

Proyecto. Medidor de altura 2

```
#ifndef FDB_LIQUID_CRYSTAL_I2C_H
#define FDB_LIQUID_CRYSTAL_I2C_H

#include <inttypes.h>
#include <Print.h>

// commands
#define LCD_CLEARDISPLAY 0x01
#define LCD_RETURNHOME 0x02
#define LCD_ENTRYMODESET 0x04
#define LCD_DISPLAYCONTROL 0x08
#define LCD_CURSORSHIFT 0x10
#define LCD_FUNCTIONSET 0x20
#define LCD_SETCGRAMADDR 0x40
#define LCD_SETDDRAMADDR 0x80

// flags for display entry mode
#define LCD_ENTRYRIGHT 0x00
#define LCD_ENTRYLEFT 0x02
#define LCD_ENTRYSHIFTINCREMENT 0x01
#define LCD_ENTRYSHIFTDECREMENT 0x00

// flags for display on/off control
#define LCD_DISPLAYON 0x04
#define LCD_DISPLAYOFF 0x00
#define LCD_CURSORON 0x02
#define LCD_CURSOROFF 0x00
#define LCD_BLINKON 0x01
#define LCD_BLINKOFF 0x00

// flags for display/cursor shift
#define LCD_DISPLAYMOVE 0x08
#define LCD_CURSORMOVE 0x00
#define LCD_MOVERIGHT 0x04
#define LCD_MOVELEFT 0x00

// flags for function set
#define LCD_8BITMODE 0x10
#define LCD_4BITMODE 0x00
```

Tecnología B. Proyecto. Medidor de altura 2

Proyecto. Medidor de altura 2

```

#define LCD_2LINE 0x08
#define LCD_1LINE 0x00
#define LCD_5x10DOTS 0x04
#define LCD_5x8DOTS 0x00

// flags for backlight control
#define LCD_BACKLIGHT 0x08
#define LCD_NOBACKLIGHT 0x00

#define En B00000100 // Enable bit
#define Rw B00000010 // Read/Write bit
#define Rs B00000001 // Register select bit

/**
 * This is the driver for the Liquid Crystal LCD displays that use the I2C bus.
 *
 * After creating an instance of this class, first call begin() before anything else.
 * The backlight is on by default, since that is the most likely operating mode in
 * most cases.
 */
class LiquidCrystal_I2C : public Print {
public:
    /**
     * Constructor
     *
     * @param lcd_addr      I2C slave address of the LCD display. Most likely
    printed on the
     *
     *                      LCD circuit board, or look in the supplied LCD
    documentation.
     * @param lcd_cols      Number of columns your LCD display has.
     * @param lcd_rows      Number of rows your LCD display has.
     * @param charsize      The size in dots that the display has, use LCD_5x10DOTS
    or LCD_5x8DOTS.
     */
    LiquidCrystal_I2C(uint8_t lcd_addr, uint8_t lcd_cols, uint8_t lcd_rows, uint8_t
    charsize = LCD_5x8DOTS);

    /**
     * Set the LCD display in the correct begin state, must be called before anything
    else is done.

```

Proyecto. Medidor de altura 2

```
    */
void begin();

/**
 * Remove all the characters currently shown. Next print/write operation will
start
 * from the first position on LCD display.
 */
void clear();

/**
 * Next print/write operation will will start from the first position on the
LCD display.
 */
void home();

/**
 * Do not show any characters on the LCD display. Backlight state will remain
unchanged.
 * Also all characters written on the display will return, when the display in
enabled again.
 */
void noDisplay();

/**
 * Show the characters on the LCD display, this is the normal behaviour. This
method should
 * only be used after noDisplay() has been used.
 */
void display();

/**
 * Do not blink the cursor indicator.
 */
void noBlink();

/**
 * Start blinking the cursor indicator.
 */
void blink();
```

Proyecto. Medidor de altura 2

```

/**
 * Do not show a cursor indicator.
 */
void noCursor();

/**
 * Show a cursor indicator, cursor can blink on not blink. Use the
 * methods blink() and noBlink() for changing cursor blink.
 */
void cursor();

void scrollDisplayLeft();
void scrollDisplayRight();
void printLeft();
void printRight();
void leftToRight();
void rightToLeft();
void shiftIncrement();
void shiftDecrement();
void noBacklight();
void backlight();
void autoscroll();
void noAutoscroll();
void createChar(uint8_t, uint8_t[]);
void setCursor(uint8_t, uint8_t);
virtual size_t write(uint8_t);
void command(uint8_t);

inline void blink_on() { blink(); }
inline void blink_off() { noBlink(); }
inline void cursor_on() { cursor(); }
inline void cursor_off() { noCursor(); }

// Compatibility API function aliases
void setBacklight(uint8_t new_val); // alias for backlight()
and nobacklight()

void load_custom_character(uint8_t char_num, uint8_t *rows); // alias for
createChar()

```

Proyecto. Medidor de altura 2

```
void printstr(const char[]);

private:
void send(uint8_t, uint8_t);
void write4bits(uint8_t);
void expanderWrite(uint8_t);
void pulseEnable(uint8_t);
uint8_t _addr;
uint8_t _displayfunction;
uint8_t _displaycontrol;
uint8_t _displaymode;
uint8_t _cols;
uint8_t _rows;
uint8_t _charsize;
uint8_t _backlightval;
};

#endif // FDB_LIQUID_CRYSTAL_I2C_H
```