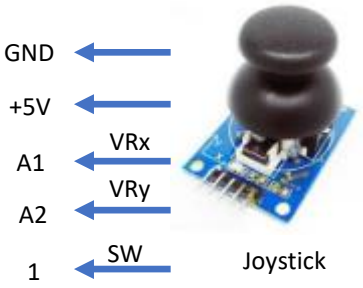
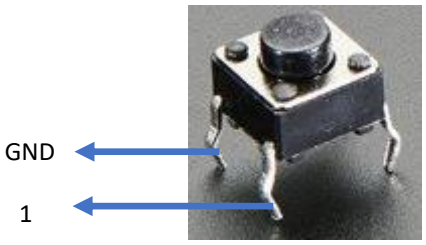
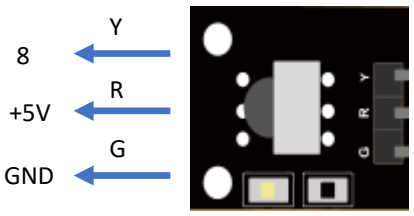


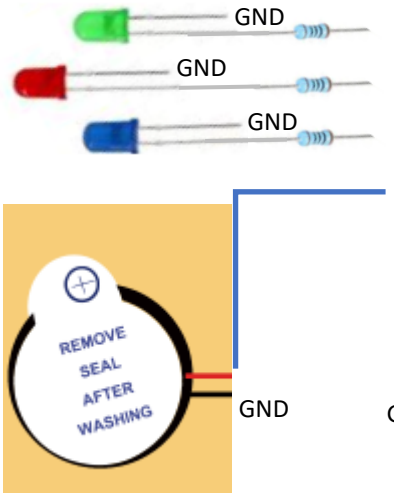
Joystick

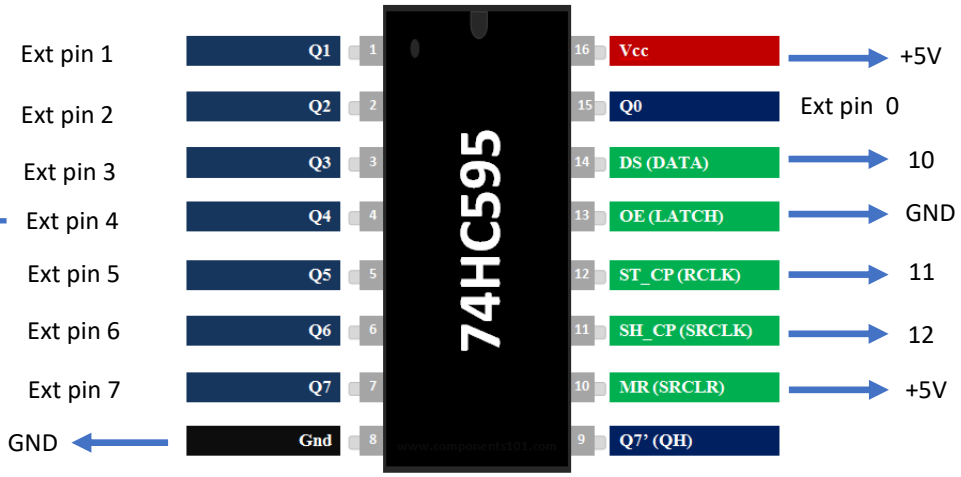


Push button

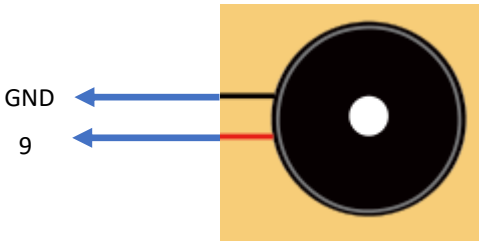


IR receiver

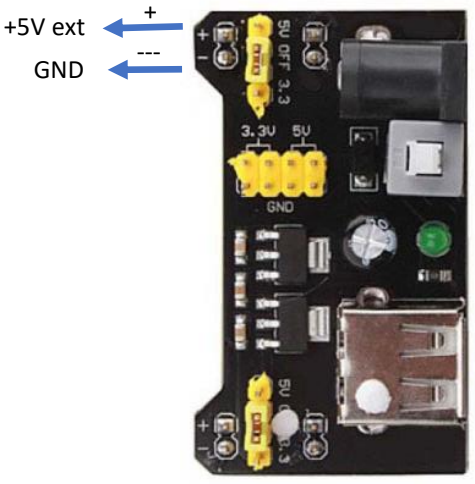




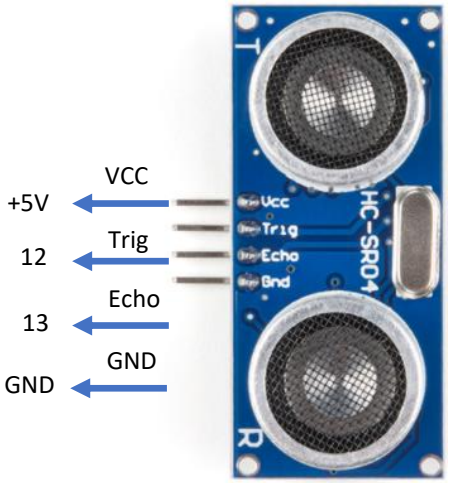
Transforms 3 ports into 8 binary ports (outputs)



