

AD5171 Digital Potentiometer

This example shows how to control a Analog Devices AD5171 Digital Potentiometer (<http://datasheet.octopart.com/AD5171BRJZ10-R2-Analog-Devices-datasheet-8546.pdf>) which communicates via the I2C synchronous serial protocol (<http://en.wikipedia.org/wiki/I2C>). Using Arduino's I2C **Wire Library** ([//www.arduino.cc/en/Reference/Wire](http://www.arduino.cc/en/Reference/Wire)), the digital pot will step through 64 levels of resistance, fading an LED.

The I2C protocol involves using two lines to send and receive data: a serial clock pin (SCL) that the Arduino or Genuino Master board pulses at a regular interval, and a serial data pin (SDA) over which data is sent between the two devices. As the clock line changes from low to high (known as the rising edge of the clock pulse), a single bit of information - that will form in sequence the address of a specific device and a command or data - is transferred from the board to the I2C device over the SDA line. When this information is sent - bit after bit -, the called upon device executes the request and transmits it's data back - if required - to the board over the same line using the clock signal still generated by the Master on SCL as timing.

Because the I2C protocol allows for each enabled device to have it's own unique address, and as both master and slave devices to take turns communicating over a single line, it is possible for your Arduino or Genuino board to communicate (in turn) with many devices, or other boards, while using just two pins of your microcontroller.

Hardware Required

- Arduino or Genuino Board
- AD5171 Digital Pot
- LED
- 680 ohm resistor
- 2 4.7k ohm resistors
- hook-up wires
- breadboard

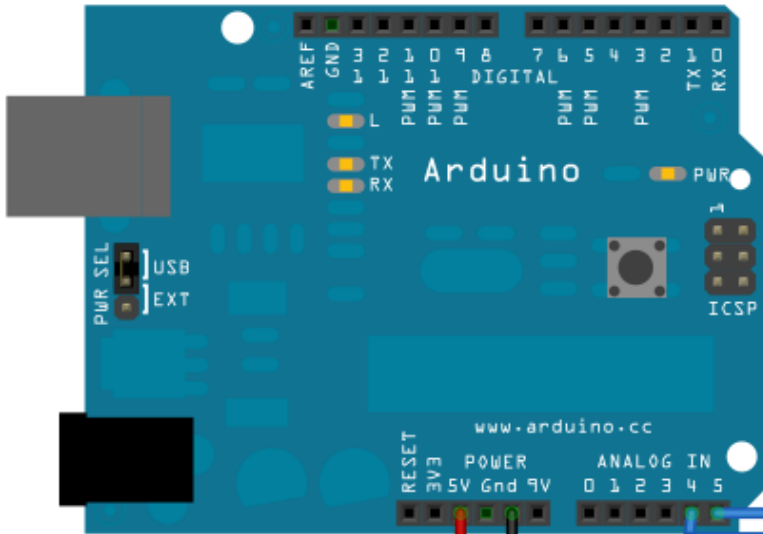
Circuit

Connect pins 3, 6, and 7 of the AD5171 to GND, and pins 2 and 8 to +5V.

(<https://www.arduino.cc>)

Finally, wire an LED to pin 1, the AD5171's "wiper", with a 680 ohm LED in series.

STORE (//STORE.ARDUINO.CC) SOFTWARE EDUCATION (//WWW.AR



(//www.arduino.cc/en/uploads/Tutorial/AD5171_bb.png)

