

# CustomChars

---

```
//YWROBOT
//Compatible with the Arduino IDE 1.0
//Library version:1.1
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

#if defined(ARDUINO) && ARDUINO >= 100
#define printByte(args) write(args);
#else
#define printByte(args) print(args,BYTE);
#endif

uint8_t bell[8] = {0x4,0xe,0xe,0xe,0x1f,0x0,0x4};
uint8_t note[8] = {0x2,0x3,0x2,0xe,0x1e,0xc,0x0};
uint8_t clock[8] = {0x0,0xe,0x15,0x17,0x11,0xe,0x0};
uint8_t heart[8] = {0x0,0xa,0x1f,0x1f,0xe,0x4,0x0};
uint8_t duck[8] = {0x0,0xc,0x1d,0xf,0xf,0x6,0x0};
uint8_t check[8] = {0x0,0x1,0x3,0x16,0x1c,0x8,0x0};
uint8_t cross[8] = {0x0,0x1b,0xe,0x4,0xe,0x1b,0x0};
uint8_t retarrow[8] = { 0x1,0x1,0x5,0x9,0x1f,0x8,0x4};

LiquidCrystal_I2C lcd(0x27,20,4); // set the LCD address to 0x27 for a 16
chars and 2 line display

void setup()
{
    lcd.init();                // initialize the lcd
    lcd.backlight();

    lcd.createChar(0, bell);
    lcd.createChar(1, note);
    lcd.createChar(2, clock);
    lcd.createChar(3, heart);
    lcd.createChar(4, duck);
    lcd.createChar(5, check);
    lcd.createChar(6, cross);
    lcd.createChar(7, retarrow);
}
```

```

    lcd.home();

    lcd.print("Hello world...");
    lcd.setCursor(0, 1);
    lcd.print(" i ");
    lcd.printByte(3);
    lcd.print(" arduinos!");
    delay(5000);
    displayKeyCodes();

}

// display all keycodes
void displayKeyCodes(void) {
    uint8_t i = 0;
    while (1) {
        lcd.clear();
        lcd.print("Codes 0x"); lcd.print(i, HEX);
        lcd.print("-0x"); lcd.print(i+16, HEX);
        lcd.setCursor(0, 1);
        for (int j=0; j<16; j++) {
            lcd.printByte(i+j);
        }
        i+=16;

        delay(4000);
    }
}

void loop()
{

}

```

# 코드 기능 요약

## ✓ 목적

I2C LCD (20x4 LCD) 화면에 사용자 정의 문자(아이콘)를 생성·출력하고 LCD 문자 코드셋을 순차적으로 화면에 보여주는 프로그램

---

## 주요 구성

구분	내용
라이브러리	<code>Wire.h</code> , <code>LiquidCrystal_I2C.h</code> (I2C LCD 제어용)
LCD 설정	I2C 주소 <code>0x27</code> , 화면 크기 <code>20x4</code>
커스텀 문자 생성	벨, 음표, 시계, 하트, 오리, 체크, X표, 돌아가기 화살표 (총 8개)
출력 내용	<code>"Hello world..."</code> , <code>"i ♥ arduinos!"</code> , 그 후 LCD 문자코드셋 차례로 표시

---

## 세부 코드 동작

코드	동작
<code>lcd.init()</code>	LCD 초기화
<code>lcd.backlight()</code>	백라이트 켜기
<code>lcd.createChar()</code>	커스텀 문자(아이콘) 등록 (0~7번 슬롯)
<code>lcd.home()</code>	커서 맨 좌측 상단 이동
<code>lcd.print("Hello world...")</code>	<code>"Hello world..."</code> 출력
<code>lcd.setCursor(0, 1)</code>	2번째 줄 첫 칸으로 이동
<code>lcd.print(" i ")</code>	<code>"i"</code> 출력
<code>lcd.printByte(3)</code>	커스텀 문자(하트) 출력
<code>lcd.print(" arduinos!")</code>	<code>" arduinos!"</code> 출력

```
delay(5000)
```

5초 대기

```
displayKeyCodes()
```

LCD 문자 코드셋을 16개씩 차례로  
표시 (무한 반복)