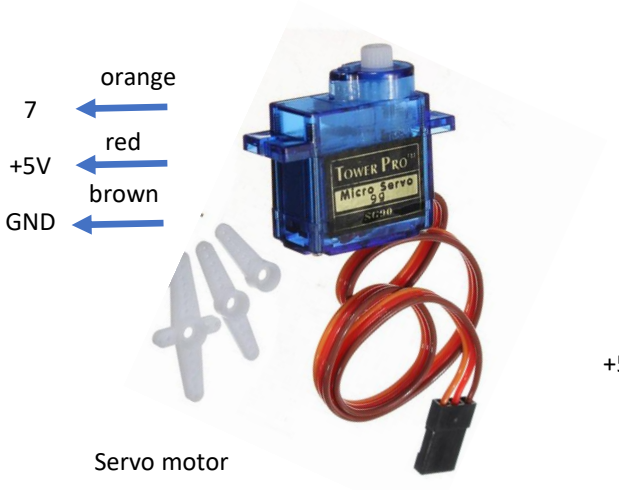
Passive buzzer



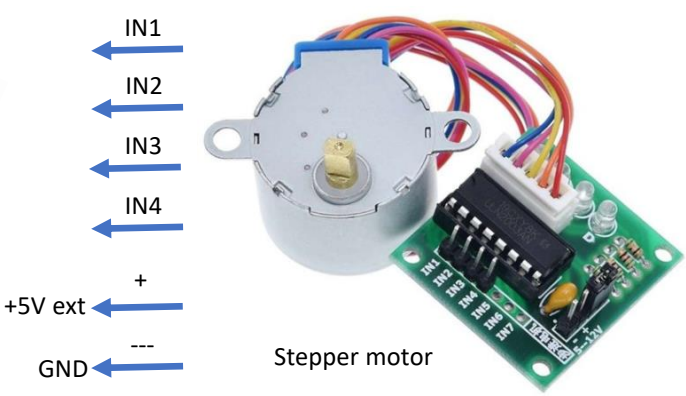
Breaboard Power Supply Module



Ultrasonic sensor HC-SR04



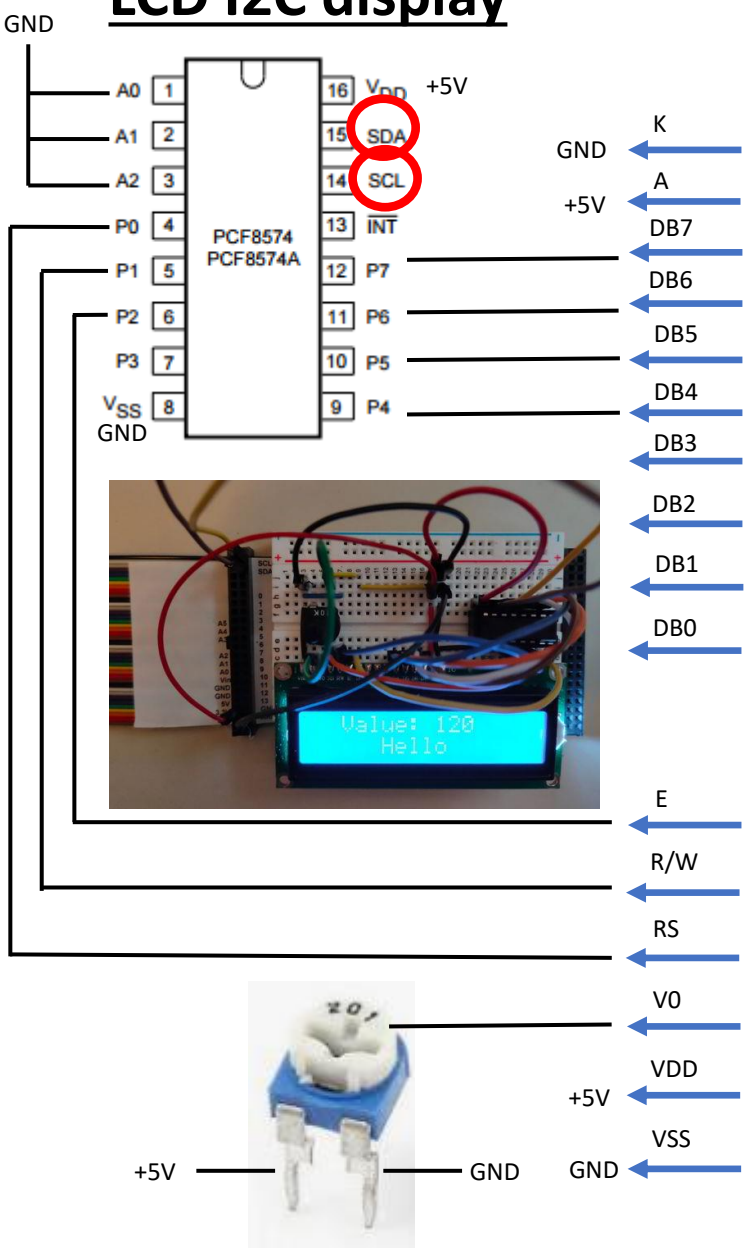
Servo motor

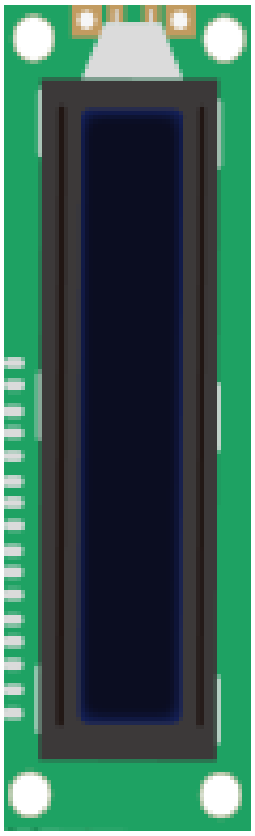


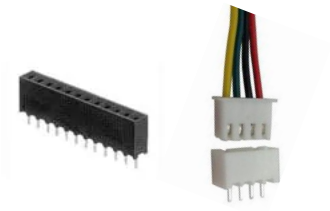
Stepper motor

LCD I2C display

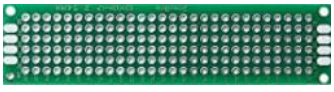
I2C port 0x38
Library: LiquidCrystal_PCF8574
LCD LCM1602/HD44780



