

Laboratory 9: Arduino with other components

written by: M. Rao

Revision Date: November 10, 2016

Seven segment display: HDSP5501

Seven segment display consists of seven segments of LED for displaying decimal numerals. These are widely used in digital clocks, electronic meters and other electronic devices. Use Seven segment display unit, and arduino microcontroller to complete the following connections:

Arduino Pin	7 Segment Pin
2	7
3	6
4	4
5	2
6	1
7	9
8	10
9	5

Connect pin 3 and 8 (common-anode) of seven-segment display to 5 V. Note that HDSP5501 is a common-anode display. If a pin of 7 Segment is connected to 0V (GND), that particular LED will glow. For common-cathode, the pin needs to be connected to 5 V so as to glow the LED. Use the code uploaded in the LMS to show a display running from 9 to 0. Can you modify the code such that only prime numbers starting from 0 to 9 are displayed. We can assume that 0 and 1 are prime numbers.

Potentiometer driven servo control

Use the CIRC-04 given in the Arduino manual and connect the servo hobby motor to Arduino controller. The code given in the manual rotates the servo from 0 to 180 and back. However we would like some knob control to rotate the servo. Refer CIRC-08 to rotate the knob and control the servo from 0 to 180 and back. You can also refer to the link: <http://www.arduino.cc/en/Tutorial/Knob> to use potentiometer and rotate the knob. Show the working of this circuit to your lab instructor.

User input driven servo control

Your arduino controller can take inputs from the computer which is connected via USB cable. The USB cable acts as a virtual serial port (USB interface) connecting device A (arduino) to device B (PC). The

device A and device B follows serial (RS232 or RS485) communication protocol over USB interface. (NOTE that USB can encapsulate different protocols such as serial, bluetooth, RF, zigbee and wifi) To send your inputs from PC to arduino, open the SERIAL-MONITOR (click right-side icon) from your arduino software interface. The SERIAL-MONITOR works as user console. User can provide inputs to arduino using Serial-monitor. Serial-monitor can be used as display unit as well. Check whether the baud-rate is fixed to 9600. Your program needs to set the baudrate to 9600 as shown below. Work on this simple example given below. Check your output on SERIAL-MONITOR.

```
int incomingByte = 0;    // for incoming serial data

void setup()
{
  Serial.begin(9600);    // opens serial port, sets data rate to 9600 bps
}

void loop()
{
  // send data only when you receive data:
  if (Serial.available() > 0)
  {
    // read the incoming byte:
    incomingByte = Serial.read();

    // say what you got:
    Serial.print("I received: ");
    Serial.println(incomingByte, DEC);
  }
}
```

Now program your servo in such a way that servo rotates to user inputs and comes back from that position to zero.