11. LCD

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. These 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.

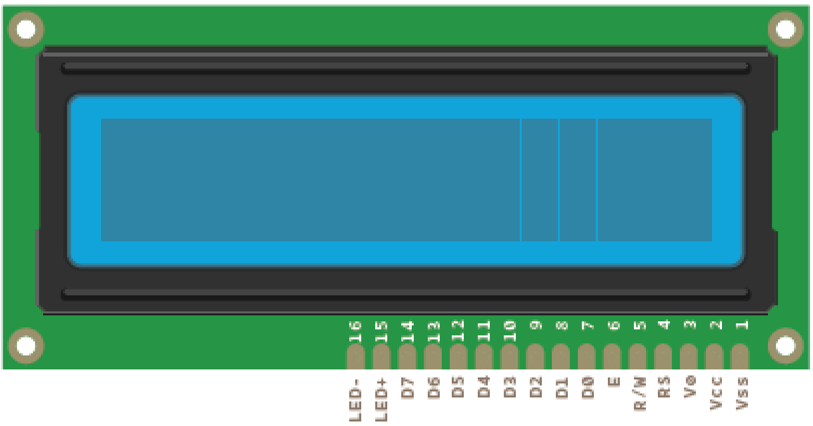
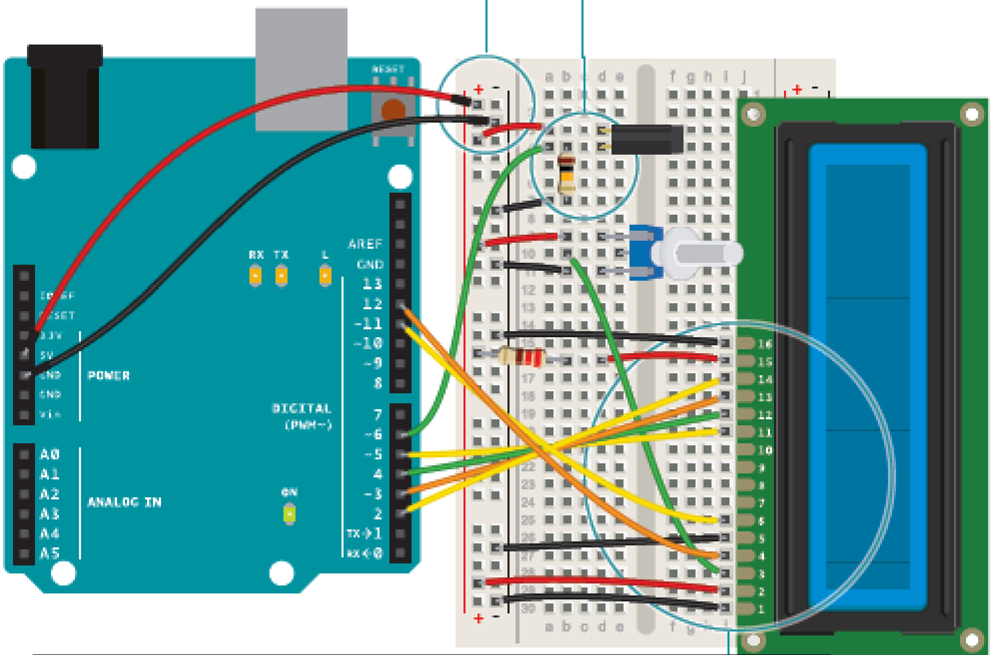


