Embedded Hardware Design

Monsoon 2017

Experiment 3

**Date : ** 24-08-2017
Topics: ADC and LCD Display

Analog to Digital Converter (ADC) on Arduino-

Arduino Uno has 6 ADC channels (A0 to A5) as shown below-

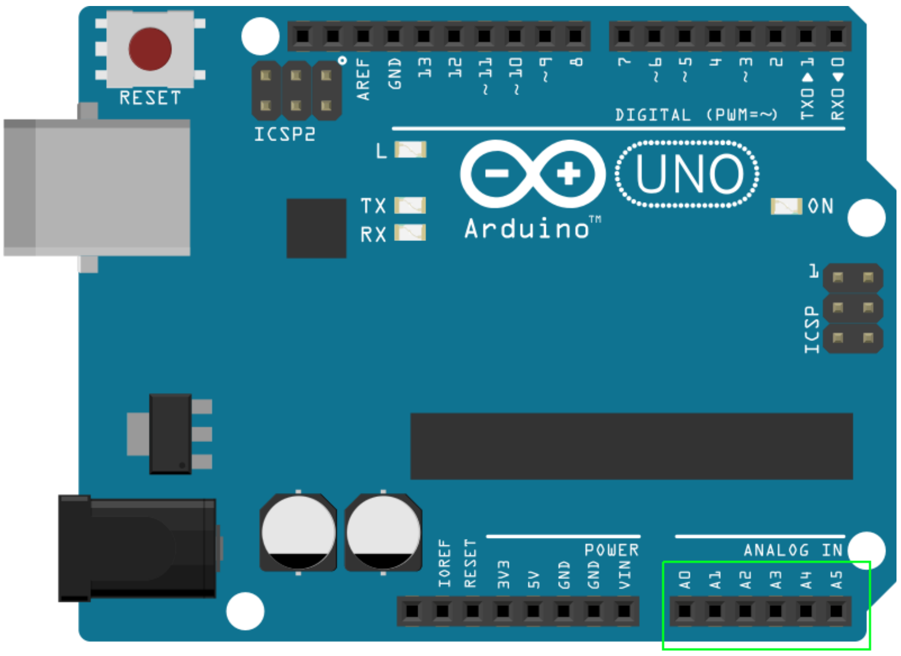

Illustration 1: Arduino ADC Pins

You may choose to work with any number of Analog pins (out of 6) at a time. The converter has 10-bit resolution, i.e, it returns integers between 0 to 1023 based on the voltage sensed. This means that the input voltages between 0 and 5V will be mapped to integer values between 0 and 1023. So, this yields a resolution between readings of (5V / 1024 units) or (4.9 mV) per unit. The main utility of the analog pins for most Arduino users is for reading analog sensors.

The function to read analog input from a pin in arduino is `analogRead(pin)`. It returns a value between 0 and 1023. The ADC value between 0 and 1023 can be converted into voltage by: $Voltage=\frac{5}{1024}*(ADC\,Value)$

Note that to use an analog pin as an input, you need not specify it in pinMode() as A0-A5 are already set as input by default.

SHARP GP2Y0A21YK Sensor-

This is an analog IR sensor which is used to measure short distances. Distance sensors are actively used in robotics to detect obstacles.

