Sheffield Hallam University
School of Engineering and Built Environment

Activity ID	Activity Title	Laboratory Room No.	Level
Lab 4	Building systems	4302	Intermediate

Equipment (per student/group)

Number	Item
1	PC
1	Arduino kit

Learning Outcomes

	Learning Outcome
3	Develop good laboratory practice and demonstrate knowledge, understanding and ability to safely use relevant materials, equipment, tools, processes or products.
4	Demonstrate competent use of prototyping, modelling and basic analytical techniques.

Building systems

Introduction

Computing is about more than the PC on your desktop! Embedded devices are everywhere – from wireless telecommunications infrastructure points to electronic point of sale terminals. One definition of an embedded system is:

> "An embedded system is a computer system designed to perform one or a few dedicated functions often with real-time computing constraints."
>
> (http://en.wikipedia.org/wiki/Embedded_system)

In this series of labs you are going to be introduced to the open source Arduino platform – a cheap, simple to program, well documented prototyping platform for designing electronic systems. The heart of this prototyping system is the Arduino Uno microcontroller board itself which is based on the commonly used ATMega328 chip.

So far in the laboratory sessions we have provided a fairly gentle introduction to both the software and hardware aspects of the Arduino platform, some of the basic electronics you'll need to create functional embedded systems, and input / output with Arduino microcontrollers. In this laboratory session we will look at putting these concepts together to build actual systems.

Bibliography

There are no essential bibliographic resources for this laboratory session aside from this tutorial sheet. However the following websites and tutorials may be of help (especially if you haven't done much electronics previously or your digital logic and/or programming is a bit rusty):

- http://www.arduino.cc/
- http://www.ladyada.net/learn/arduino/index.html
- http://tronixstuff.wordpress.com/tutorials/

Methodology

Check that you have all the necessary equipment (see Figure 1)!