Fig. 1

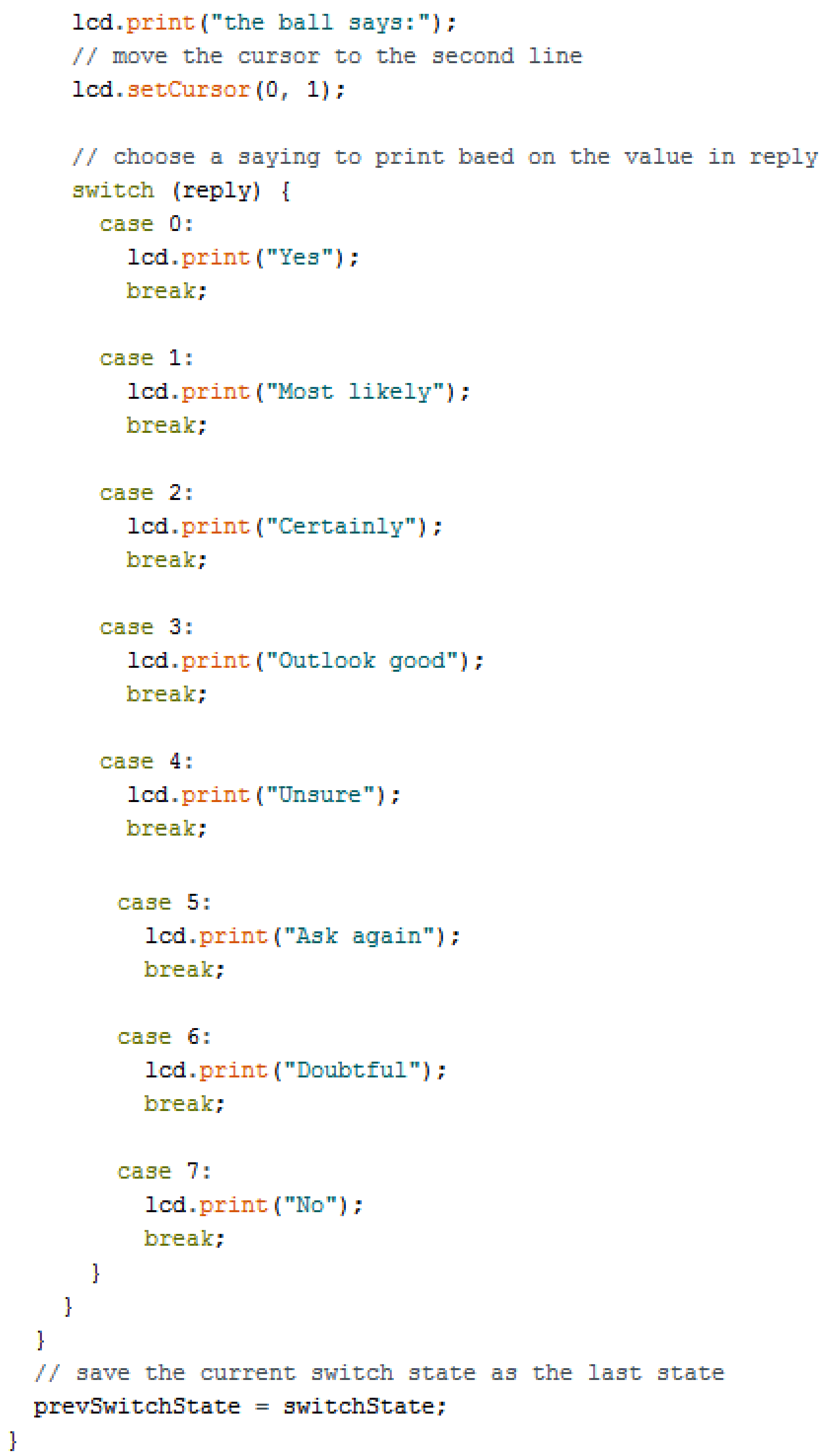
Place the tilt switch on the breadboard and attach one lead to 5V. Attach the other side to ground through a 10-kilohm resistor, and to your Arduino's pin 6. You're wiring this as a digital input.



