Recap: Last week - Using the IDE program, plus first sketch - Blink. - any Questions.

!!!NOTE – the NEW_TAB icon (in the IDE) lets you build large projects.

This Week – Using an LCD (Liquid Crystal Display), and reading the push buttons.

Background: An LCD (or Liquid Crystal Display) is a display device or screen.
Used in laptop computers, TVs, and many household appliances (Oven; Microwaves etc)
They use very little power, have good contrast and vibrant colours.
They are made from up to 5 layers.
We do not need to know how they are made or how they work – that is for the engineers.
We only need to know how to use them.

Types: All LCDs for your Arduino, are basically the same. They are packaged in different ways.
They come in sizes – eg 16 characters x 2 rows and 20 x 4; and in different colours.
1. The standard LCD has 16 pin connections. We only need to use about 8 of these.
   We could use ribbon cable, but running 8 or 9 connections can be messy.
2. They also make a special board called a SHIELD, with the LCD on top.
   An LCD shield, comes pre-wired and simply plugs on top of your Arduino.
   Advantage – very simple to use – almost plug and play;
   Disadvantage – it is prewired so you are restricted to using the pins the board uses.
3. An LCD can also have a simpler connection called I2C. I2C has only 4 pins. (+ve; -ve + 2 data). You can mount the board away from your Arduino. NICE!!
   You can plug the 4 wire cable into an I2C socket on a Sensor Shield. Simple.

Shield: An LCD shield simply plugs on top of your Arduino. (See photos on last page).
It includes a set of push buttons. You can display messages on the LCD to ask the user to press a button, and have your Arduino detect if/when the button is pressed.
You only need 4 steps in the code to make the LCD work. These are explained next.

LCD Code: To write the message “Hello World” on your LCD screen.

```c
#include <LiquidCrystal.h>           // step 1 - This is the library needed.

/* The LCD library has an “object” called LiquidCrystal. To use that object we must create our own copy or “instance”, and give it a name. We will call ours - lcd.
This “object” is connected to 6 Arduino pins, in a special order. We must include the pin numbers. For the LCD shield, these are pins 8,9,4,5,6,7 - in that order.
*/

LiquidCrystal  lcd( 8, 9, 4, 5, 6, 7); // step 2 – create our “instance” of LiquidCrystal
   // and declare the pins used by the lcd display

void setup()
{
    lcd.begin(16, 2);    // step 3 - Begin the lcd and declare the size.
                          // eg 16 = number of characters; 2 = number of lines.
    lcd.print("Hello, World!");    // step 4 - print to the lcd. [ in setup() or loop() ]
} // end of setup

void loop()
{
    // your main loop code goes here
} // end of loop
```
**LCD Commands**

The LCD library has a set of commands you can use. They include:

- `lcd.begin();`  // you must include the number of Columns and Rows.
- `lcd.print();`  // prints what is inside the ( ). Put text in inverted commas “ “.
- `lcd.println();`  // prints as above, then moves to the next line.
- `lcd.clear();`  // Clears the LCD AND repositions the cursor to the start.
- `lcd.home();`  // Repositions the cursor to the start. It does NOT clear the LCD.
- `lcd.setCursor();`  // moves the cursor to column and row number – starts from 0,0
- `lcd.blink();`

A full list of LCD commands is at the Arduino webpage – Reference; or Google Arduino LCD.

**!!! Project** – to print the time elapsed since the Arduino was turned on (or reset):

In the loop() section, set the cursor (use `lcd.setCursor`) at the start of the second line (0,1) then `lcd.print("Time elapsed = ");`  // recall `lcd.print` stays on the same line.
- `lcd.print(millis() / 1000);`  // `millis()` - an Arduino function to calculate milliseonds.

**Buttons**

The LCD Shield includes 5 buttons you can use/read as input.

The buttons are arranged in a handy pattern - Select, Left, Up, Down and Right.

It's totally up to you to decide what to do when any button is pressed.

**!!! NOTE** - two types of input/output. Digital (= 0 or 1) and Analog (0,1,2,...1023).

All the buttons are connected to a single analog input - pin A0, using a chain of resistors that causes a different voltage on Arduino pin A0, for each button pressed.

When you `analogRead(A0)` you get different values, depending on which button is pressed.

**!!NOTE** – the exact values depend on the manufacturer of the shield. See code below.

This diagram shows the input buttons and associated resistors:

!!!NOTE - You do not need to know this – it is only to explain how it works.
LCD plus Button Code:  There are various ways to write code to read the buttons.
Here is one example. (Google the web for other examples).

// This code reads the value on Arduino pin A0, when a button on the LCD shield is pressed,
// and prints the value to your PC monitor.

#include <LiquidCrystal.h> // This is the library needed.
LiquidCrystal lcd(8, 9, 4, 5, 6, 7); // Declare the pins used by the LCD

void setup()
{
    lcd.begin(16,2); // begin the lcd and declare the size
    Serial.begin(9600); // begin the serial communication – to print to your monitor
    pinMode(A0, INPUT); // define pin A0 as an INPUT pin.
} // end of setup()

void loop()
{
    Serial.println(analogRead(A0)); // print to your monitor, the value read from pin A0
    delay(2000); // delay for 2 seconds
} // end of loop()

!!! PROJECT – Amend the code, to print the value of A0, on your LCD screen.

Steps – 1. Above setup(), create an integer type variable to hold the value -
   int   lcd_button =0;
2. In loop() - read the A0 pin, and store the value in your variable
   lcd_button = analogRead(A0);
3. Set the cursor at the start of line 2 of your lcd -
   lcd.setCursor(0,1);
4. Print a message, and the value
   lcd.print(" A0 = "); // this explains which Arduino pin we are reading
   lcd.print(lcd_button); // This prints the value we have read from pin A0.
5. delay(2000); // a delay so the screen does not flicker.

!!! NOTE – why are there still 4 numbers displayed, when you press a button?
Solve this by adding one line before lcd.setCursor(0,1) -
lcd.clear(); // clear the lcd display.

!!! CONGRATULATIONS - you are now able to

1. Print messages and values to your PC monitor
2. Print messages and values to an LCD screen
3. Read values from an analog input device (eg the LCD buttons).

Next week, we look at how to use the value from A0 (when a button is pressed), to take action. We will look at the IF statement, the ELSE statement and the CASE statement. These are called CONDITIONAL statements – because they take action based on a “condition”.
For example, if lcd_button < 100, we know the RIGHT button was pressed.
ELSE - if lcd_button < 200 we know the UP button was pressed
ELSE - if lcd_button < 400 we know the DOWN button was pressed
.....etc.

!!! NOTE – the order of the IF and ELSE. If would not work, if we tested for 400 first. WHY?

Our Arduino tutorials are on our U3A Website - here.
BARE LCD.

LCD SHIELD

LCD with I2C connector.

LCD TFT Shield