various options for Pin 7
// SEL0 SEL1 DIR
//  0    0   I:0 O:1       - *General-purpose digital I/O with port interrupt and wake up from LPMx.5
//  0    1    0            - TB1 CCR2 capture: CCI2A input
//  0    1    1            - TB1 CCR2 compare: Out2 (not available on devices without TB1)
//  1    0    X            - Slave out, master in - eUSCI_B0 SPI mode, I2C clock - eUSCI_B0 I2C mode
//  1    1    0            - TA1 CCR0 capture: CCI0A input
//  1    1    1            - TA1 CCR0 compare: Out0
//
// Passed: No variables passed
// Locals: No local variables
// Returned: No values returned
//
// Author: John Doe
// Date: Sept 2013
// Compiler: Built with IAR Embedded Workbench Version: V4.10A/W32 (5.40.1)
// Version: 1.0
//------------------------------------------------------------------------------
void Init_Port1(void){
  // Set Selections bits
  P1SEL0 = CLEAR_REGISTER;         // P1 set default as I/0
  P1SEL1 = CLEAR_REGISTER;         // P1 set default as I/0
  P1SEL0 &= ~SPI_SIMO;             // SPI Function SIMO_B selected
  P1SEL1 |= SPI_SIMO;              // SPI Function SIMO_B selected
  P1SELC |= V_DETECT_R;            // Analog V_DETECT_R selected
  P1SELC |= V_DETECT_L;            // Analog V_DETECT_L selected
  P1SELC |= V_THUMB;               // Analog V_THUMB selected
  //Set Pin Direction bits
  P1OUT = CLEAR_REGISTER;          // Set all outputs low for safety
  P1DIR = INPUT_SET;               // Set P1 default direction to input
  P1DIR |= SPI_SIMO;               // SIMO_B set to Output
  P1DIR |= RS_LCD;                 // Set P1 RS_LCD direction to output
  P1DIR |= GPS_RESET;              // Set P1 GPS_RESET direction to output
  P1DIR |= IR_LED;                 // Set P1 IR_LED direction to output

  //Initialize outputs
  P1OUT &= ~GPS_RESET;             // Set GPS_RESET low
  P1OUT &= ~IR_LED;                // Set IR_LED low
  P1OUT |= SPI_SIMO;               // Set IR_LED high Configure for pullup resistor
  P1OUT |= RS_LCD;                 // Set IR_LED high Configure for pullup resistor

  //Enable resistors
  P1REN |= RS_LCD;                 // Enable pullup resistor
  P1REN |= SPI_SIMO;               // Enable pullup resistor

}
```