

Arduino

 search

[Buy](#) | [Download](#) | [Getting Started](#) | [Learning](#) | [Reference](#) | [Hardware](#) | [FAQ](#)
[Blog »](#) | [Forum »](#) | [Playground »](#)
[Learning](#) | [Examples](#) | [Foundations](#) | [Hacking](#) | [Links](#)

Parallel to Serial Shifting-In with a CD4021BE

Started By Carlyn Maw and Tom Igoe Jan, '07

Shifting In & the CD4021B

Sometimes you'll end up needing more digital input than the 13 pins on your Arduino board can readily handle. Using a parallel to serial shift register allows you collect information from 8 or more switches while only using 3 of the pins on your Arduino.

An example of a parallel to serial register is the CD4021B, sometimes referred to as an "8-Stage Static Shift Register." This means you can read the state of up to 8 digital inputs attached to the register all at once. This is called Asynchronous Parallel Input. "Input" because you are collecting information. "Parallel" because it is all at once, like hearing a musical cord. "Asynchronous" because the CD4021B is doing all this data collection at its own pace without coordinating with the Arduino.

That happens in the next step when those 8 pin states are translated into a series of HIGH and LOW pulses on the serial-out pin of the shift register. This pin should be connected to an input pin on your Arduino Board, referred to as the **data pin**. The transfer of information itself is called Synchronous Serial Output because the shift register waits to deliver linear sequence of data to the Arduino until the Arduino asks for it. Synchronous Serial communication, input or output, is heavily reliant on what is referred to as a **clock pin**. That is what makes it "synchronous." The clock pin is the metronome of the conversation between the shift register and the Arduino. Every time the Arduino sends the clock pin from LOW to HIGH it is telling the shift register "change the state of your Serial Output pin to tell me about the next switch."

The third pin attached to the Arduino is a "Parallel to Serial Control" pin. You can think of it as a **latch pin**. When the latch pin is HIGH the shift register is listening to its 8 parallel ins. When it is LOW it is listening to the clock pin and passing information serially. That means every time the latch pin transitions from HIGH to LOW the shift register will start passing its most current switch information.

The pseudo code to coordinate this all looks something like this:

1. Make sure the register has the latest information from its parallel inputs (i.e. that the latch pin is HIGH)
2. Tell the register that I'm ready to get the information serially (latch pin LOW)
3. For each of the inputs I'm expecting, pulse the clockPin and then check to see if the data pin is low or high

This is a basic diagram.

```

switch -> | _____ |
switch -> |   C   |
switch -> |   D   |
switch -> |   4   | -> Serial Data to Arduino
switch -> |   0   |
switch -> |   2   |
switch -> |   1   | <- Clock Data from Arduino
switch -> | _____ | <- Latch Data from Arduino

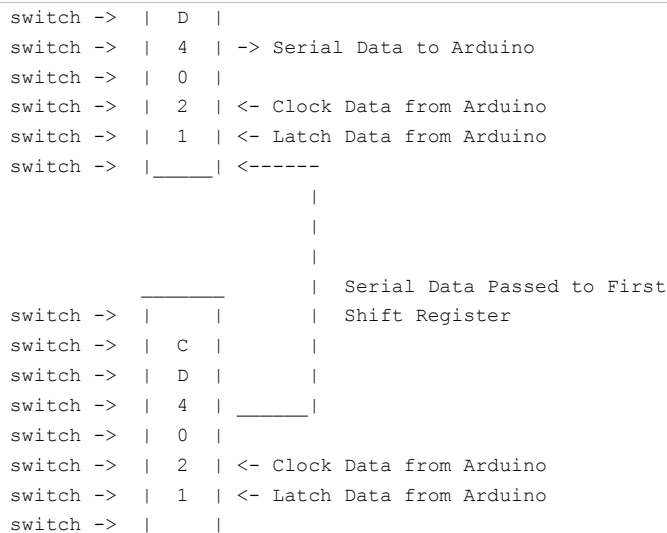
```

If supplementing your Arduino with an additional 8 digital-ins isn't going to be enough for your project you can have a second CD4021 pass its information on to that first CD4021 which will then be streaming all 16 bits of information to the Arduino in turn. If you know you will need to use multiple shift registers like this check that any shift registers you buy can handle Synchronous Serial Input as well as the standard Synchronous Serial Output capability. Synchronous Serial Input is the feature that allows the first shift register to receive and transmit the serial-output from the second one linked to it. The second example will cover this situation. You can keep extending this daisy-chain of shift registers until you have all the inputs you need, within reason.