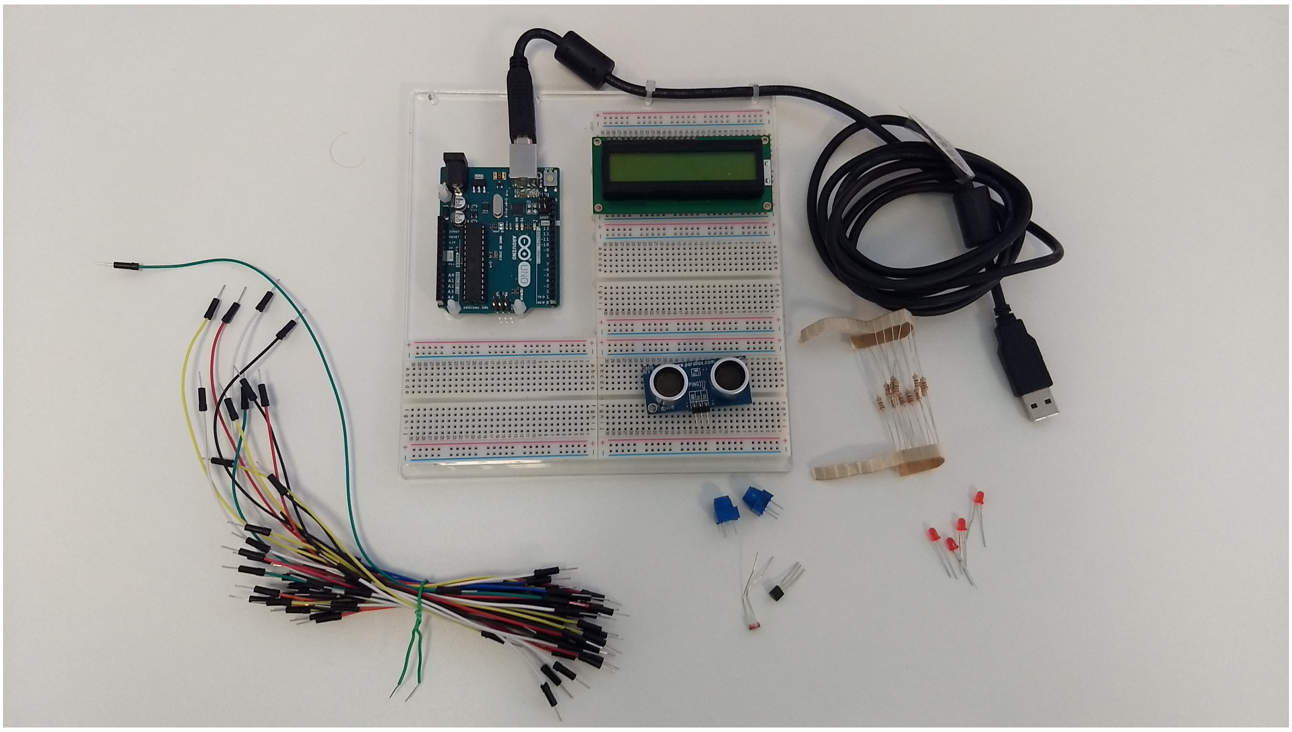

Figure 1 – The necessary equipment for this lab (don't worry if you have LEDs of a different colour or slightly different breadboards)

Task 1

In this task we are going to wire up the LCD display unit to the Arduino via our breadboard so that we can display messages. In these examples we will be using the LiquidCrystal library for Arduino that allows us to control LCD displays that are compatible with the Hitachi HD44780 driver (this covers the majority of LCD displays that you will encounter).

Firstly we need to know some stuff about LCDs. The LCD displays we are using in these laboratory sessions use a 16 pin parallel interface (which means that the Arduino has to alter the state of multiple interface pins simultaneously to control the display). These pins are:

LCD Pin number	**Symbol**	**Function**
1	Vss	Display power to ground
2	Vdd	Display power to +5V
3	Vo	Contrast voltage (grounding it sets it to maximum contrast)
4	RS	Register select
5	R/W	Read / Write Selector (grounding it sets it permanently to write)
6	E	Enable strobe
7	D0	Data pin 0
8	D1	Data pin 1
9	D2	Data pin 2
10	D3	Data pin 3
11	D4	Data pin 4
12	D5	Data pin 5
13	D6	Data pin 6
14	D7	Data pin 7
15	A	Display backlight power to +5V
16	K	Display backlight power to ground

Table 1: LCD pins

Now build the circuit shown in Figure 2, below. This has data pins from the Arduino connected to RS, E, D4, D5, D6, and D7. We are not connecting D0-D3 in this circuit.

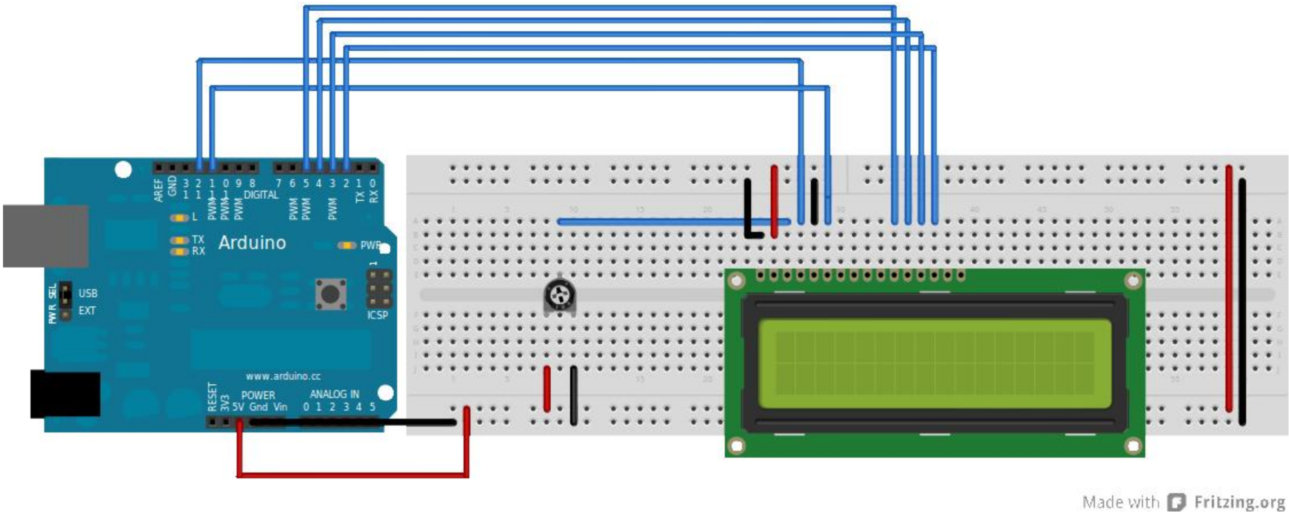

Figure 2 – The wired up LCD circuit

Q1 Note the use of a potentiometer connected to V_o – based on the information in Table 1, what do you think this does?