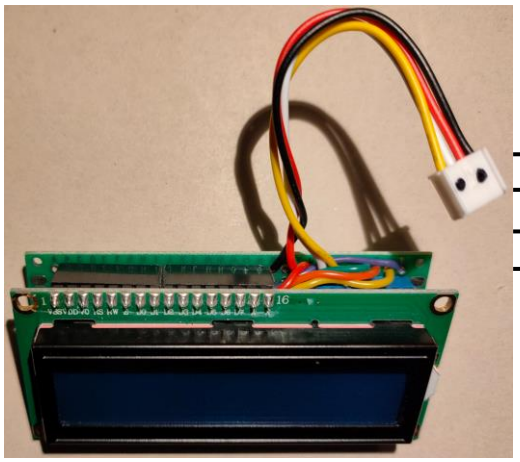




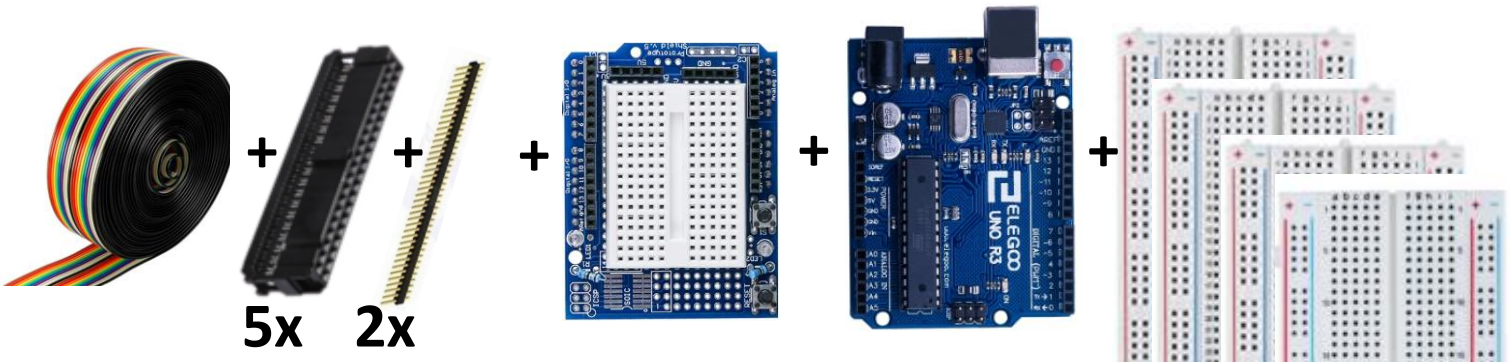
PCB connectors

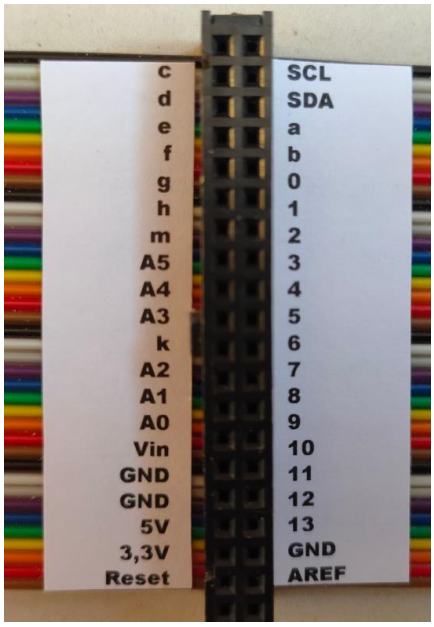


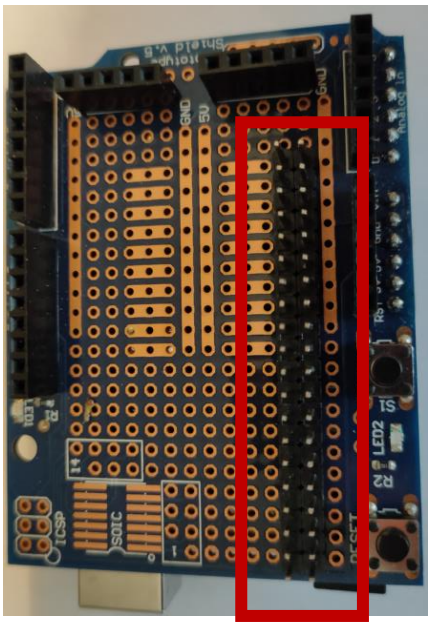
