Master Writer/Slave Receiver

Sometimes, the folks in charge just don't know when to shut up! In some situations, it can be helpful to set up two (or more!) Arduino or Genuino boards to share information with each other. In this example, two boards are programmed to communicate with one another in a Master Writer/Slave Receiver configuration via the I2C synchronous serial protocol (http://en.wikipedia.org/wiki/I2C). Several functions of Arduino's **Wire Library** (//www.arduino.cc/en/Reference/Wire) are used to accomplish this. Arduino 1, the Master, is programmed to send 6 bytes of data every half second to a uniquely addressed Slave. Once that message is received, it can then be viewed in the Slave board's serial monitor window opened on the USB connected computer running the Arduino Software (IDE).

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 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. The initial eight bits (i.e. eight clock pulses) from the Master to Slaves contain the address of the device the Master wants data from. The bits after contain the memory address on the Slave that the Master wants to read data from or write data to, and the data to be written, if any.

Each Slave device has to have its own unique address and both master and slave devices need to take turns communicating over a the same data line line. In this way, it's possible for your Arduino or Genuino boards to communicate with many device or other boards using just two pins of your microcontroller, using each device's unique address.

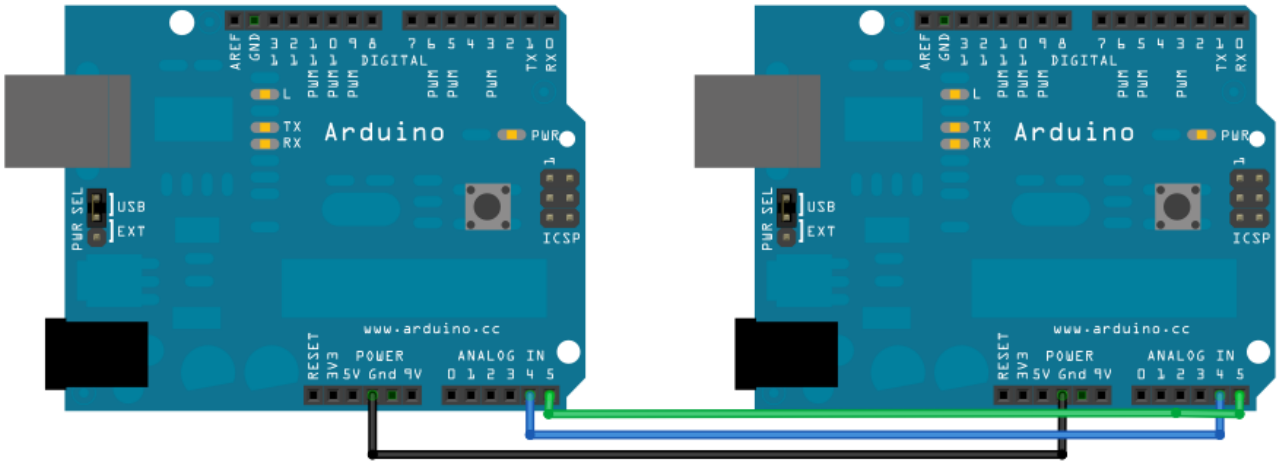
Hardware Required

- 2 Arduino or Genuino Boards
- hook-up wires

Circuit

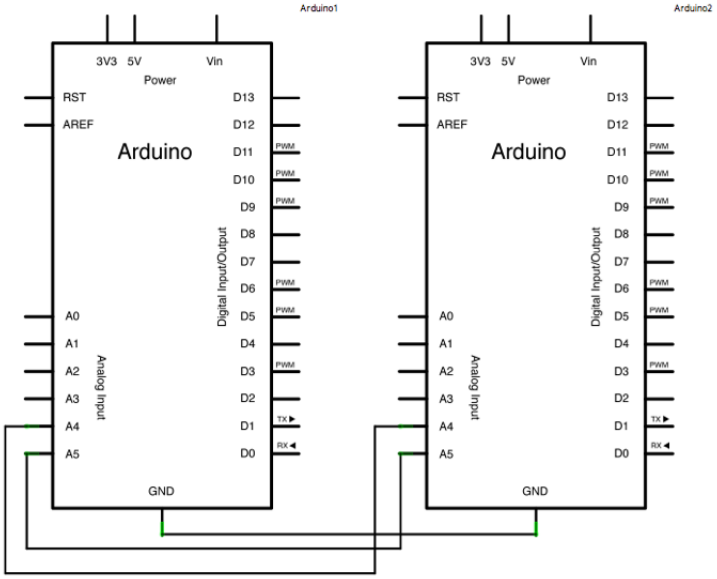
Connect pin 5 (the clock, or SCL, pin) and pin 4 (the data, or SDA, pin) on the master Arduino to their counterparts on the slave board. Make sure that both boards share a common ground. In order to enable serial communication, the slave Arduino must be connected to your computer via USB.

