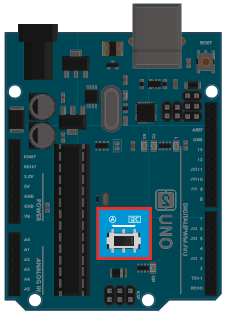
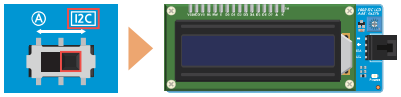
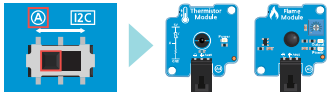


I2C (Inter-Integrated Circuit) is an NFC that can connect a 1 master (Arduino) : multiple slave (sensor modules) in one direction using a SCL (Serial Clock) pin and SDA (Serial Data) pin with full-up resistance connected. Arduino can use SDA, SCL pins as I2C communication pins or analog A4, A5 pins as functions of SDA and SCL respectively. In the starter kit, the SDA and SCL pins of the Uno board were placed for easy selection with the slide switch in the center of the Arduino Uno board.

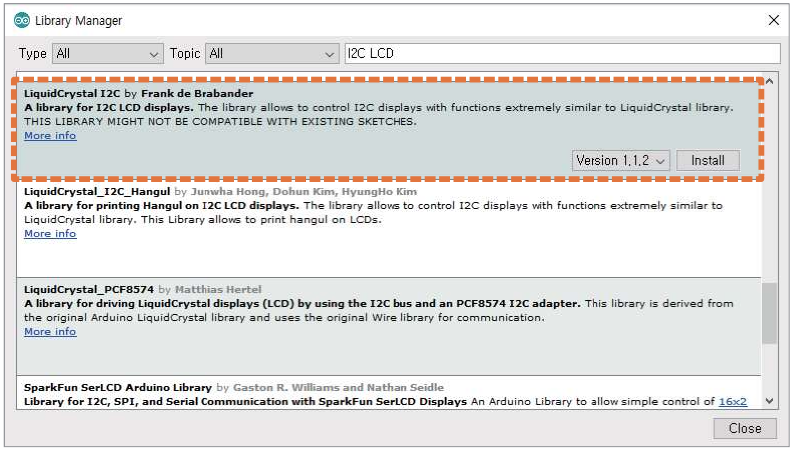
! CAUTION!

You cannot use the I2C communication interface and the A4, A5 pins at the same time on the Arduino board. Therefore, the thermistor module and flame sensor module cannot be used simultaneously with the I2C LCD in the starter kit. Therefore, when using I2C LCD, define the module of use by placing the slide switch left and right as shown in Figure below..





For I2C communication, the Wire library must be added. To add a library, click **Sketch > Include Library > Manage Libraries** to run the Library Manager. Search for **"I2C LCD"** in the search box and install **"LiquidCrystal I2C by Frank de Brabander."**



The default address is 0×27 (hexadecimal value), but sometimes an error occurs when using a module, the address must be checked. Address scanning is recommended first to find the address of the I2C LCD interface module you want to use.

CODING ARRAY
STARTER KIT FOR ARDUINO

CAK Starter Code > 09_01_1602LCD_AddressScanning

```
/* This sketch is designed to use 1602 LCDs using I2C communication.
 * Scans the address to show the results on the serial monitor.
 */

#include <Wire.h>          // Includes the Wire Library for I2C communication.

void setup() {
  Wire.begin();
  Serial.begin(9600);          // Starts serial communication at 9600 speeds
  while(!Serial);              // Wait for the serial monitor.
  Serial.println("\nI2C Scanner");
}

void loop() {
  byte error,address;
  int nDevices;
  Serial.println("Scanning...");
  nDevices =0;

  for (address =1; address <127; address++)  {
    // The i2c_scanner has determined whether the device has approved the address.
    // Use the Write.endTransmission return value to know.
    Wire.beginTransmission(address);
    error =Wire.endTransmission();

    if (error ==0) {
      Serial.print("I2C device found at address 0x");
      if(address<16)
        Serial.print("0");
      Serial.print(address,HEX);
      Serial.println("  !");
      nDevices++;
    }
    else if(error==4) {
      Serial.print("Unknown error at address 0x");
      if (address<16)
        Serial.print("0");
      Serial.println(address,HEX);
    }
  }
  if (nDevices ==0)
    Serial.println("No I2C devices found\n");
  else
    Serial.println("done\n");

  delay(5000);          // Wait five seconds for the next scan..
}
```

Let's learn about the method used by LiquidCrystal_I2C.h..