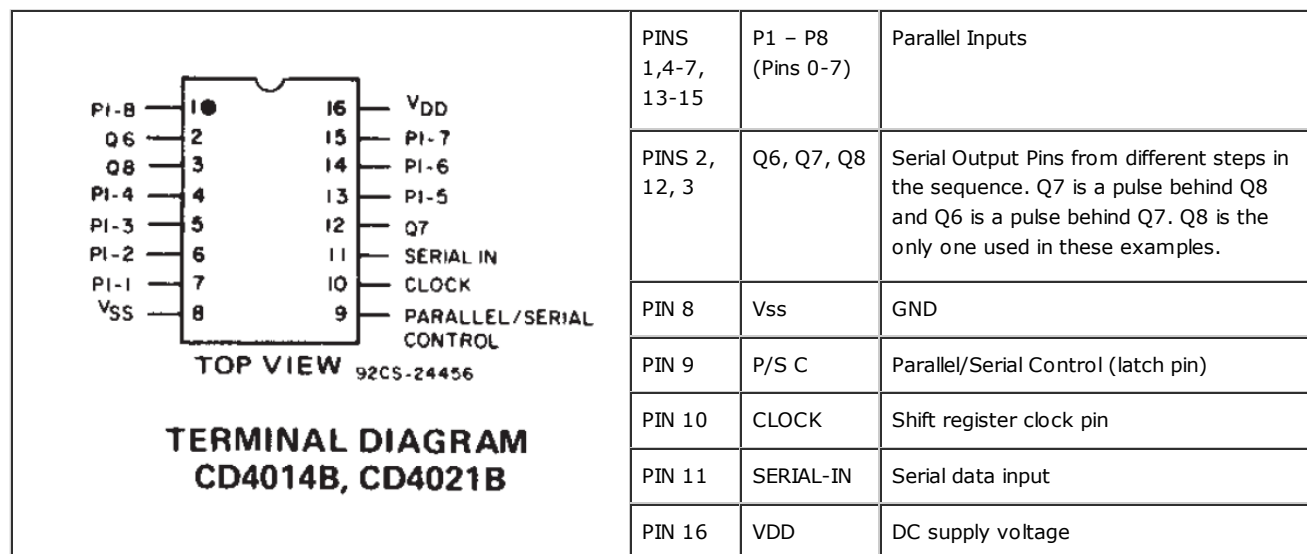
```

switch -> | _____ |
switch -> |   C   |

```



There is more information about shifting in the [ShiftOut](#) tutorial, and before you start wiring up your board here is the pin diagram of the CD4021 from the Texas Instruments Datasheet



Example 1: One Shift Register

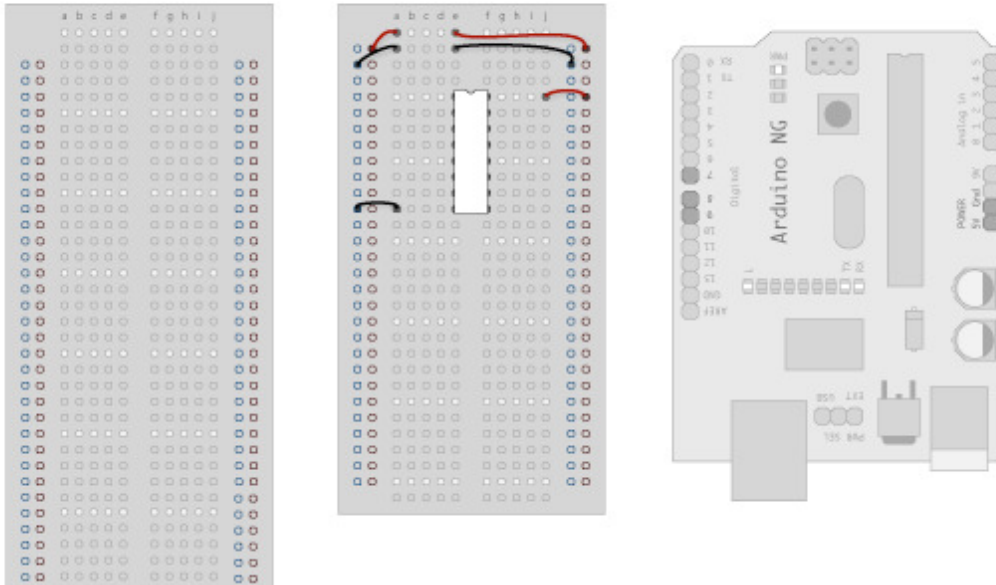
The first step is to extend your Arduino with one shift register.

