CRYSTAL BALL

Create a crystal ball to tell your future

Discover : LCD displays, switch/case statements, random()

Time : 1 hour

Level : ■■■■■

Builds on projects : 1,2,3

Crystal ball can help "predict" the future. You ask a question to the all-knowing ball, and turn it over to reveal the answer. The answers will be predetermined, but you can write in anything you like. You'll use your Arduino to choose from a total of 8 responses. The tilt switch in your kit will help replicate the motion of shaking the ball for answers.

The LCD can be used to display alphanumeric characters. The one in your kit has 16 columns and 2 rows, for a total of 32 characters. There are a large number of connections on the board. The pins are used for power and communication, so it knows what to write on screen, but you won't need to connect all of them. See *Fig.1* for the pins you need to connect.

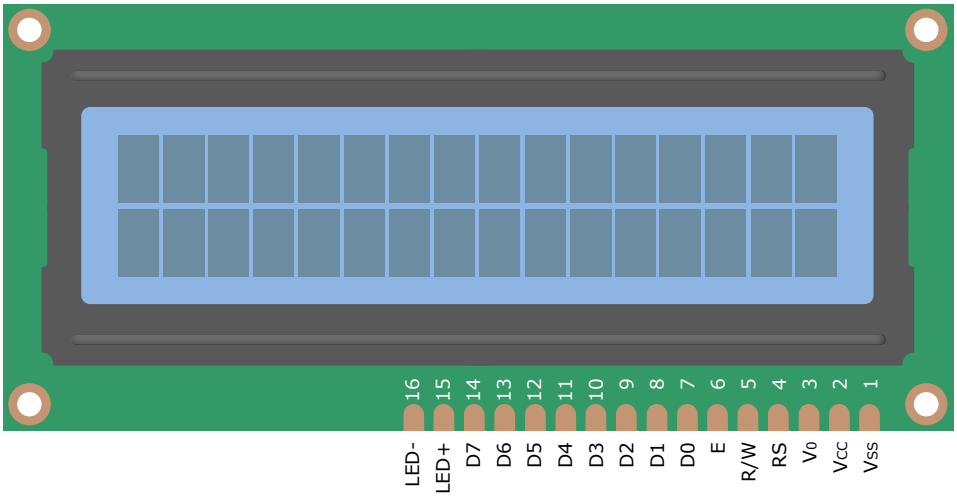
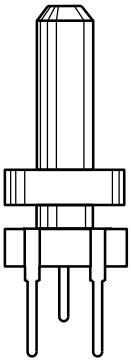
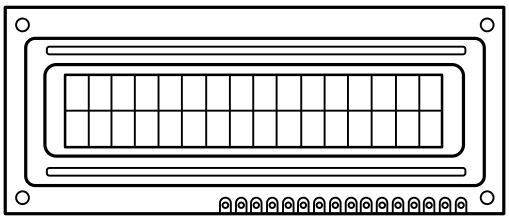


Fig. 1 - The pins on the LCD screen that are used in the project and labels

INGREDIENTS

				
RESISTOR 10 kΩ	RESISTOR 220 Ω	POTENTIOMETER	TILT SWITCH	LCD SCREEN
x2	x1	x1	x1	x1

BUILD THE CIRCUIT

Fig.2 - [Circuit]