---

## 핵심 함수 - `displayKeyCodes()`

동작

설명

```
lcd.clear()
```

화면 지우기

```
lcd.print("Codes 0x..")
```

현재 문자 코드 범위  
표시

```
lcd.printByte(i+j)
```

코드 값에 해당하는  
문자 출력

```
i+=16
```

다음 16개 문자셋으로  
넘어감

반복

LCD 문자 코드셋 전체  
반복 표시

---

## 사용된 커스텀 문자들 (`createChar`)

번호

모양 설명

배열명

0

 벨

`bell`

1

 음표

`note`

2

 시계

`clock`

3

 하트

`heart`

4

 오리

`duck`

5

 체크


`check`

6

 X표

`cross`

7

 되돌리기

`retarrow`

# Serial Display

---

```
* Displays text sent over the serial port (e.g. from the Serial Monitor) on
* an attached LCD.
* YWROBOT
*Compatible with the Arduino IDE 1.0
*Library version:1.1
*/
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27,20,4); // set the LCD address to 0x27 for a 16
chars and 2 line display

void setup()
{
    lcd.init();           // initialize the lcd
    lcd.backlight();
    Serial.begin(9600);
}

void loop()
{
    // when characters arrive over the serial port...
    if (Serial.available()) {
        // wait a bit for the entire message to arrive
        delay(100);
        // clear the screen
        lcd.clear();
        // read all the available characters
        while (Serial.available() > 0) {
            // display each character to the LCD
            lcd.write(Serial.read());
        }
    }
}
```

# 코드 전체 요약

목적:

PC의 **Serial Monitor**에 입력한 텍스트를 LCD 화면에 그대로 표시한다.

---

## 코드 구성 설명 (구조별)

부분	설명
<code>#include &lt;Wire.h&gt;</code>	I2C 통신용 라이브러리
<code>#include &lt;LiquidCrystal_I2C.h&gt;</code>	I2C 방식 LCD 라이브러리
<code>LiquidCrystal_I2C lcd(0x27, 20, 4);</code>	LCD 주소 0x27, 20칸 X 4줄 (보통 20x4 LCD) → 사실 주석은 "16칸 2줄"이라 잘못됨

---

## setup()

```
lcd.init();           // LCD 초기화
lcd.backlight();      // LCD 백라이트 켜기
Serial.begin(9600);   // PC ↔ Arduino 시리얼 통신 시작 (속도: 9600 bps)
```

- **LCD** 준비 + 백라이트 켜기
  - 시리얼 통신(**Serial Monitor**) 활성화
- 

## loop()

```
if (Serial.available()) { // 시리얼 입력이 있으면
    delay(100);           // 데이터 전송 끝날때까지 잠시 기다림
    (100ms)
    lcd.clear();          // LCD 화면 초기화
```

```
while (Serial.available() > 0) { // 입력 데이터 남아있는 동안 반복
  lcd.write(Serial.read());      // 한 글자씩 읽어서 LCD에 출력
}
}
```

**Serial Monitor**에 입력값이 들어오면

LCD 화면 지우고(**clear**)

→ 입력한 문자열을 LCD 화면에 그대로 표시

# Hello World

```
//YWROBOT
//Compatible with the Arduino IDE 1.0
//Library version:1.1
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27,20,4); // set the LCD address to 0x27 for a 16
chars and 2 line display

void setup()
{
  lcd.init();           // initialize the lcd
  // Print a message to the LCD.
  lcd.backlight();
  lcd.print("Hello, world!");
}

void loop()
{
  lcd.setCursor(0, 1);
  lcd.print(millis() /1000);
}
```

## 코드 목적

LCD 첫 번째 줄에는 "Hello, world!" 출력

LCD 두 번째 줄에는 아두이노 실행 후 경과 시간(초)를 실시간으로 표시

---

## 코드 구조 요약

구분	코드 내용	설명
라이브러리	<pre>#include &lt;Wire.h&gt; #include &lt;LiquidCrystal_I2C.h&gt;</pre>	I2C 통신 및 LCD 라이브러리 포함
LCD 객체 생성	<pre>LiquidCrystal_I2C lcd(0x27,20,4);</pre>	20글자 × 4줄 LCD, I2C 주소 0x27
초기화 (setup)	<pre>lcd.init(); lcd.backlight(); lcd.print("Hello, world!");</pre>	LCD 초기화, 백라이트 켜기, 첫 줄에 "Hello, world!" 출력
반복 실행 (loop)	<pre>lcd.setCursor(0,1); lcd.print(millis()/10 00);</pre>	두 번째 줄 첫 칸에 현재 초단위 시간 표시 (매초 갱신됨)

---

## 실행 시 LCD 화면 예시

```
Hello, world!
0   ← 실행 직후
```

```
Hello, world!
1   ← 1초 후
```

```
Hello, world!
2   ← 2초 후
```

... 계속 증가

— `millis()`는 아두이노가 켜진 후 경과된 밀리초를 반환 → `/1000` 해서 초 단위로 표시