LiquidCrystal_I2C lcd (0x27,16,2); The method under I2C communication address, 16-cand 2-line lcd object creation is for example if the object was named lcd.

lcd.init();	Initialize LCD.
lcd.backlight();	Turn on the LCD backlight.
lcd.noBacklight();	Turn off the LCD backlight.
lcd.setCursor(Rows);	Sets the position of the cursor.
lcd.print(data, BASE);	Print the text on the LCD screen. Data : Data to be printed, char, byte, int, long, stringable. Characteristic data is displayed in "In" BASE (Optional) : The standard by which a number is printed. BIN (binary), DEC (decimal), OCT (8 decimal), HEX (16 decimal)
lcd.scrollDisplayLeft();	Scroll text and cursors one space to the left
lcd.scrollDisplayRight();	Scroll text and cursors one space to the right
lcd.noDisplay();	urn off the screen. The string on the face disappears, but the contents remain in internal memory. lcd.display(); when a function is called, the string is revived.
lcd.display();	Turn on the screen. lcd.noDisplay(); and lcd.display(); two functions can have a flicker effect on the entire screen.
lcd.autoscroll();	Scroll text and cursors one space to the left
lcd.noAutoscroll();	Scroll text and cursors one space to the right
lcd.clear();	Clear the LCD screen and place the cursor in the upper left corner.
lcd.cursor();	Indicates the underscore of the position in which the following characters are written
lcd.noCursor();	Hide the LCD cursor.
lcd.rightToLeft();	Set the direction of the string used for the LCD from right to left. The default value is left to right. It does not affect previously printed text.
lcd.leftToRight();	Set left to right direction of string written to LCD. It does not affect previously printed text.
lcd.write(data);	Text is written on the LCD.
lcd.createChar(number,data);	Create a custom character to use for LCD. Up to 8 characters 5*8 pixels are supported. Numeric : Numbers from 0 to 7. Specify the number of characters to create. Data: Pixel Data for Text
lcd.noBlink();	Turn off the blinking LCD cursor.
lcd.blink();	Turn on the blinking LCD cursor.
lcd.home();	Position the cursor in the upper left corner of the LCD and output the string.

Learn how to print characters on an LCD using method and how it works..

CAK Starter Code > 09_02_1602LCD_HelloCAK

```
/* This sketch shows Hello! ^_^ ! Coding Array Kit for 1602 LCDs using I2C communication
 * Show the string, using scrollDisplayRight and scrollDisplayLeft methods
 * Scrolls to the left and right.
 * Flashes the screen using noDisplay and discaly method to indicate that one loop is complete.
 */

#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27,16,2);  // Set the LCD I2C address. Use 16-kan2 line LCD.
                                   // Put the scanned address instead of 0x27.
void setup(){
 lcd.init();
 lcd.backlight();              // Turn on the backlight. (lcd.noBacklight() turns off the backlight.)
 lcd.setCursor(0,0);           // first line first column
 lcd.print("Hello ! ^_^ !");
 lcd.setCursor(0,1);           // 2nd line first column
 lcd.print("Coding Array Kit");
 delay(1000);
}

void loop(){
 // Scroll 16 length of string to the left
 for(int positionCounter = 0; positionCounter < 16; positionCounter++) {
  lcd.scrollDisplayLeft();                  // Scroll to the left by one position
  delay(200);                               // for 200 milliseconds
 }

 // String length 16+ Rows 16 Scroll to the right at a total of 32 locations
 for(int positionCounter = 0; positionCounter < 32; positionCounter++) {
  lcd.scrollDisplayRight();                 // Scroll to the left by one position
  delay(200);                               // for 200 milliseconds
 }

 // Scroll to the left 16 positions to center
 for(int positionCounter = 0; positionCounter < 16; positionCounter++) {
  lcd.scrollDisplayLeft();                  // Scroll to the left by one position
  // wait a bit:
  delay(200);                               // for 200 milliseconds
 }

 // noDisplay and dispaly
 lcd.noDisplay();   // Turn off the screen
 delay(500);       // for 0.5 second
 lcd.display();     // Turn on the screen
 delay(500);       // for 0.5 second
}
```