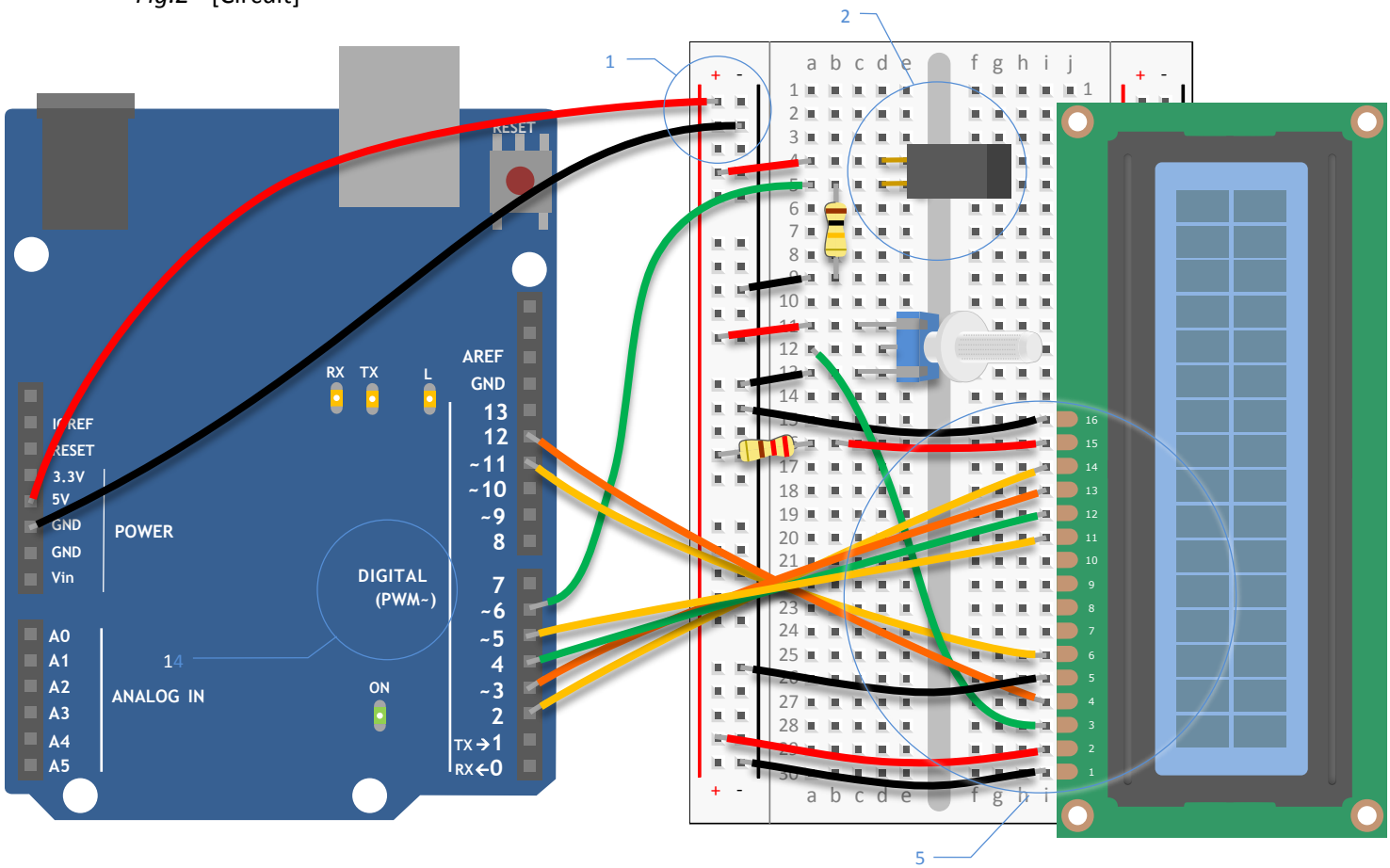
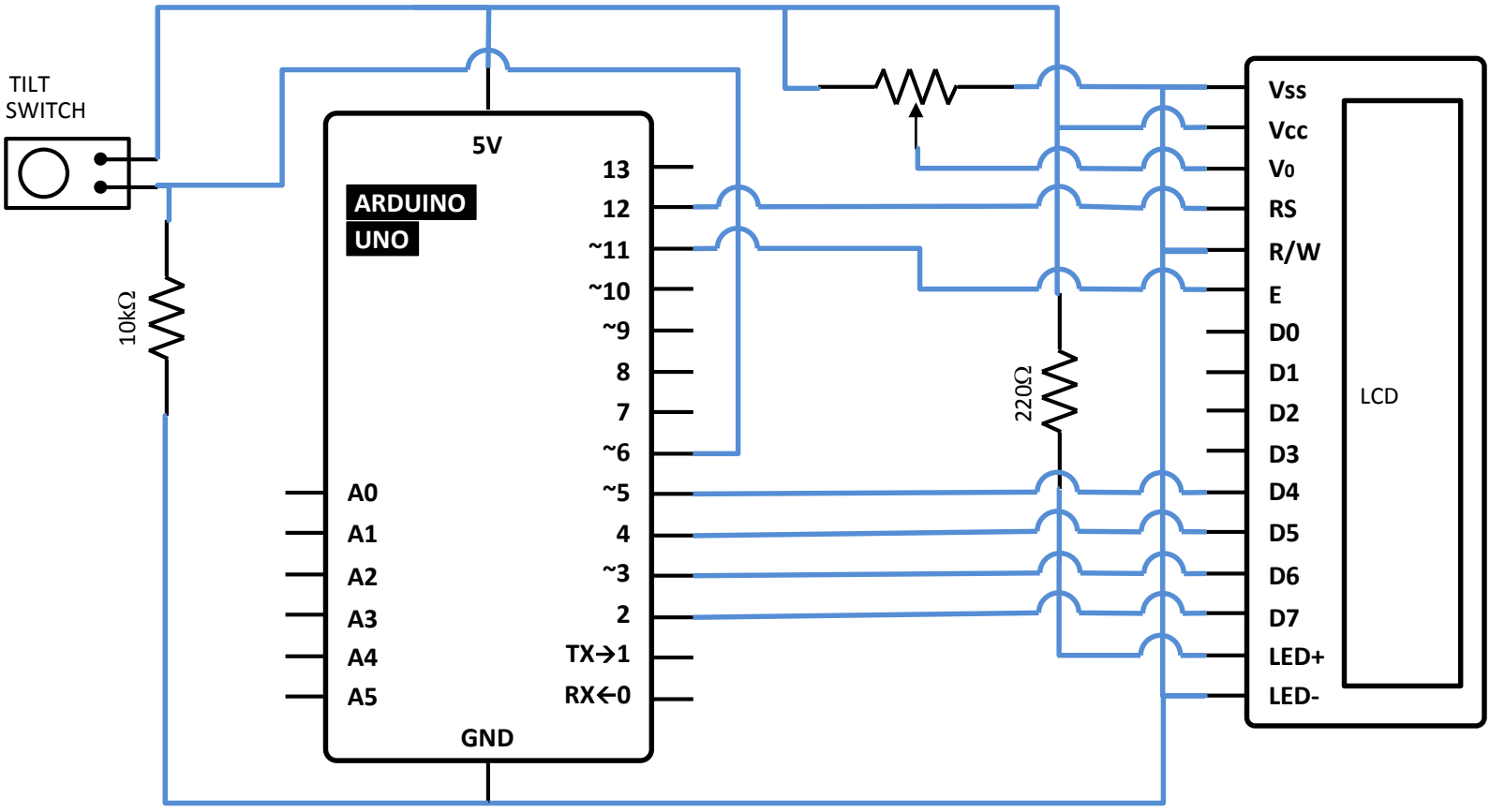