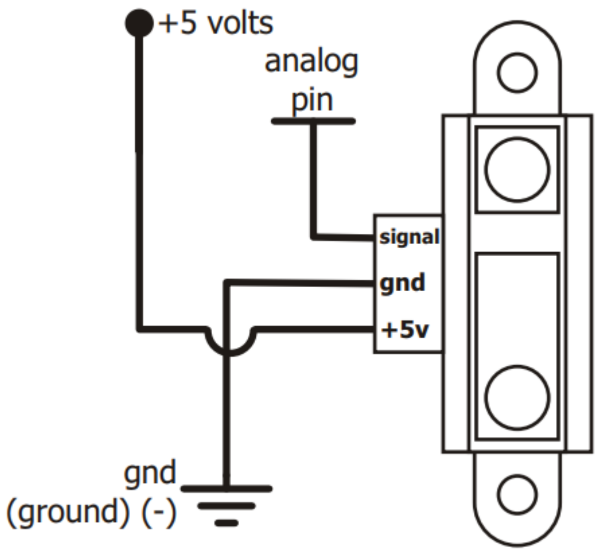

Illustration 3: Sharp Sensor Pins

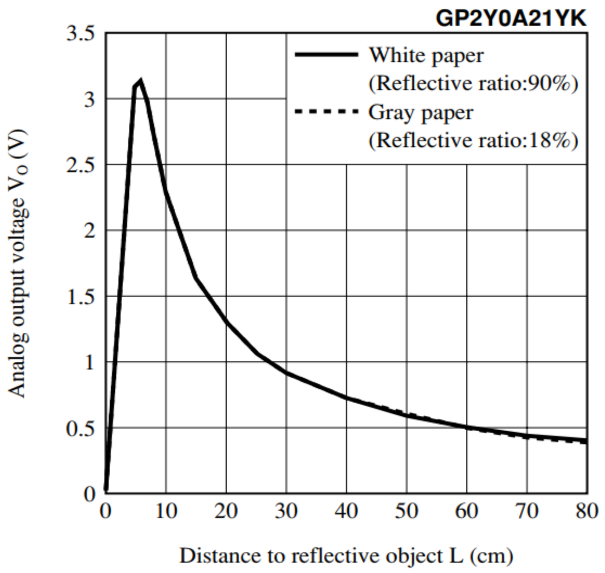

Illustration 2: Voltage-Distance Curve of Sharp Sensor

Features-

a) Detecting Distance: 10 to 80cm.
b) No external control circuit required.
c) Less influence on the color of reflective objects.
d) Low Cost

Now, we know that if we read values from the analog pin, we'll get an integer between 0 and 1023 depending on the voltage. But this is not the distance! How should we map an integer to the distance value in cm? This is where the Analog Voltage Vs Distance plot comes in handy (Figure 2). Can you think of a simple equation that can fit the curve and give a mapping from voltage to distance?

16x2 LCD Module-

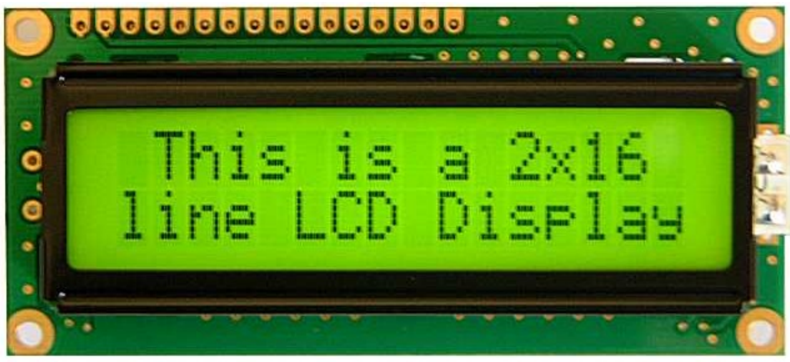

Illustration 4: 16x2 LCD character Display

Your Arduino is finally getting a display of its own! The image on the left is of a 16x2 (16 columns,2 rows)LCD character display. Numerous gadgets use this screen to display information. The RFID card readers in Mess and Front Gate also have the same screen!