CAK Starter Code > 09_03_1602LCD_Autoscroll

```
/* This sketch shows that 1602 LCD uses I2C communication.
 * Numbers from 0 to 9 are displayed on the screen.
 * Use the autoscroll( ) and noAutoscroll( ) methods.
 * Show how to move all strings left or right.
 */

#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27,16,2);  // Set the LCD I2C address. Use 16 Space 2-line LCD.

void setup(){
 lcd.init();
 lcd.backlight();              // Turn on the backlight. (lcd.noBacklight() turns off the backlight.)
}

void loop() {
 lcd.setCursor(0,0);                    // Position the cursor (0,0) in the upper left corner.

 for(int thisChar = 0; thisChar <10; thisChar++) {
  lcd.print(thisChar);                  // The numbers from 0 to 9 are shown on the LCD.
  delay(500);                           // For 0.5 Second
 }

 lcd.setCursor(16,1);                   // Position the cursor in the lower right hand corner.
 lcd.autoscroll();                      // Set to auto-scroll

 for(int thisChar = 0; thisChar <10; thisChar++) {
  lcd.print(thisChar);                  // The numbers from 0 to 9 are shown on the LCD..
  delay(500);                           // For 0.5 Second
 }

 lcd.noAutoscroll();                    // Automatic Scroll Revocation
 lcd.clear();                           // Clear the screen before going to the next loop.
}
```

CAK Starter Code > 09_04_1602LCD_TextDirection

```
/* This sketch shows that 1602 LCD uses I2C communication.
 * Show one alphabet from a to Z.
 * 'a through l' is written from left to right.
 * 'm through r' is written from right to left
 * 's through z' causes text to appear from left to right again.
 */

#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27,16,2);  // Set the LCD I2C address. Use 16 Space 2-line LCD.

int thisChar ='a';                          // thisChar Save

void setup(){
 lcd.init();
 lcd.backlight();            // Turn on the backlight. (lcd.noBacklight() turns off the backlight.)
 lcd.cursor();               // Turn on the cursor.
}

void loop(){
 if(thisChar =='m') {              // If thisChar is m, change direction.
  lcd.rightToLeft();                  // From the next letter, mark to the left.
 }

 if(thisChar =='s') {              // If thisChar is s
  lcd.leftToRight();                  // From the next letter, mark to the left.
 }

 if(thisChar >'z') {               // When thisChar is out of z,
  lcd.home();                         // Go to (0,0)
  thisChar ='a';                      // Restart from the beginning
 }

 lcd.write(thisChar);              // Indicate thisChar value on LCD.
 delay(1000);                      // for a second
 thisChar++;                       // increase thisChar value one by one
}
```

