```
/*
 * MCP23017_CO2_level_LED_simple_display.h
 *
 * Created: 4/26/2022 9:25:26 PM
 *  Author: jason
 */


#ifndef MCP23017_CO2_LEVEL_LED_SIMPLE_DISPLAY_H_
#define MCP23017_CO2_LEVEL_LED_SIMPLE_DISPLAY_H_


#include <avr/io.h>
#define F_CPU 4000000     // 4MHz default clock
#include <util/delay.h>

//Defines for GIO, b1 means when 8-bit mode, BANK = 1
#define IOCONaddr_b0 0x0A   //address at reset, default 16-bit mode
#define IOCONaddr_b1 0x05   //Control registers
#define IODIRAaddr_b1 0x00  //Directional register for PORTA
#define IODIRBaddr_b1 0x10  //Directional register for PORTB
#define GPPUAaddr_b1 0x06
#define GPIOAaddr_b1 0x09   //GPIOA I/O PORT register
#define OLATBaddr_b1 0x1A   //GPIO output registers
#define WRITE_opcode 0x40
#define READ_opcode 0x41

//Function Prototypes that will be used
void I2C0_MCP23017_init();
void MCP23017_I2C_init();
void MCP23017_I2C_write(uint8_t, uint8_t, uint8_t);



//Initializes the AVR128DB48's I2C0 to commmunicate with the MCP23017.
//The bit transfer rate between the AVR128DB48 and the MCP23017 must be
//as fast as possible, but less than or equal to 400 kb/s.
void I2C0_MCP23017_init()
{
    //Baud rate for the I2C which set to 0 assuming that is the fastest you can get to
    TWI0.MBAUD = 0;


    //Enable for the I2C Master
    TWI0.MCTRLA = TWI_ENABLE_bm;

    //Force the I2C to the idle state
    TWI0.MSTATUS = TWI_BUSSTATE_IDLE_gc;
}

//This function initializes the MCP23017. Port A of the GPIO (GPA) must be
//configured as all inputs with pull ups enabled. GPB must be
```

```
//configured as all outputs.
void MCP23017_I2C_init(){

    MCP23017_I2C_write(WRITE_opcode, IOCONaddr_b0, 0xA0);

    //Configure PORTB of the GPIO MCP23017 as the output
    MCP23017_I2C_write(WRITE_opcode, IODIRBaddr_b1, 0x00);
}


//This function is what write to the actual GPIO expander MCP23017
//to access the registers and modifying those bit fields
void MCP23017_I2C_write(uint8_t opcode, uint8_t address, uint8_t data){
    TWI0_MADDR = opcode;   //Pass the opcode to the address

    //Poll until master transmit address of byte write operation is complete
      regardless
    while(!(TWI0.MSTATUS & TWI_WIF_bm));

    TWI0_MDATA = address; //Pass the address to master data

    //Poll until master transmit address of byte write operation is complete
      regardless
    while(!(TWI0.MSTATUS & TWI_WIF_bm));

    //Pass the data to master data
    TWI0_MDATA = data;

    //Poll until master transmit address of byte write operation is complete
      regardless
    while(!(TWI0.MSTATUS & TWI_WIF_bm));


    //Execute acknowledge action followed by issuing a stop condition
    TWI0_MCTRLB = TWI_MCMD_STOP_gc;
}




#endif /* MCP23017_CO2_LEVEL_LED_SIMPLE_DISPLAY_H_ */
```