The Circuit

1. Power Connections

Make the following connections:

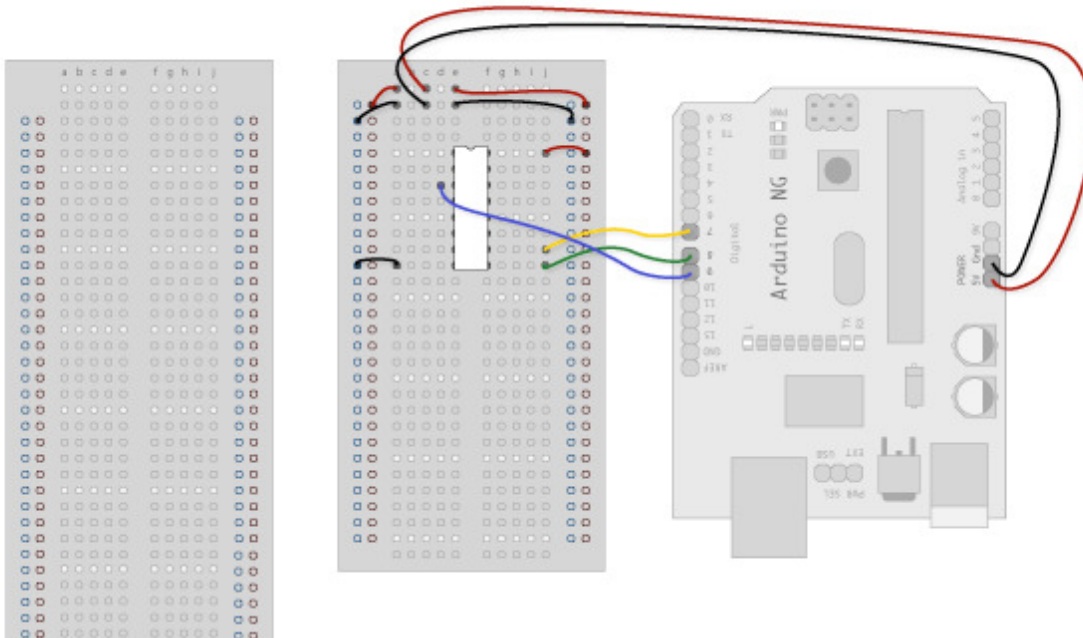
- GND (pin 8) to ground,
- VDD (pin 16) to 5V



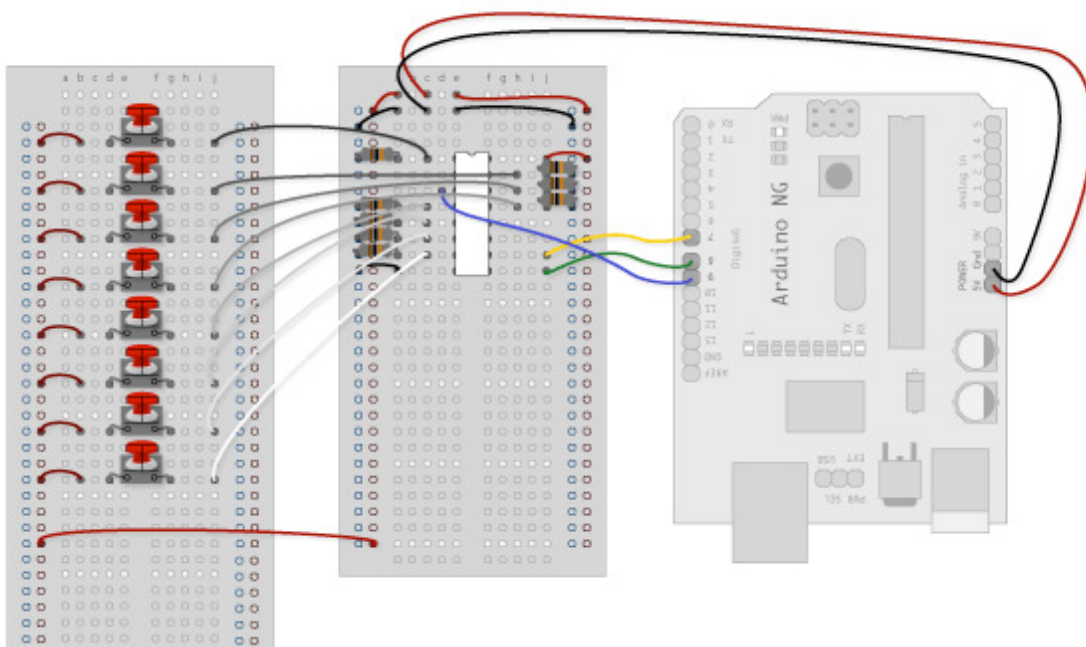
2.Connect to Arduino

- Q8 (pin 3) to Arduino DigitalPin 9 (blue wire)
- CLOCK (pin 10) to Arduino DigitalPin 7 (yellow wire)
- P/S C (pin 9) to Arduino DigitalPin 8 (green wire)