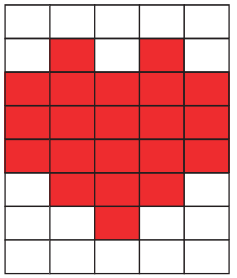

CAK Starter Code > 09_05_1602LCD_CustomCharacters

```
/* This sketch shows that 1602 LCD uses I2C communication.
 * Set up a special character or Figure to "I □ Array Kit !" and smile on the first line of the LCD.
 * In the second row, the shape of a person who raises and lowers his or her arms is displayed.
 * Learn how to indicate that the value of A0 variable resistance is constant.
 */

#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27,16,2);  // Set the LCD I2C address. Use 16 Space 2-line LCD.

// Create special characters
byte heart[8] ={
  0b00000,
  0b01010,
  0b11111,
  0b11111,
  0b11111,
  0b01110,
  0b00100,
  0b00000
};

byte smiley[8] ={
  0b00000,
  0b00000,
  0b01010,
  0b00000,
  0b00000,
  0b10001,
  0b01110,
  0b00000
};

byte frownie[8] ={
  0b00000,
  0b00000,
  0b01010,
  0b00000,
  0b00000,
```

```
  0b00000,
  0b01110,
  0b10001
};

byte armsDown[8] ={
  0b00100,
  0b01010,
  0b00100,
  0b00100,
  0b01110,
  0b10101,
  0b00100,
  0b01010
};

byte armsUp[8] ={
  0b00100,
  0b01010,
  0b00100,
  0b10101,
  0b01110,
  0b00100,
  0b00100,
  0b01010
};

void setup(){
  lcd.init();
  lcd.backlight();          // Turn on the backlight. (lcd.noBacklight() turns off the backlight.)

  // Define New Characters
  lcd.createChar(0,heart);
  lcd.createChar(1,smiley);
  lcd.createChar(2,frownie);
  lcd.createChar(3,armsDown);
  lcd.createChar(4,armsUp);

  lcd.setCursor(0,0);          // first line first column
  // Write text to LCD
  lcd.print("I ");
  lcd.write(byte(0));                  // Use a heart that is stored at 0 bytes..
  lcd.print(" Array Kit! ");
  lcd.write((byte)1);                  // Use a smiley stored in one byte..
}

void loop(){
  int sensorReading = analogRead(A0);      // Read the variable resistance value of A0.
  int delayTime = map(sensorReading,0,1023,200,1000);  // Map resistance values from 200 to 1000
  lcd.setCursor(4,1);                  // Set the cursor to the bottom 5th position
  lcd.write(3);                    // Draw a person with his arm down on number 3.
  delay(delayTime);                    // Delay by variable resistance value
  lcd.setCursor(4,1);                  // Set the cursor to the bottom 5th position
  lcd.write(4);                    // Draw the person with the arm up stored in number 4.
  delay(delayTime);                    // Delay by variable resistance value
}
```

byte Save the 8-bit signless number from 0 to 255

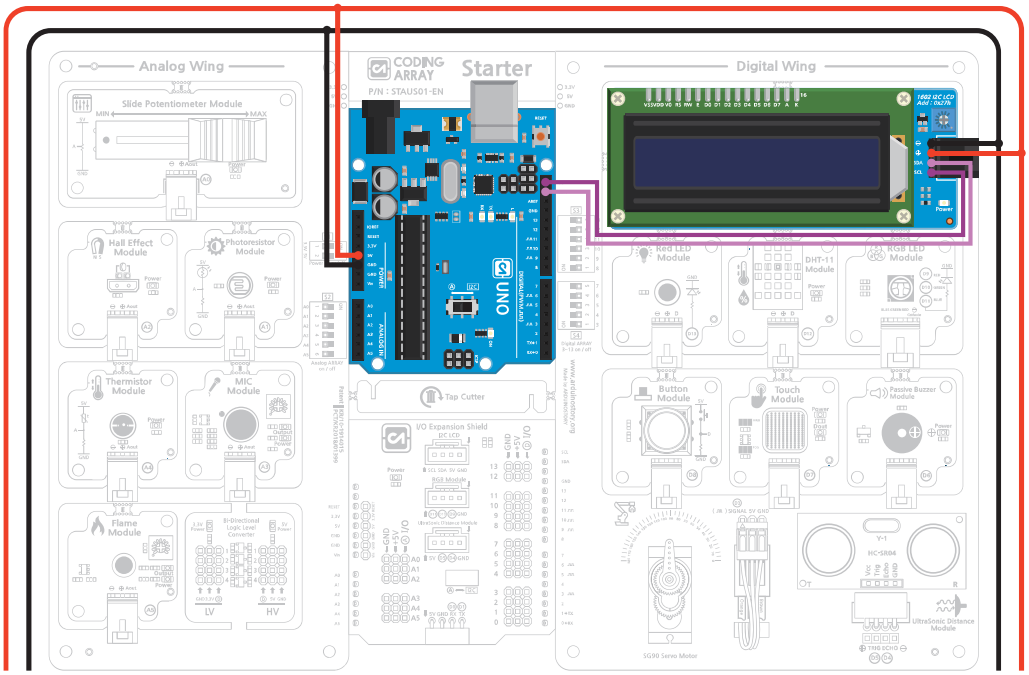
The shape of each user-defined character consists of 5×8 dots. Each row is specified in an array of eight bytes, one by one, and each row is defined as a hexadecimal. Assuming that you make a heart, you can set it as follows:



00000	0b00000
01010	0b01010
11111	0b11111
11111	0b11111
11111	0b11111
01110	0b01110
00100	0b00100
00000	0b00000

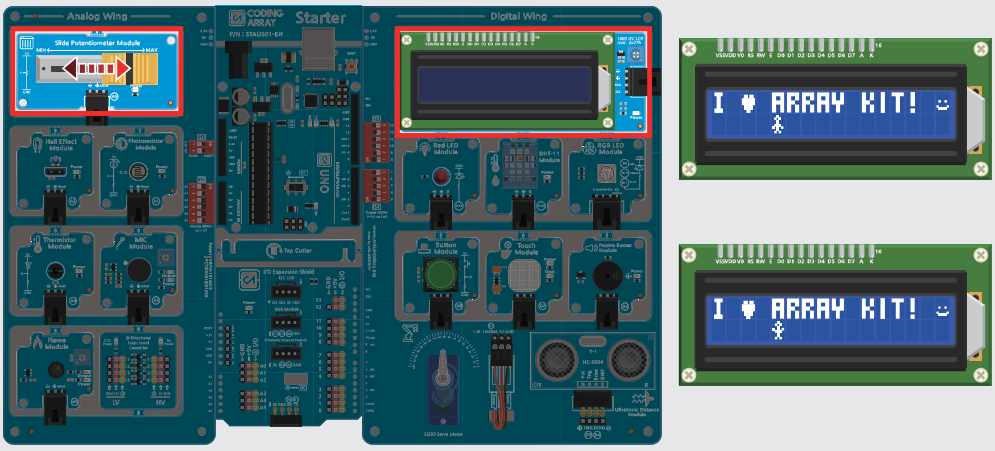
Special characters specified in the array are defined as lcd.createChar (number, data). The numbers can then be defined by a total of eight characters from 0 to 7, and the data can be named after the array. lcd to output user-defined characters to LCD.use a number.

CODING ARRAY CIRCUIT



Print out characters using the I2C interface module, which controls 1602 LCDs consisting of 2 rows wide and 16 spaces with SDA (Serial Data, A4) and SCL (Serial Clock, A5), as well as GND, and 5 volt total..

View Results



You can adjust the rate of special character change in the second row by adjusting the variable resistance of the slide

CAK Starter Code > 09_06_I2C_1602LCD_SerialDisplay

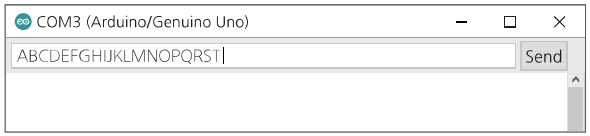
```
/*  This sketch shows that 1602 LCD uses I2C communication.
    Try printing the entered characters in the serial window.
*/

#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2); // Set the LCD I2C address. Use 16 Space 2-line LCD.

void setup() {
  lcd.init();
  lcd.backlight();              // Turn on the backlight. (lcd.noBacklight() turns off the backlight.)

  Serial.begin(9600);                   // Starts serial communication at 9600 speed
}

void loop() {
  if (Serial.available()) {   // When the text arrives on the serial communication,
    delay(100);               // Wait 0.1 seconds for the entire message to arrive.
    lcd.clear();              // Clear the screen.
    while (Serial.available() > 0) { // while the text is coming in
      lcd.write(Serial.read());       // Write the characters you read on the LCD.
    }
  }
}
```

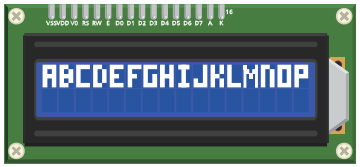
Serial.available(); returns the number of data when received by serial communication.
Returns zero if no data has been received.