ELEGOO Prototype board 2x8cm

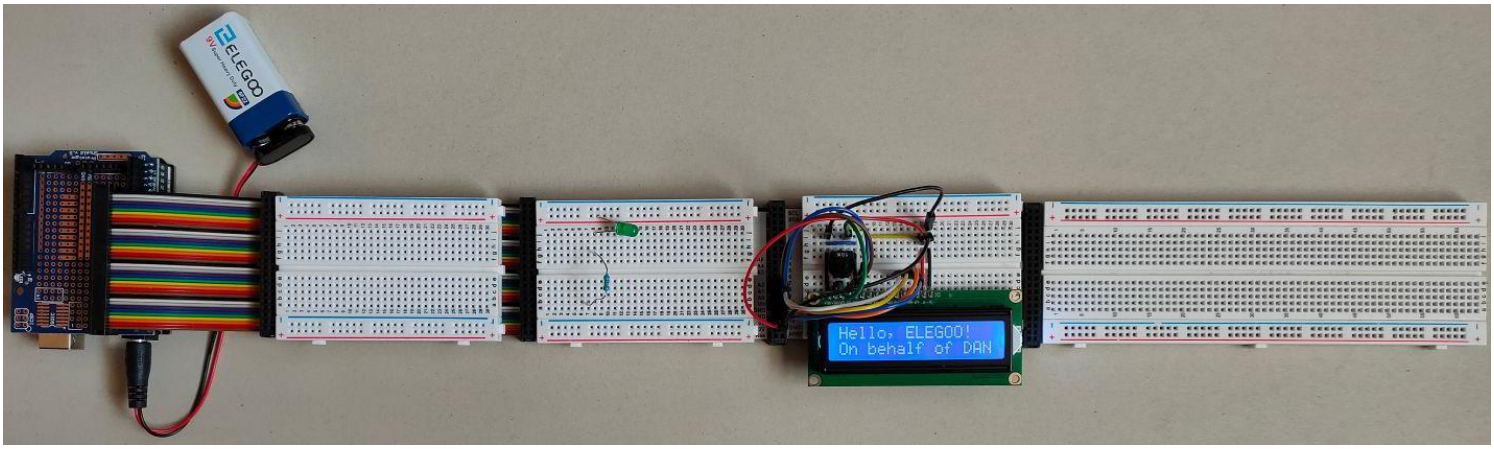


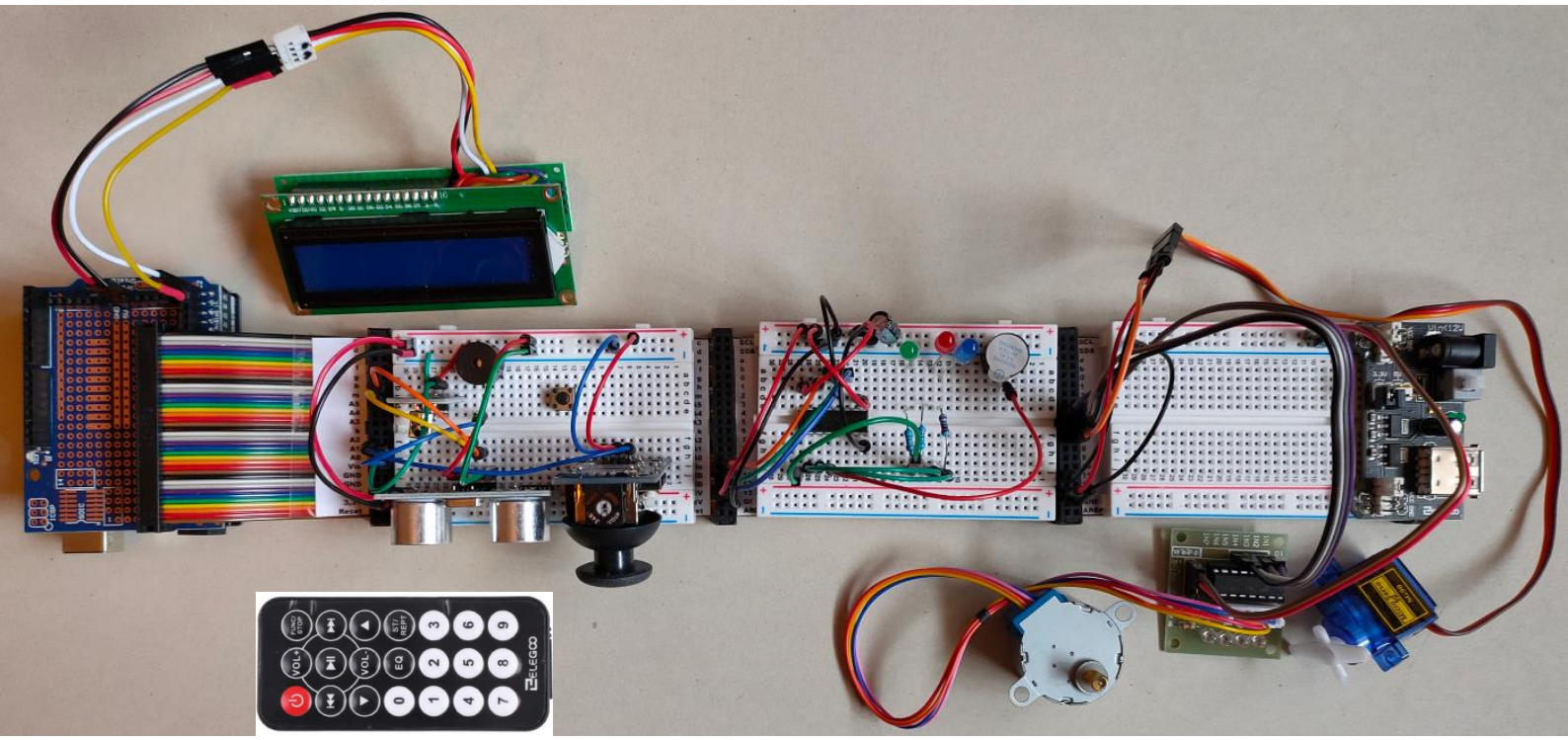
Circuit PCF8574 reduces active UNO connections from 6 to 2