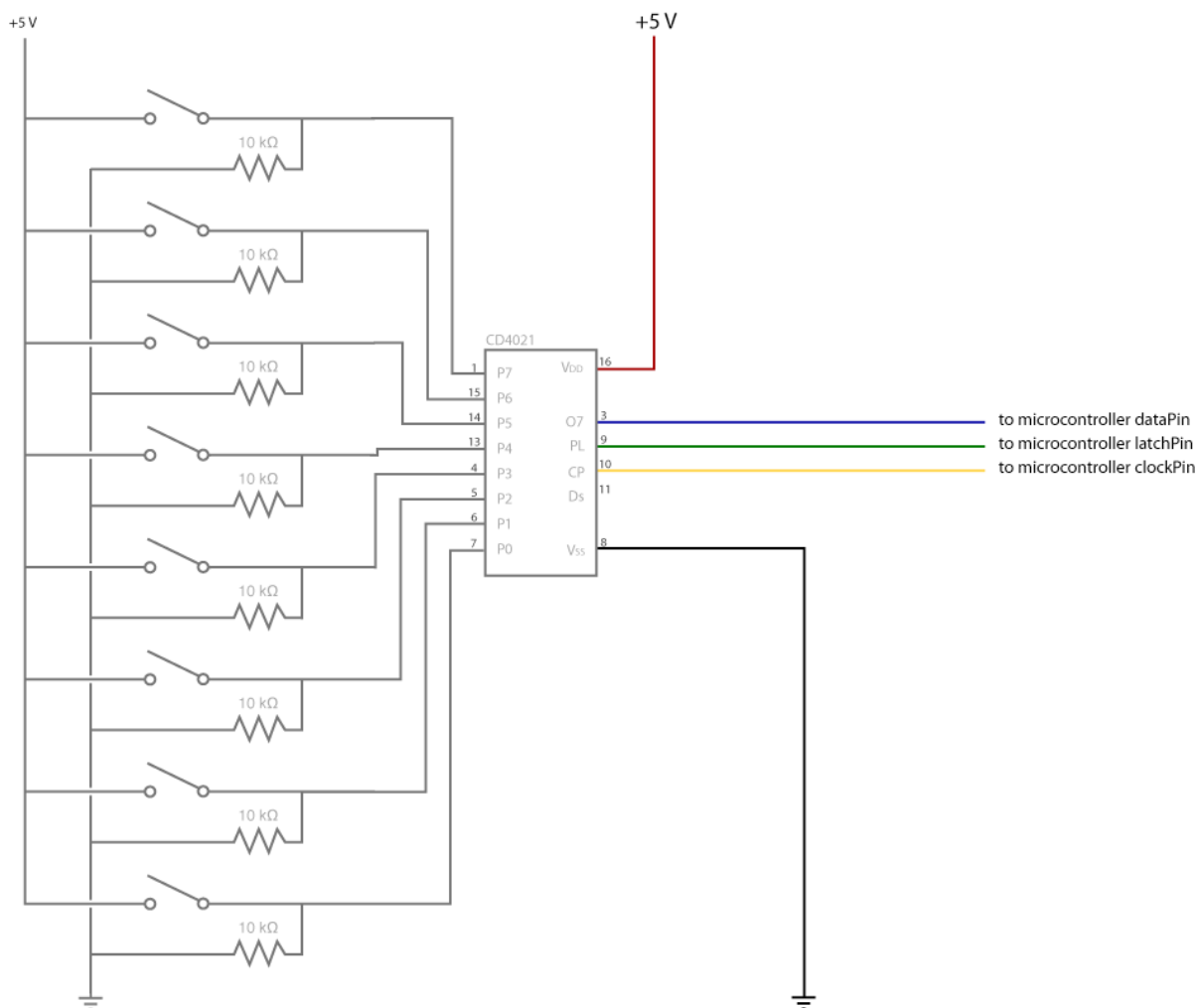
From now on those will be referred to as the dataPin, the clockPin and the latchPin respectively.



