In the Max tutorials, under the **Communications** heading is a tutorial for Serial Communication. (Here is the web version of it, but you need to access the tutorial through Max to open the patcher.)

Here is the Arduino code for the tutorial. (The Max tutorial window doesn't let you copy, so you can copy it from here.) Copy the code and paste it into an Arduino sketch (that's the name for an Arduino program). Upload it to your Arduino board.

Read through the entire tutorial. This explains how Arduino and Max communicate. Pay special attention to:

- How the [serial] object works.
    - assigning the correct serial port
    - *Baud rate* is the speed at which Max and Arduino talk to each other. The baud rate of the [serial] object has to be the same as the baud rate specified in the Arduino program. The code we uploaded to the Arduino board has the line `Serial.begin(9600);` which tells it to communicate at 9600 baud, so that is the rate that is set in the [serial] object too.
    - Transmitting data: the [serial] object transmits integers between 0 and 255 that come in its inlet.
    - Receiving data: the [serial] object polls the serial port (in a way similar to how [mousestate] polls the mouse information)

Arduino + Max/MSP/Jitter = Arduino2Max

Arduino2Max is a simple Max patch and corresponding Arduino code to allow Max to read from the Arduino pins (but it doesn't write to the Arduino pins). You can download it from this page. Open the Arduino code and upload it to your board. Then open the Max patch. Generally, it works right when you start it up. Hook up one of the circuits for reading from a switch/button or reading from a variable resistor. You should then see Arduino2Max respond to changes to the switch/button or variable resistor.

Modifying Arduino2Max to write to Arduino

With the information from the Max Serial Communication tutorial, we can modify it to get data to travel the other direction. Simply using Max to turn on of the Arduino pins on (HIGH) or off (LOW) is pretty easy.

I have made a modified Arduino2Max patch and Arduino program to show how this might work. You can download it here: Arduino2Max_withDigOut.

- In the Max patch, I have added a section at the bottom with some toggles that I use to send number via [serial] to Arduino. If you open up the DigitalOuts patcher object, you can see that I'm using [+] objects so each toggle sends a unique pair of numbers. Arduino2Max's Max patch sends the letter 'r' repeatedly to Arduino to cause it to read from all of its pins. With these modifications, it continues to send the 'r', but also sends the numbers whenever the toggles are changed. You can add to the DigitalOuts patcher object to send a larger range of numbers.
- In the Arduino code we have to deal with the new input from Max. In order to do this:
    - We have to decide which pins will be INPUTs and which will be OUTPUTs. In the setup() routine we tell Arduino which pins are going to

be OUTPUTs with the command `pinMode(12, OUTPUT);` setting pin 12 as an output. In my sample code, pins 12 and 13 are outputs.

- We have to modify the "for" loop that reads the digital pins so that it doesn't include the pins that we have set as outputs. So, if we are going to use the upper pins as outputs, then we can just modify the higher number in the "for" statement so it doesn't include them. In my example, the statement is `for (int pin= 2; pin<=11; pin++)` so it will read all of the pins up to pin 11. So, we could either tell it to start at a later pin rather than pin 2 or we can keep it from reading the higher pins by changing the 11 to some other number.
- Then we have to do something when it receives a number. The easiest thing to do is to set a pin HIGH or LOW. In my sample, a 0 (the first toggle off) sets pin 12 LOW, a 1 (the first toggle on) sets pin 12 HIGH. A 2 (second toggle off) sets pin 13 low and a 3 (second toggle on) sets pin 13 high. We can modify this to have the Arduino do whatever we want when it receives those numbers.