Serial.read(); to read data entering the serial by 1 byte and return it to the decimal (constant) ASCII code value. Returns -1 if the receive buffer is empty.

lcd.write(**Serial.**read()); data entered into the serial are passed in aske code numbers.
It is represented by converting it to a letter using lcd.write.
If you write lcd.print, the number of Aski codes will be output.

View Results



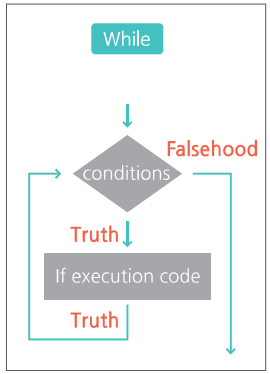


When you enter and transfer data in the serial window, the data is output on the first line of the LCD. If you enter more than 16 data, only 16 will appear in the first line..

Let's find out about While Sentence.

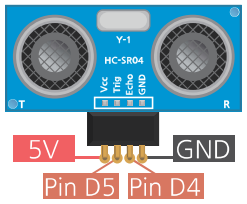
while {execution code; }

'While' is one of the repeats used to give the same command over and over, like 'for'. 'While statement' is a structure that gives a certain repeat condition and repeats the execution code while satisfying the condition. For statement lists the multiplication table under the iterative conditions, but in the whle statement, the multiplication ceremony is placed in the execution code..



10 Distance measurement with ultrasonic sensor

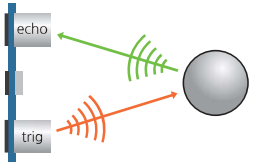
CHAPTER 2

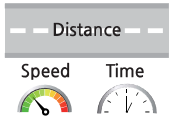


Ultrasonic Distance Module

Ultrasonic is a type of sound wave, which is the sound wave of a frequency (>20KHz) area that is higher than the area that a person can hear. Ultrasonic sensors can calculate the distance using the time it takes for sound waves to return to an obstacle that is close to 3-400 cm..

Ultrasonic sensors have two speakers that look like speakers. One side has to produce ultrasonic waves with digital output HIGH and then stops ultrasonic waves with LOW. Ultrasonic sensors used in the starter kit will use a pulse width of 10 μs and therefore maintain the HIGH for 10 μs. The other is the role (Echo) of detecting ultrasonic waves reflected on an object. To detect the return of ultrasound, make sure that the reflector and the ultrasonic sensor are at right angles





Distance, velocity and time form the following formular

$$\text{velocity} = \frac{\text{Distance}}{\text{time}}, \quad \text{time} = \frac{\text{Distance}}{\text{velocity}}, \quad \text{Distance} = \text{velocity} \times \text{time}$$

Ultrasonic waves can be generated from the trig pin of the ultrasonic sensor and obtained by means of the reciprocating distance, reciprocating time and sound velocity reflected on the barrier..

$$Round-trip\ distance = Sound\ speed \times travel\ time$$

$$\therefore Distance\ to\ object = \frac{Sound\ speed \times the\ time\ it\ takes\ for\ a\ microwave\ to\ return}{2}$$

In Arduino, the time it takes for the ultrasound to return is in microseconds (), so the distance can be obtained in cm after the unit conversion..

$$\therefore Distance\ to\ object = \frac{Sound\ speed \times the\ time\ it\ takes\ for\ a\ microwave\ to\ return(\mu s)}{2}$$

$$= \frac{1}{2} \times \frac{340\ m}{1\ s} \times \frac{100\ cm}{1\ m} \times the\ time\ it\ takes\ for\ a\ microwave\ to\ return\ \mu s \times \frac{1\ s}{1000000\ \mu s}$$

$$= 0.017 \times the\ time\ it\ takes\ for\ a\ microwave\ to\ return(cm)$$

At this point, the speed of sound is about 340 m/s at 15° C, and as the temperature rises by 1° C, the speed of 0.6 m/s increases. Therefore, it may be used with the following corrections:.