If powering the boards independently is an issue, connect the 5V output of the Master to the VIN pin on the slave.



(//www.arduino.cc/en/uploads/Tutorial/Master_Sender_bb.png)
image developed using Fritzing (http://www.fritzing.org). For more circuit examples, see the Fritzing project page (http://fritzing.org/projects/)

Schematic



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

Code

Master Writer Code - Program for Arduino 1

```
// Wire Master Writer
// by Nicholas Zambetti <http://www.zambetti.com>

// Demonstrates use of the Wire library
// Writes data to an I2C/TWI slave device
// Refer to the "Wire Slave Receiver" example for use with this

// Created 29 March 2006

// This example code is in the public domain.


#include <Wire.h>

void setup() {
  Wire.begin(); // join i2c bus (address optional for master)
}

byte x = 0;

void loop() {
  Wire.beginTransmission(8); // transmit to device #8
  Wire.write("x is ");        // sends five bytes
  Wire.write(x);              // sends one byte
  Wire.endTransmission();    // stop transmitting

  x++;
  delay(500);
}
```

[Get Code] (//www.arduino.cc/en/Tutorial/MasterWriter?action=sourceblock&num=1)

Slave Receiver Code - Program for Arduino 2

```
// Wire Slave Receiver
// by Nicholas Zambetti <http://www.zambetti.com>

// Demonstrates use of the Wire library
// Receives data as an I2C/TWI slave device
// Refer to the "Wire Master Writer" example for use with this

// Created 29 March 2006

// This example code is in the public domain.
```

```
#include <Wire.h>

void setup() {
  Wire.begin(8);                // join i2c bus with address #8
  Wire.onReceive(receiveEvent); // register event
  Serial.begin(9600);           // start serial for output
}

void loop() {
  delay(100);
}

// function that executes whenever data is received from master
// this function is registered as an event, see setup()
void receiveEvent(int howMany) {
  while (1 < Wire.available()) { // loop through all but the last
    char c = Wire.read(); // receive byte as a character
    Serial.print(c);         // print the character
  }
  int x = Wire.read();    // receive byte as an integer
  Serial.println(x);         // print the integer
}
```

[Get Code] (//www.arduino.cc/en/Tutorial/MasterWriter?action=sourceblock&num=2)

See Also

- Wire.begin (//www.arduino.cc/en/Reference/WireBegin)()
- Wire.beginTransmission (//www.arduino.cc/en/Reference/WireBeginTransmission)()
- Wire.endTransmission (//www.arduino.cc/en/Reference/WireEndTransmission)()
- Wire.send (//www.arduino.cc/en/Reference/WireSend)()
- Wire.OnReceive (//www.arduino.cc/en/Reference/WireOnReceive)()
- Wire.available (//www.arduino.cc/en/Reference/WireAvailable)()
- Wire Library (//www.arduino.cc/en/Reference/Wire) – Your reference for the Wire Library.
- DigitalPotentiometer (//www.arduino.cc/en/Tutorial/DigitalPotentiometer) - How to control an Analog Devices AD5171 Digital Potentiometer.
- SFRRanger_reader (//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) - Two Arduino are programmed to communicate with one another in a Master Reader/Slave Sender configuration

via the I2C.

Last revision 2018/05/17 by SM

Share