Now we need to make the LCD display actually do something! An example hello LCD world program is shown in code listing 1.

Code listing 1: Hello LCD World!

```
/*
 * 1_lcd_hello_world.ino
 *
 * this is just a simple sketch to test our lcd display
 *
 * author: prof. alex shenfield
 * date:   2024/11/12
 */

// include the lcd library
#include <LiquidCrystal.h>

// initialise the lcd display
// pin inputs are rs, enable, d4, d5, d6, d7
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

// CODE

// set everything up
void setup()
{
  // setup the lcd display (16x2 lines)
  lcd.begin(16, 2);

  // print a message
  lcd.setCursor(0, 0);
  lcd.print("lcd hello world!");
}

// main program loop
void loop()
{
  // set the lcd cursor to be the first element of the second row
  lcd.setCursor(0, 1);

  // print the time on the second row of the lcd
  lcd.print(millis() / 1000);
}
```

Task 2

In this task we are going to use light dependent resistors to obtain important information about the state of our environment (i.e. how dark it is!). Figure 3 shows one of the variants of light dependent resistor stocked by the university.

Figure 3 – An LDR

You are required to integrate this LDR into your "hello LCD world" circuit (see Figure 2) and display the results on the LCD display.

You might notice that the LDR only has two legs – this means that, like the pushbutton switch, you will need a pull up or pull down resistor in your circuit.

You should also rescale the values of your LDR to be a percentage. To do this you will need to know the maximum and minimum values output by the LDR.

Q1 Devise a way of finding this out:

Now write your program!

Task 3

We are now going to build an ultrasonic tape measure that will use an ultrasonic sensor to read distance to an accuracy of 1cm and display it on the LCD screen. The completed system is shown in Figure 4.

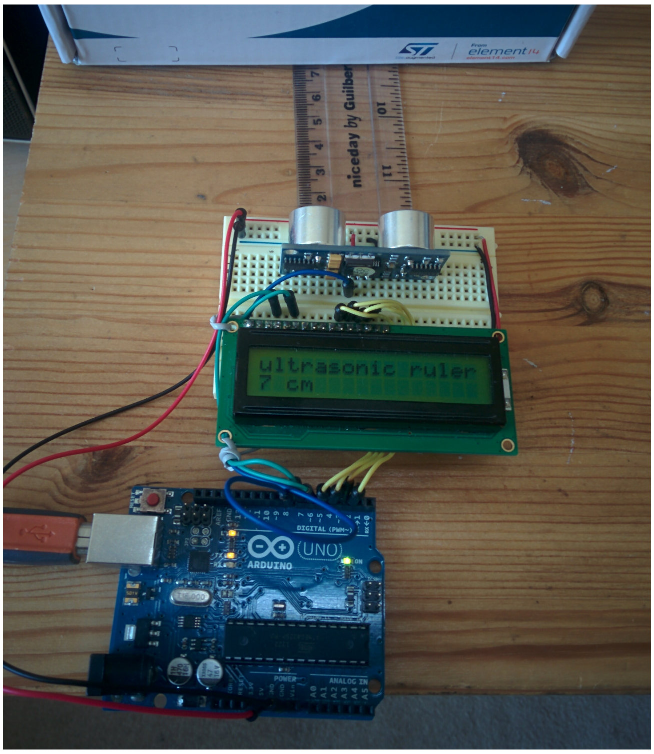

Figure 4 – An ultrasonic tape measure

Figure 5 shows a bread board schematic layout of this system (where the wiring can be seen with greater clarity).