Sound speed=340+0.6×(current temperature-15)

Using the principles of ultrasonic sensors, one can measure a key or measure a distance to an obstacle in front of one. It is also possible to implement a parking system by sensing that the parking space is empty or parked, such as a parking lot in a large shopping mall. Ultrasonic waves are also used to examine the condition of the human organs or to identify deep undersea terrain, as the density of the medium that transmits sound is higher. In vacuum, there is no medium, so distance measurements using ultrasonic waves are not allowed..

CODING ARRAY
STARTER KIT FOR ARDUINO

CAK Starter Code > 10_Ultrasonic_Distance

```
/*  This sketch uses ultrasonic sensors to measure the distance.
 *  The trigger pin is connected to Arduino's number 5. The trigger pin produces ultrasonic waves.
 *  Echo pins are connected to Arduino's number 4. The echo pin detects reflected ultrasound.
 *  The measured distance should be shown in both the serial and LCD windows.
 *  If the measured distance exceeds the set value, the green LED,
 *  If the measured distance exceeds the set value, turn on the red LED.
 */

#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27,16,2);  // Set the LCD I2C address. Use 16 Space 2-line LCD

int redLED = 13;                // Red LED No. 13
int greenPin=10;                // Green LED No. 13
int threshold = 15;             // Set Distance Threshold Value

void setup() {
 pinMode(5,OUTPUT);     // Trigger pin connection No. 5
 pinMode(4,INPUT);      // Echopin to Pin 4

  pinMode(redLED,OUTPUT);     // Set pin 13 to output
  pinMode(greenPin, OUTPUT);   // Set pin 10 to output

  Serial.begin(9600);          // Starts serial communication at 9600 speeds

  // LCD Setting
  lcd.init();
  lcd.backlight();              // Turn on the backlight. (lcd.noBacklight() turns off the backlight.)
  lcd.clear();
}

void loop() {
//  Distance measurement with ultrasonic sensor
  float Duration, Distance;
  digitalWrite(5, HIGH);                          // Fire an ultrasound.
  delayMicroseconds(10);                          // for 10 microseconds
  digitalWrite(5, LOW);                           // Turn off the ultrasound.
  Duration =pulseIn(4, HIGH);                     // Save the time the ehofin is held in HIGH
  Distance =((float)(340 *Duration) /10000) /2;           // convert the distance to cm

// Measured Distance Output
  Serial.print(Distance);                 // Print distance in serial window without changing lines
  Serial.println("cm ");                  // unit output

  if (Distance < threshold){          // If the measurement distance is less than the threshold value, turn on the red LED.
    digitalWrite(redLED,HIGH);
    digitalWrite(greenPin,LOW);
    // Show Distance to LCD Window
    lcd.clear();
    lcd.setCursor(0,0);             // first line first column
    lcd.print(Distance);            // Measured Distance Output
    lcd.print("[cm]");   // unit output
  }
  else {                        // If the measured value is greater than the threshold value, turn on the green LED.
    digitalWrite(redLED,LOW);
    digitalWrite(greenPin,HIGH);
    // Show Distance to LCD Window
    lcd.clear();
    lcd.setCursor(0,0);             // first line first column
    lcd.print(Distance);            // Measured Distance Output
    lcd.print("[cm]");   // unit output
  }
  delay(2000);                  // 2000 millisecond delay to reliably read values
}
```