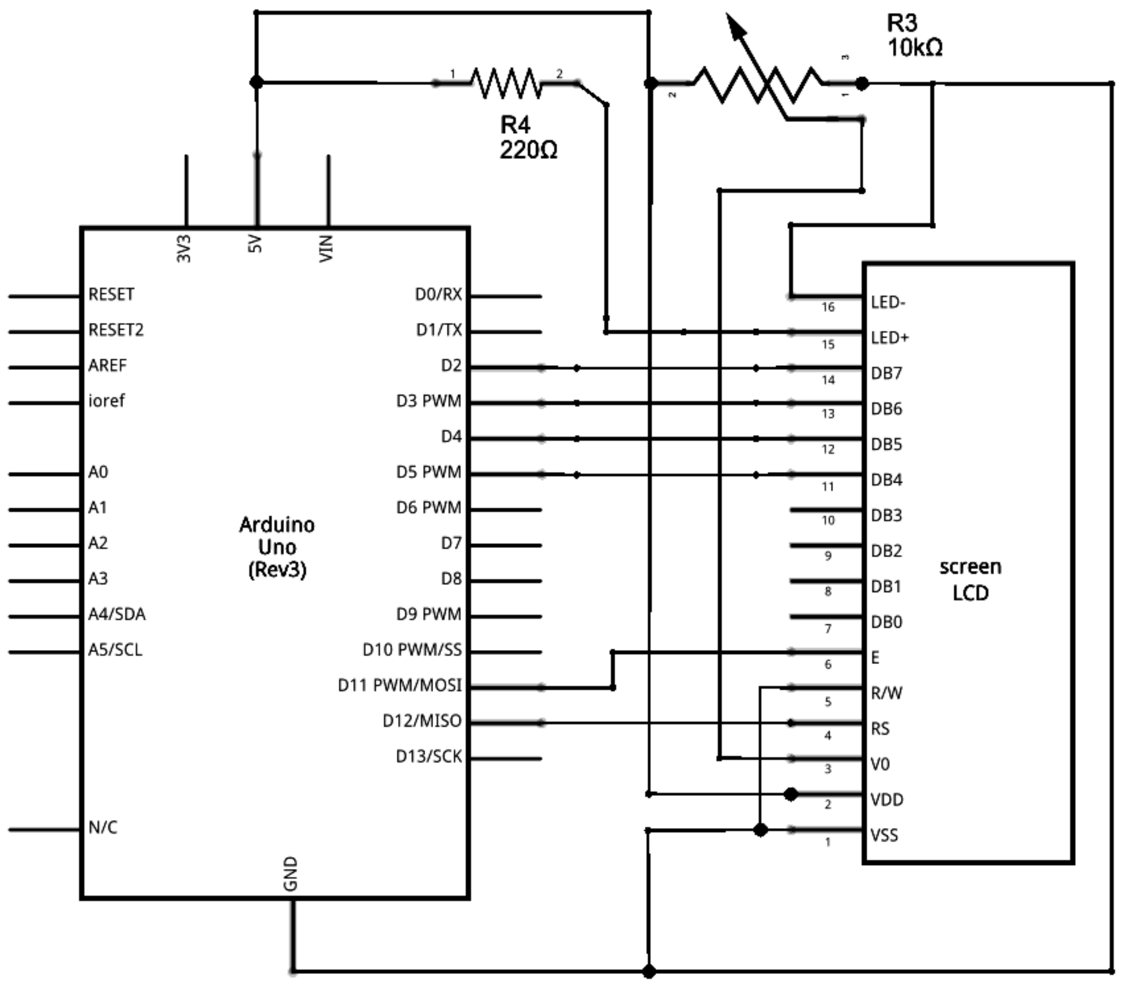

Illustration 5: Arduino to LCD connections

The “Liquid Crystal Library” of Arduino allows you to control LCD displays very easily.