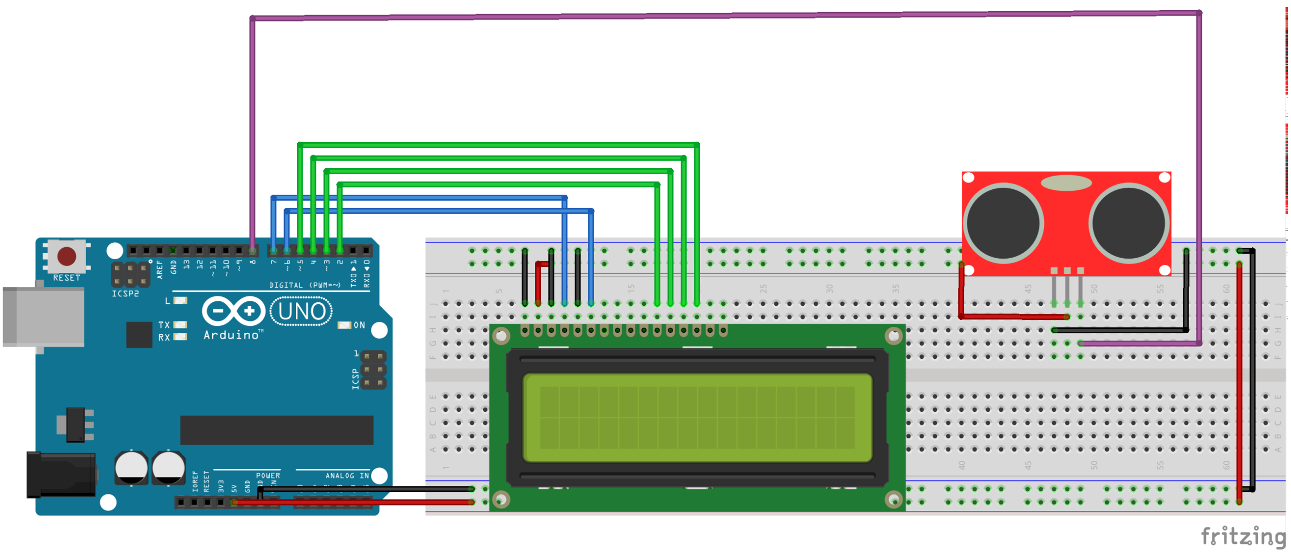

Figure 5 – Schematic of the ultrasonic ruler system

These ultrasonic sensors send a pulse of high frequency sound (well above the range of human hearing) and then listen for the returning echo pulse. The time that this takes can then be converted into a distance using the speed of sound. The easiest way of doing this is by using a purpose built library, and we will be using the NewPing library by Tim Eckel.

As with the installation of the other libraries in previous lab sessions (see lab 2), we can search for and install this via the library manager in the Arduino IDE.

The complete listing for this application is shown in code listing 2, below.

Code listing 2: Ultrasonic tape measure code

```
/*
 * 3_ultrasonic_tape_measure.ino
 *
 * A simple application to use the newping library and the parallax ping)))
 * sensor to measure the distance from the sensor to an object.
 *
 * author: prof. alex shenfield
 * date:   2024/11/12
 */

// include some libraries
#include <NewPing.h>
#include <LiquidCrystal.h>

// define the range of our sensor (in cm)
#define MAX_DISTANCE 200

// create our sensor object (on pin 8)
NewPing sonar(8, 8, MAX_DISTANCE);

// initialise the lcd display
// pin inputs are rs, enable, d4, d5, d6, d7
LiquidCrystal lcd(7, 6, 2, 3, 4, 5);

// CODE

// set everything up
void setup()
{
  // start serial comms
  Serial.begin(9600);

  // setup the lcd display (16x2 lines)
  lcd.begin(16, 2);

  // print a message
  lcd.setCursor(0, 0);
  lcd.print("ultrasonic ruler");
  delay(500);
}
```

```
// main program loop
void loop()
{
  // wait half a second between measurements
  delay(500);

  // send the ping, listen for a response, and get the distance in cm
  int distance = sonar.ping_cm();

  // print to the serial monitor for debugging
  Serial.print(distance);
  Serial.print(" cm\n");

  // set the lcd cursor to be the first element of the second row
  lcd.setCursor(0,1);

  // clear the line, reset the cursor position, and write the new data to it
  lcd.print("                ");
  lcd.setCursor(0,1);
  lcd.print(distance);
  lcd.print(" cm");
}
```

Enter this code and test it works. Use a ruler to test how accurate the measurements are.

Check list

Task 1 – Program	
Task 2 – Circuit	
Task 2 – Program	
Task 3 – Program	

Feedback/ reflections