Fig.3 - [Schematic]



In this schematic the LCD pins arrangement does not match the physical order depicted in Fig. 2. In a schematic, the pins are arranged by logical grouping to make the schematic as clear as possible. This is a little confusing to newcomers until you get used to it.

The circuit is not overly complex, but there are a lot of wires. Pay attention when wiring everything up to make sure it's correct.

1) Connect power and ground to one side of your breadboard.

2) Place the tilt switch on the breadboard and attach one lead to 5V. Attach the other side to ground through a 10-kilohm resistor, and to your Arduino's pin 6. You're wiring this as a digital input, just as you've done in several other projects.

3) The register select (**RS**) pin controls where the characters will appear on screen. The read/write pin (**R/W**) puts the screen in read or write mode. You'll be using the write mode in this project. The enable (**EN**) tells the LCD that it will be receiving a command. The data pins (**D0-D7**) are used to send character data to the screen. You'll only be using 4 of these (**D4-D7**). Finally, there's a connection for adjusting the contrast of the display. You'll use a potentiometer to control this.

4) The LiquidCrystal library that comes with the Arduino software handles all the writing to these pins, and simplifies the process of writing software to display characters.
The two outside pins of the LCD (**Vss** and LED -) need to be connected to the ground. Also, connect the R/W pin to ground. This places the screen in write mode. The LCD power supply (**Vcc**) should connect directly to 5V. The LED+ pin on the screen connects to power through a 220-omh resistor.

5) Connect : Arduino Digital pin 2 to LCD **D7**, Arduino Digital pin 3 to LCD **D6**, Arduino Digital pin 4 to LCD **D5**, Arduino Digital pin 5 to LCD **D4**. These are the data pins that tell the screen what character to display.

6) Connect **EN** on the screen to pin 11 on your Arduino. **RS** on the LCD connects to pin 12. This pin enables writing on the LCD.

7) Place the potentiometer on the breadboard, connecting one end pin to power and the other to ground. The center pin should connect to **V0** on the LCD. This will allow you to change the contrast of the screen.

THE CODE

Set up the LiquidCrystal library

First, you'll need to import the **`LiquidCrystal`** library.

Next, you'll initialize the library, somewhat similar to the way you did with the **`Servo`** library, telling it what pins it will be using to communicate.

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
```

Now that you've set up the library, it's time to create some variables and constants. Create a constant to hold the pin of the switch pin, a variable for the current state of the switch, a variable for the previous state of the switch, and one more to choose which reply the screen will show.

```
const int switchPin = 6;
int switchState     = 0;
int prevSwitchState = 0;
int reply           = 0;
```

Print your first line

Set up the switch pin as an input with **pinMode()** in your **setup()**. Start the LCD library, and tell how large the screen is.