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(RS, EN, D4, D5, D6, D7);    //We make an
object of LiquidCrystal and initialize it with the Arduino pins
connected to the display.

void setup() {
    lcd.begin(16,2);    //Since this is a 16x2 LCD
    lcd.print(“Hello world”);
}
void loop() {
    ......  //Logic
}
```

Other Useful Functions-

1. **SetCursor(row, column):** Bring cursor to any value in the 16x2 matrix.
2. **Clear():** Clear screen and set cursor to (0,0).
3. **noDisplay():** Turns off the LCD display, without losing the text currently shown on it.
4. **display():** Turns on the LCD display, after it's been turned off with noDisplay(). This will restore the text (and cursor) that was on the display.

Experiment 3A: Reading a Potentiometer

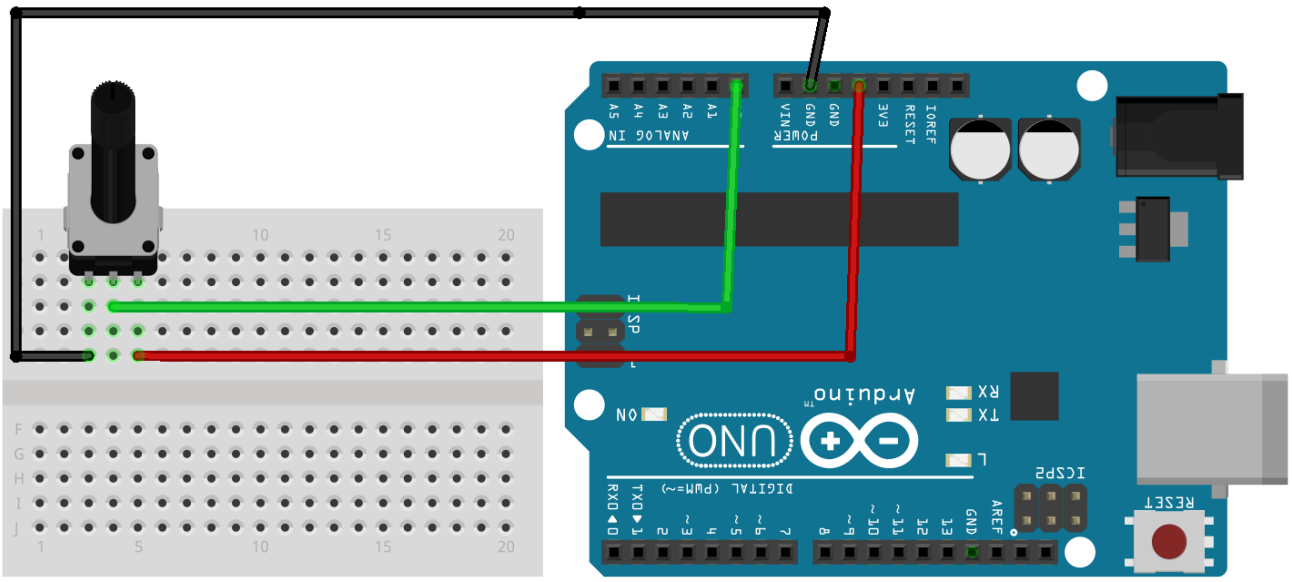

Illustration 6: Potentiometer to Arduino

1. Build the above circuit. The middle pin of the potentiometer (wiper) should go in the analog 0 pin (A0) of the arduino.
2. Initialize the Serial Monitor.
3. Read A0 every 50ms and print its value on the Serial Monitor. Observe the minimum and maximum value.
4. In the code, convert the ADC value to voltage. Now print the voltage value on the Serial monitor.

Experiment 3B: Reading Distance from Sharp IR Sensor

1. Interface the Sharp sensor as shown in figure 3. Use analog pin A1.
2. Initialize the Serial Monitor.
3. Read A1 every 50ms. Convert the analog value into distance (cm) and print it on the Serial Monitor.
4. Check the accuracy of the sensor by keeping objects at known distances.

Experiment 3C: Displaying Information on the LCD

1. Make the connections as per Figure 5.
2. Adjust the contrast of the screen via the potentiometer.
3. Print “Hello world” using the pseudo code provided. Don’t do anything in loop() yet. This is just to test the screen.
4. Combine Experiment 2 and print the distance value to the LCD in the following format:

```
Distance:17.7cm
```