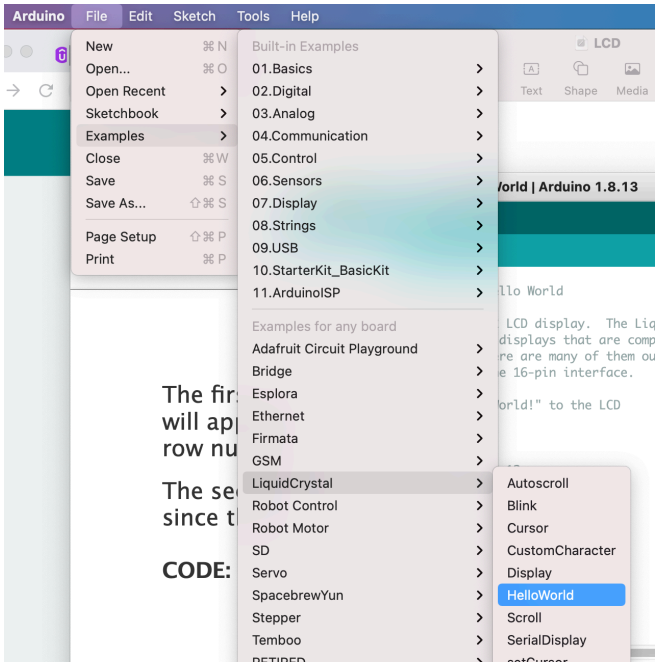
~~The first sets the cursor position (where the next text will appear) to column 0 & row 1. Both column and row numbers start at 0 rather than 1.~~

~~The second line displays the number of milliseconds since the Arduino was reset.~~

~~**CODE: "Hello world" can be found as shown below**~~



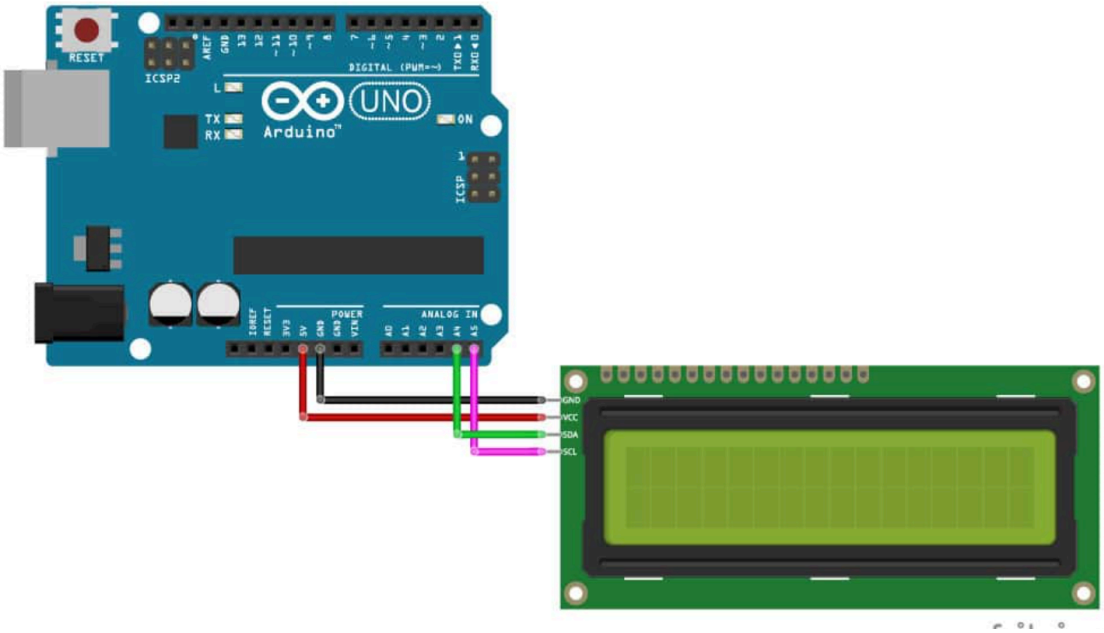
****

I2c LCD

USING I2C LCD(MUCH DIFFERENT):

i2c uses a serial bus instead of the more 'pin consuming' parralell bus system. The wiring will not use a single pin on the Digital side, but rather 2 A

pins from the Analog side, power and ground. You will also need to install the i2c library.



Displaying Text on I2C LCD

```
#include <LiquidCrystal_I2C.h>

int totalColumns = 16;
int totalRows = 2;

LiquidCrystal_I2C lcd(0x27, totalColumns,
totalRows);

void setup(){
lcd.init();
```

```
lcd.backlight(); // use to turn on and turn off
LCD back light
}

void loop()
{
lcd.setCursor(0, 0);
lcd.print("Microcontrollers");
lcd.setCursor(0,1);
lcd.print("I2C LCD tutorial");
delay(1000);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Static text");
delay(1000);
lcd.setCursor(0,1);
lcd.print("I2C LCD tutorial");
delay(1000);
lcd.clear();
}
```

Scrolling text:

```
#include <LiquidCrystal_I2C.h>

int totalColumns = 16;
int totalRows = 2;

LiquidCrystal_I2C lcd(0x27, totalColumns,
totalRows);
```

```
String staticMessage = "I2C LCD Tutorial";
String scrollingMessage = "Welcome to
Microcontrollerslab! This is a scrolling
message.";

void scrollMessage(int row, String message, int
delayTime, int totalColumns) {
  for (int i=0; i < totalColumns; i++) {
    message = " " + message;
  }
  message = message + " ";
  for (int position = 0; position <
message.length(); position++) {
    lcd.setCursor(0, row);
    lcd.print(message.substring(position, position
+ totalColumns));
    delay(delayTime);
  }
}

void setup(){
  lcd.init();
  lcd.backlight();
}

void loop(){
  lcd.setCursor(0, 0);
  lcd.print(staticMessage);
  scrollMessage(1, scrollingMessage, 250,
totalColumns);
}
```

Display Custom Characters on I2C LCD using Arduino

In this section, we will display custom characters on the LCD screen.

For our 16×2 LCD display that we are using, we have the option to display custom characters as well. In this particular LCD, each block consists of 5×8 pixels. These can be used to display custom characters by setting the state of each pixel by inside a byte variable.

There is a very simple way to generate the byte variable of your own custom character. Head over to the following custom character generator: (Custom Character Generator for HD44780 LCD Modules- https://omerk.github.io/lcdchargen/).

Specify the custom character you want to display by clicking the pixels in the 5×8 pixel block and the corresponding byte variable will be generated.

In our case, we will display a '+' character on the screen. This is the byte variable that we will use in our program code to display this particular character on the LCD.

Custom character CODE: (character was created in above pic):

```
#include <LiquidCrystal_I2C.h>

int totalColumns = 16;
int totalRows = 2;

LiquidCrystal_I2C lcd(0x27, totalColumns, totalRows);

byte customChar[8] = {
  0b00000,
  0b00100,
  0b00100,
  0b11111,
  0b00100,
  0b00100,
  0b00000,
  0b00000
};

void setup()
{
  lcd.init();
  lcd.backlight();
  lcd.createChar(0, customChar);
}

void loop()
{
 lcd.setCursor(0, 0);
 lcd.write(0);
}
```