The two outside pins of the LCD (Vss and LED-) need to be connected to 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-ohm resistor. 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. Connect EN on the screen to pin 11 on your Arduino. RS on the LCD connects to pin 12. This pin enables writing to the LCD. 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 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.

```
// include the library code:
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

// set up a constant for the tilt switchPin
const int switchPin = 6;

// variable to hold the value of the switchPin
int switchState = 0;

// variable to hold previous value of the switchpin
int prevSwitchState = 0;

// a variable to choose which reply from the crystal ball
int reply;

void setup() {
  // set up the number of columns and rows on the LCD
  lcd.begin(16, 2);

  // set up the switch pin as an input
  pinMode(switchPin, INPUT);

  // Print a message to the LCD.
  lcd.print("Ask the");
  // set the cursor to column 0, line 1
  // line 1 is the second row, since counting begins with 0
  lcd.setCursor(0, 1);
  // print to the second line
  lcd.print("Crystal Ball!");
}

void loop() {
  // check the status of the switch
  switchState = digitalRead(switchPin);

  // compare the switchState to its previous state
  if (switchState != prevSwitchState) {
    // if the state has changed from HIGH to LOW
    // you know that the ball has been tilted from
    // one direction to the other
    if (switchState == LOW) {
      // randomly chose a reply
      reply = random(8);
      // clean up the screen before printing a new reply
      lcd.clear();
      // set the cursor to column 0, line 0
      lcd.setCursor(0, 0);
      // print some text
```



```
     lcd.print("the ball says:");
     // move the cursor to the second line
     lcd.setCursor(0, 1);

     // choose a saying to print baed on the value in reply
     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;
      }
    }
  }
  // save the current switch state as the last state
  prevSwitchState = switchState;
}
```

Crystal balls can help “predict” the future. You ask a question to the all-knowing ball, and turn it over to reveal an 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.

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.

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.

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

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. 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!”.

Now, when you start the program, it will say “Ask the Crystal ball!” on your screen.

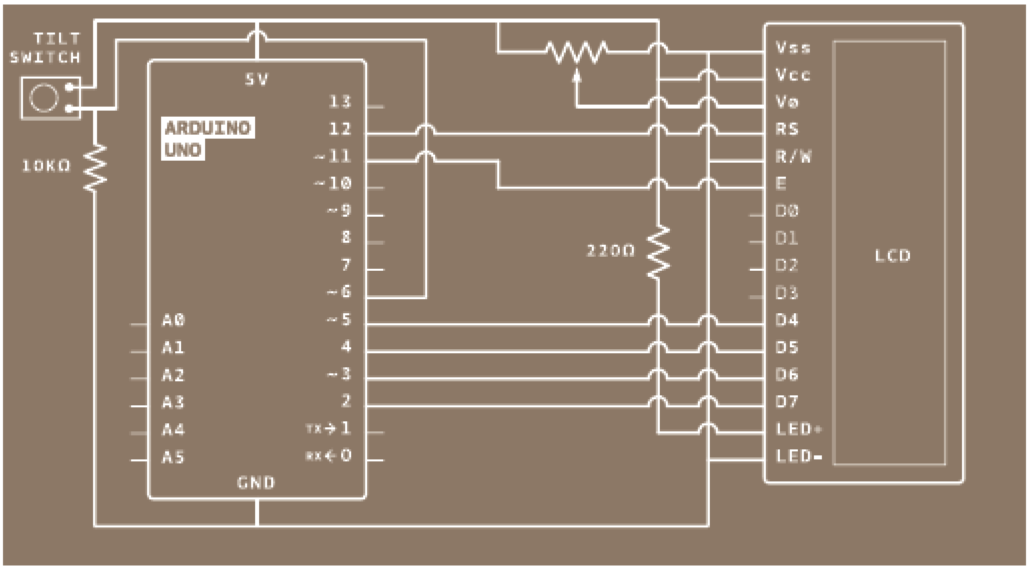
In the loop(), you’re going to check the switch first, and put the value in the switchState variable.

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. Clear the screen with the function lcd.clear(). This also moves the cursor back to 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.

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 the 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 will ask you to try again. The 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.


TILT SWITCH
10KΩ
5V
ARDUINO UNO
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
A0
A1
A2
A3
A4
A5
GND
220Ω
Vss
Vcc
Vo
RS
R/W
E
D0
D1
D2
D3
D4
D5
D6
D7
LED+
LED-
LCD