```
void setup() {
       lcd.begin(16, 2);
       pinMode(switchPin, INPUT);
```

LCD library reference : http://www.arduino.cc/lcdlibrary

Move the cursor

Now it's time to write a small introductory screen welcoming you to the 8-ball. The **print()** function writes to the LCD screen.

You're going to write the words "Ask the" on the top line of the screen. The cursor is automatically at the beginning of the top line.

```
       lcd.print("Ask the");
```

In order to write to the next line, you'll have to tell the screen where to move the cursor. The coordinates of the first column on the second line are 0,1 (recall that computers are zero indexed. 0,0 is the first column of the first row). Use the function **lcd.setCursor()** to move the cursor to the proper place, and tell it to write "Crystal Ball!".

```
       lcd.setCursor(0, 1);
       lcd.print("Crystal Ball!");
}
```

Now, when you start the program, it will say "Ask the Crystal Ball!" on the screen.

In the **loop()**, you're going to check the switch first, and put the value in the **switchState** variable.

```
void loop() {
       switchState = digitalRead(switchPin);
```

Choose a random answer

Use an **if()** statement to determine if the switch is in a different position than it was previously. If it is different than it was before, and it is currently LOW, then it's time to choose a random reply.

The **random()** function returns a number based on the argument you provide it. To start, you'll have a total number of 8 different responses for the ball. Whenever the statement **random(8)** is called, it will give a number between 0-7. Store that number in your **reply** variable.

```
	if (switchState != prevSwitchState) {
		if (switchState == LOW) {
			reply = random(8);
```

Random reference : http://www.arduino.cc/random

Clear the screen with the function **lcd.clear()**. This also moves the cursor back at location 0,0; the first column in the first row of the LCD. Print out the line "The ball says:" and move the cursor for the output.

```
			lcd.clear();
			lcd.setCursor(0, 0); // Already done by the lcd.clear() function
			lcd.print("The ball says:");
			lcd.setCursor(0, 1);
```

Predict the future

The **switch()** statement executes different pieces of code depending on the value you give it. Each of these different pieces of code is called a **case**. **switch()** checks the value of the variable **reply**; whatever value **reply** holds will determine what named case statement is executed.

Inside the **case** statements the code will be same, but the messages will be different. For example, in case 0 the code says **lcd.print("Yes")**. After the **lcd.print()** function, there's another command : **break**. It tells the Arduino where the end of the case is. When it hits **break**, it skips to the end of the switch statement. You'll be creating a total of 8 case statements to start out. Four of the responses will be positive, 2 will be negative, and the final 2 ask you to try again.

```
			switch(reply) {
				case 0:
					lcd.print("Yes");
					break;
				case 1:
					lcd.print("Most likely");
					break;
				case 2:
					lcd.print("Certainly");
					break;
				case 3:
					lcd.print("Outlook good");
					break;
				case 4:
					lcd.print("Unsure");
					break;
				case 5:
					lcd.print("Ask again");
					break;
				case 6:
					lcd.print("Doubtful");
					break;
				case 7:
					lcd.print("No");
					break;
			}
		}
	}
```

The last thing to do in your **loop()** is to assign **switchState**'s value to the variable **prevSwitchState**. This enables you to track changes in the switch the next time the loop runs.

```
        prevSwitchState = switchState;
}
```

Switch Case reference : http://www.arduino.cc/switchcase

USE IT

To use the magic ball, power the Arduino. Check the screen to make sure it says "Ask the Crystal ball". If you can't see the characters, try turning the potentiometer. It will adjust the contrast of the screen.

Ask a question of your crystal ball, and try tilting the switch upside down and back again. You should get an answer to your question. If the answer doesn't suit to you, ask again.

Try adding your own sayings to the **print()** statements, but be mindful of the fact that there are only 16 characters to use per line. You can also try adding more responses. Make sure when you add additional switch cases, you adjust the number of options that will randomly populate the **reply** value.

LCD work by changing the electrical properties of a liquid sandwiched between polarized glass. The glass only allows certain kinds of light to pass through. When the liquid between the glass is charged, it starts to form into a semi-solid state. This new state runs in a different direction than the polarized glass, blocking light from passing through, thus creating the characters you see on the screen.

The functions covered here for changing the LCD screens text are fairly simple. Once you have a handle on how it works, look at some of the other functions the library has. Try getting text to scroll, or continually update. To find out more about how the **LiquidCrystal** library works, visit: http://www.arduino.cc/lcd

*An LCD display enables you to show text on a screen, using the **LiquidCrystal** library. With a switch…case statements control the flow of programs by comparing a variable to specified values.*