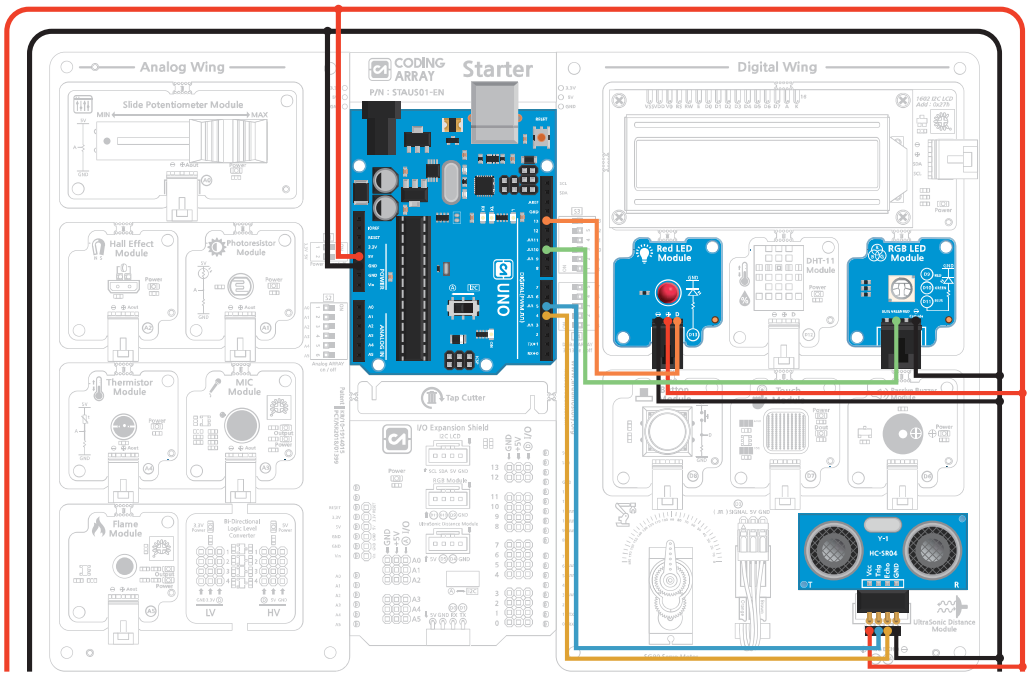
float Duration, Distance; A variable with the same type of data can be declared at once..

pulseIn (Pin number, Value, timeout); Measure the amount of time it takes to return after a pulse occurs.

Pin number : pin number to read the pulse
Value : Type of pulse to read. HIGH or LOW
timeout (optional) : The time (microseconds) to wait for the pulse to start, and the length of the pulse (unshinged long) in microseconds.
Returns zero if the pulse does not start within the specified timeout.
It can operate from 10 microseconds to 3 minutes.
pulseIn (echoPin, HIGH); when the value of echoPin reaches HIGH, start the timer and return the time that HIGH is maintained.

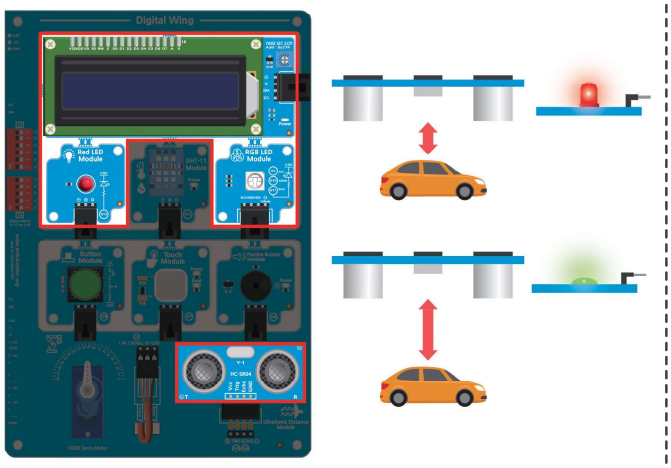
((float)(340 *Duration) /10000) /2; Since the calculation result is a true type with a decimal point, attach the data type as float.

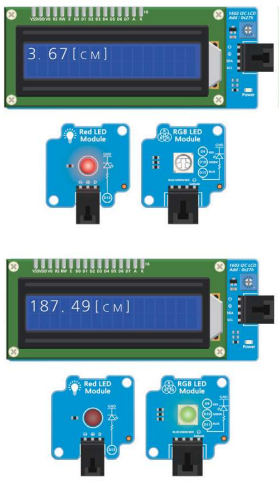
CODING ARRAY CIRCUIT



The trigger that produces ultrasonic waves is connected to digital No. 5 and the echo that detects reflected ultrasonic waves is connected to digital No. 4. Using the principles of ultrasonic sensors, measure the distance to the obstacle in front.. .

View Results





Once the program is uploaded, you can see that the green LED illuminates when the obstacle is more than 15cm apart in front of the ultrasonic sensor, and that the red LED turns on when the obstacle is less than 15cm away.