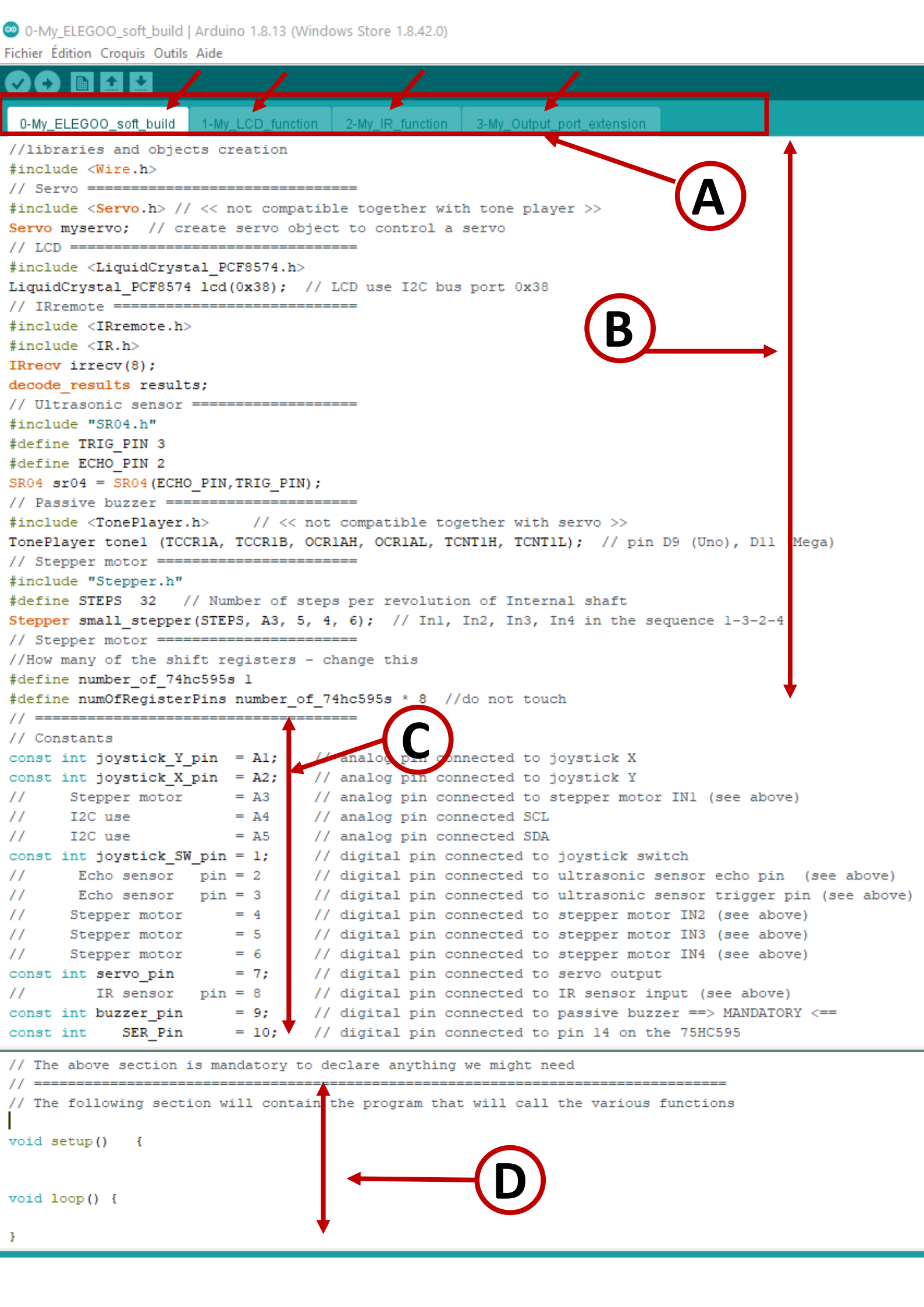

5x
2x
ELEGOO
UNO R3


c SCL
d SDA
e a
f b
g 0
h 1
m 2
A5 3
A4 4
A3 5
k 6
A2 7
A1 8
A0 9
Vin 10
GND 11
GND 12
5V 13
3,3V GND
Reset AREF


GND
5V
ICSP
SOIC
RESET


ELEGOO
Hello, ELEGOO!
On behalf of DAN


VOL-
VOL+
EQ
ST/REPT
0
1
2
3
4
5
6
7
8
9
ELEGOO


0-My_ELEGOO_soft_build | Arduino 1.8.13 (Windows Store 1.8.42.0)
Fichier Édition Croquis Outils Aide
0-My_ELEGOO_soft_build
1-My_LCD_function
2-My_IR_function
3-My_Output_port_extension
A
B
C
D
//libraries and objects creation
#include <Wire.h>
// Servo ================================
#include <Servo.h> // << not compatible together with tone player >>
Servo myservo; // create servo object to control a servo
// LCD ==================================
#include <LiquidCrystal_PCF8574.h>
LiquidCrystal_PCF8574 lcd(0x38); // LCD use I2C bus port 0x38
// IRremote =============================
#include <IRremote.h>
#include <IR.h>
IRrecv irrecv(8);
decode_results results;
// Ultrasonic sensor ====================
#include "SR04.h"
#define TRIG_PIN 3
#define ECHO_PIN 2
SR04 sr04 = SR04(ECHO_PIN,TRIG_PIN);
// Passive buzzer =======================
#include <TonePlayer.h>   // << not compatible together with servo >>
TonePlayer tone1 (TCCR1A, TCCR1B, OCR1AH, OCR1AL, TCNT1H, TCNT1L); // pin D9 (Uno), D11 Mega)
// Stepper motor ========================
#include "Stepper.h"
#define STEPS 32   // Number of steps per revolution of Internal shaft
Stepper small_stepper(STEPS, A3, 5, 4, 6); // In1, In2, In3, In4 in the sequence 1-3-2-4
// Stepper motor ========================
