```c
#include "LCD_IO_Init.h"


void LCD_IO_Init()
{
  RCC->APB2ENR |= RCC_APB2ENR_IOPCEN | RCC_APB2ENR_IOPBEN | RCC_APB2ENR_IOPAEN;

  GPIOC->CRL |= GPIO_CRL_MODE7 | GPIO_CRL_MODE6 | GPIO_CRL_MODE5 | GPIO_CRL_MODE4 | GPIO_CRL_MODE3 | GPIO_CRL_MODE2 | GPIO_CRL_MODE1 | GPIO_CRL_MODE0 ;
  GPIOC->CRL &= ~GPIO_CRL_CNF7 & ~GPIO_CRL_CNF6 & ~GPIO_CRL_CNF5 & ~GPIO_CRL_CNF4 & ~GPIO_CRL_CNF3 & ~GPIO_CRL_CNF2 & ~GPIO_CRL_CNF1 & ~GPIO_CRL_CNF0 ;

  GPIOB->CRL |= GPIO_CRL_MODE5 | GPIO_CRL_MODE1 | GPIO_CRL_MODE0 ;
  GPIOB->CRL &= ~GPIO_CRL_CNF5 & ~GPIO_CRL_CNF1 & ~GPIO_CRL_CNF0 ;

  commandToLCD(LCD_8B2L, 0);
  commandToLCD(LCD_8B2L, 0);
  commandToLCD(LCD_8B2L, 0);
  commandToLCD(LCD_8B2L, 0);
  commandToLCD(LCD_DCB, 0);
  commandToLCD(LCD_CLR, 0);
  commandToLCD(LCD_MCR, 0);
}

/*
* Name: commandToLCD
* Type: PUBLIC
* Parameters: a single byte of command information for the LCD controller
* Returns: nothing
* Description: This function generates control timing and data signals to send one command byte to the LCD
*/
void commandToLCD(uint8_t data, int CORD)
{
  if (CORD == 0)
  {
    GPIOB->BSRR = LCD_CM_ENA; //RS low, E high
    // GPIOC->ODR = data; //BAD: may affect upper bits on port C
    GPIOC->ODR &= 0xFF00; //GOOD: clears the low bits without affecting high bits
    GPIOC->ODR |= data; //GOOD: only affects lowest 8 bits of Port C
    delay(80000);
    GPIOB->BSRR = LCD_CM_DIS; //RS low, E low
    delay(800000);
  }
  //send to LDC screen aka display
  else if (CORD == 1)
  {
    GPIOB->BSRR = LCD_DM_ENA; //RS low, E high
    // GPIOC->ODR = data; //BAD: may affect upper bits on port C
    GPIOC->ODR &= 0xFF00; //GOOD: clears the low bits without affecting high bits
    GPIOC->ODR |= data; //GOOD: only affects lowest 8 bits of Port C
    delay(80000);
    GPIOB->BSRR = LCD_DM_DIS; //RS low, E low
    delay(800000);
  }
}
```