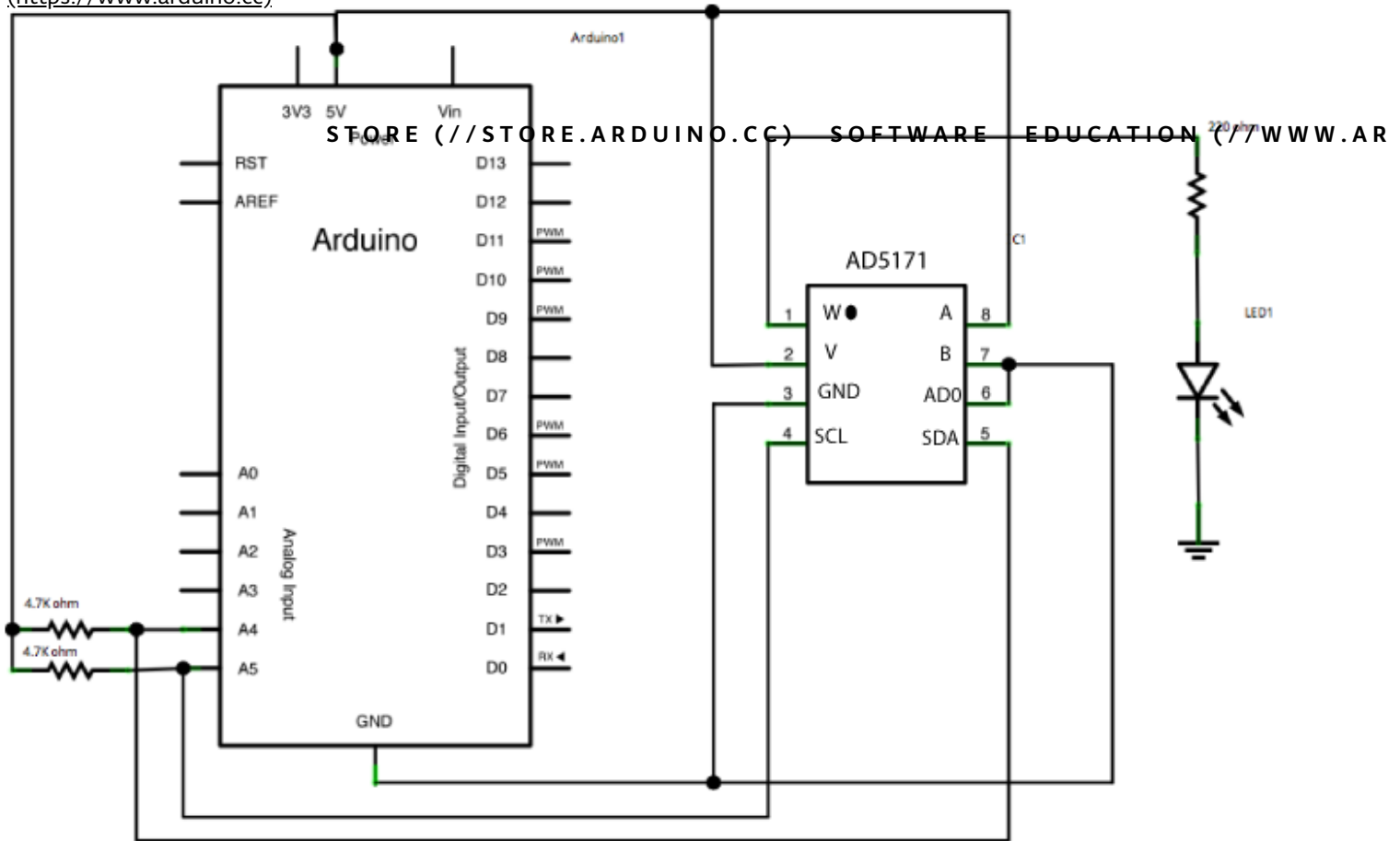
image developed using Fritzing (<http://www.fritzing.org>). For more circuit examples, see the Fritzing project page (<http://fritzing.org/projects/>)

When the AD5171's pin 6, ADO, is connected to ground, its address is 44. To add another digital pot to the same SDA bus, connect the second pot's ADO pin to +5V, changing its address to 45.

You can only use two of these digital potentiometers simultaneously.