//How many of the shift registers - change this
#define number_of_74hc595s 1
#define numOfRegisterPins number_of_74hc595s * 8 //do not touch
// ======================================
// Constants
const int joystick_Y_pin = A1;   // analog pin connected to joystick X
const int joystick_X_pin = A2;   // analog pin connected to joystick Y
//   Stepper motor   = A3   // analog pin connected to stepper motor IN1 (see above)
//   I2C use   = A4   // analog pin connected SCL
//   I2C use   = A5   // analog pin connected SDA
const int joystick_SW_pin = 1;   // digital pin connected to joystick switch
//   Echo sensor   pin = 2   // digital pin connected to ultrasonic sensor echo pin (see above)
//   Echo sensor   pin = 3   // digital pin connected to ultrasonic sensor trigger pin (see above)
//   Stepper motor   = 4   // digital pin connected to stepper motor IN2 (see above)
//   Stepper motor   = 5   // digital pin connected to stepper motor IN3 (see above)
//   Stepper motor   = 6   // digital pin connected to stepper motor IN4 (see above)
const int servo_pin   = 7;   // digital pin connected to servo output
//   IR sensor   pin = 8   // digital pin connected to IR sensor input (see above)
const int buzzer_pin   = 9;   // digital pin connected to passive buzzer ==> MANDATORY <==
const int   SER_Pin   = 10;   // digital pin connected to pin 14 on the 75HC595
// The above section is mandatory to declare anything we might need
// ==
// The following section will contain the program that will call the various functions
void setup()   {
void loop() {
}