Schematic

(<https://www.arduino.cc>)



(https://www.arduino.cc/en/uploads/Tutorial/AD5171_sch.png)

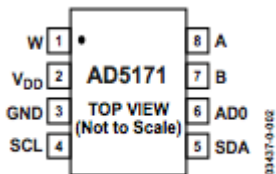


Figure 2. Pin Configuration

(https://www.arduino.cc/en/uploads/Tutorial/AD5171_pinconfig.png)

Code

digital_potentiometer

STORE ([//STORE.ARDUINO.CC](https://store.arduino.cc)) SOFTWARE EDUCATION ([//WWW.AR](https://www.arduino.cc)

```
digital_potentiome
21
22 byte val = 0;
23
24 void loop()
25 {
26   Wire.beginTransmission(44); // transmit to device #44 (0x2c)
27   // device address is specified in datasheet
28   Wire.write(byte(0x00));      // sends instruction byte
29   Wire.write(val);             // sends potentiometer value byte
30   Wire.endTransmission();      // stop transmitting
31
32   val++;                       // increment value
33   if(val == 64) // if reached 64th position (max)
34   {
35     val = 0; // start over from lowest value
36   }
37   delay(500);
38 }
39
40
```

See Also

- [Wire.begin \(\[//www.arduino.cc/en/Reference/WireBegin\]\(https://www.arduino.cc/en/Reference/WireBegin\)\)\(\)](https://www.arduino.cc/en/Reference/WireBegin)
- [Wire.beginTransmission \(\[//www.arduino.cc/en/Reference/WireBeginTransmission\]\(https://www.arduino.cc/en/Reference/WireBeginTransmission\)\)\(\)](https://www.arduino.cc/en/Reference/WireBeginTransmission)
- [Wire.endTransmission \(\[//www.arduino.cc/en/Reference/WireEndTransmission\]\(https://www.arduino.cc/en/Reference/WireEndTransmission\)\)\(\)](https://www.arduino.cc/en/Reference/WireEndTransmission)
- [Wire.send \(\[//www.arduino.cc/en/Reference/WireSend\]\(https://www.arduino.cc/en/Reference/WireSend\)\)\(\)](https://www.arduino.cc/en/Reference/WireSend)
- [Wire Library \(\[//www.arduino.cc/en/Reference/Wire\]\(https://www.arduino.cc/en/Reference/Wire\)\)](https://www.arduino.cc/en/Reference/Wire) – Your reference for the Wire Library.
- [SFRRanger_reader \(\[//www.arduino.cc/en/Tutorial/SFRRangerReader\]\(https://www.arduino.cc/en/Tutorial/SFRRangerReader\)\)](https://www.arduino.cc/en/Tutorial/SFRRangerReader) - How to read an ultra-sonic range finder interfaced via the I2C.
- [Master Reader/Slave Writer \(\[//www.arduino.cc/en/Tutorial/MasterReader\]\(https://www.arduino.cc/en/Tutorial/MasterReader\)\)](https://www.arduino.cc/en/Tutorial/MasterReader) - Two Arduino are programmed to communicate with one another in a Master Reader/Slave Sender configuration via the I2C.
- [Master Writer/Slave receiver \(\[//www.arduino.cc/en/Tutorial/MasterWriter\]\(https://www.arduino.cc/en/Tutorial/MasterWriter\)\)](https://www.arduino.cc/en/Tutorial/MasterWriter) - Two Arduino are programmed to communicate in a Master Writer/Slave Receiver configuration via the I2C.

NEWSLETTER

Enter your email to sign up

SUBSCRIBE

Terms Of Service ([//www.arduino.cc/en/Main/TermsOfService](http://www.arduino.cc/en/Main/TermsOfService))

[Privacy Policy \(//www.arduino.cc/en/Main/PrivacyPolicy\)](http://www.arduino.cc/en/Main/PrivacyPolicy)

Contact Us ([//www.arduino.cc/en/Main/ContactUs](http://www.arduino.cc/en/Main/ContactUs))

About Us ([//www.arduino.cc/en/Main/AboutUs](http://www.arduino.cc/en/Main/AboutUs))

Distributors ([//store.arduino.cc/distributors](https://store.arduino.cc/distributors))

Careers ([//careers.arduino.cc](http://careers.arduino.cc))

Security ([//www.arduino.cc/en/Main/Security](http://www.arduino.cc/en/Main/Security))

© 2020 Arduino ([//www.arduino.cc/en/Main/CopyrightNotice](http://www.arduino.cc/en/Main/CopyrightNotice))

[illegible]