3. Add 8 Switches



Diagram



The Code

[Code Sample 1.1 – Hello World](#)

[Code Sample 1.2 – What is Pressed?](#)

Code Sample 1.3 – Button Combination Check

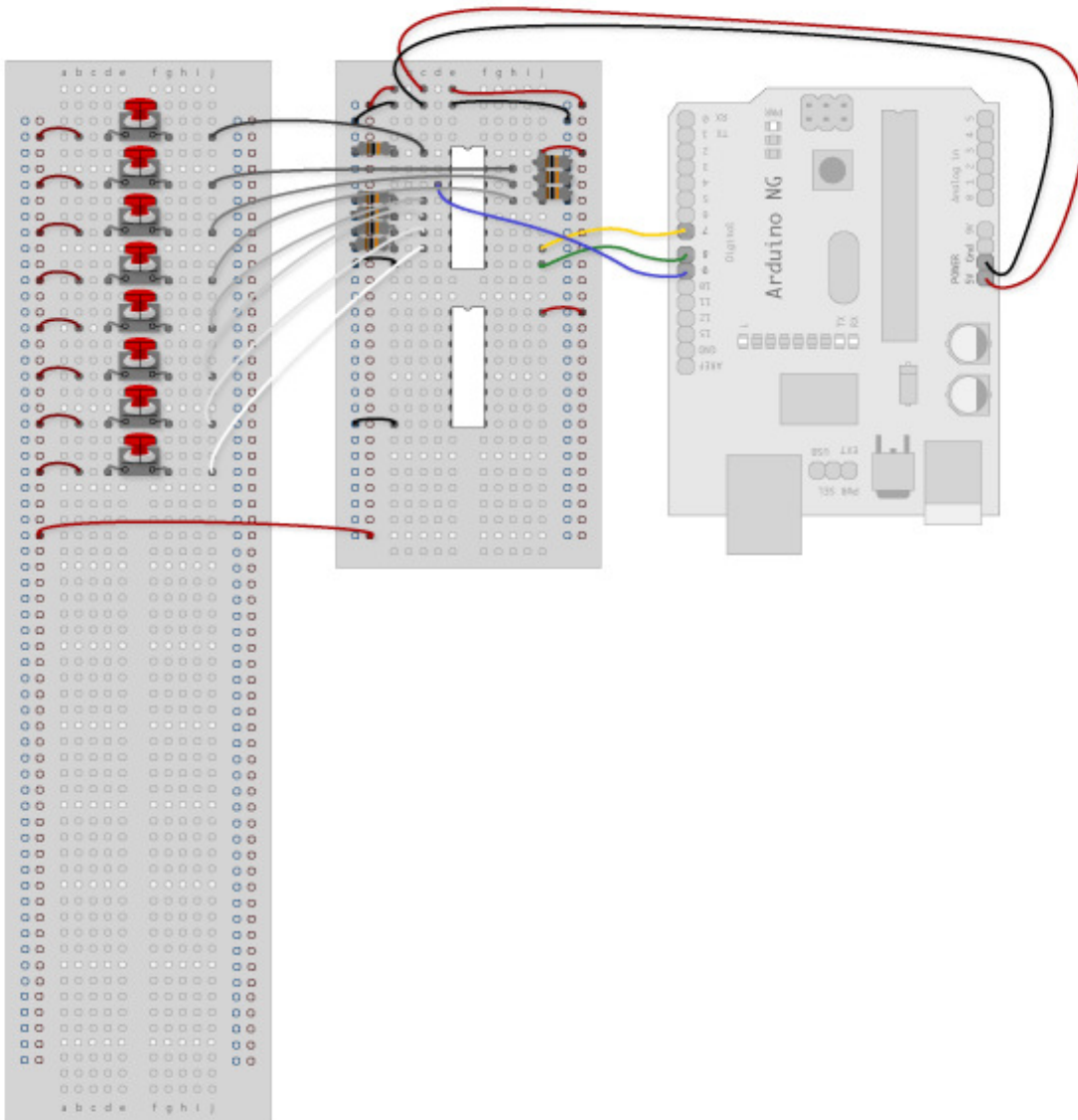
Code Sample 1.4 – Is it pressed? (sub-function)

Example 2: Daisy Chained

In this example you'll add a second shift register, doubling the number of input pins while still using the same number of pins on the Arduino.

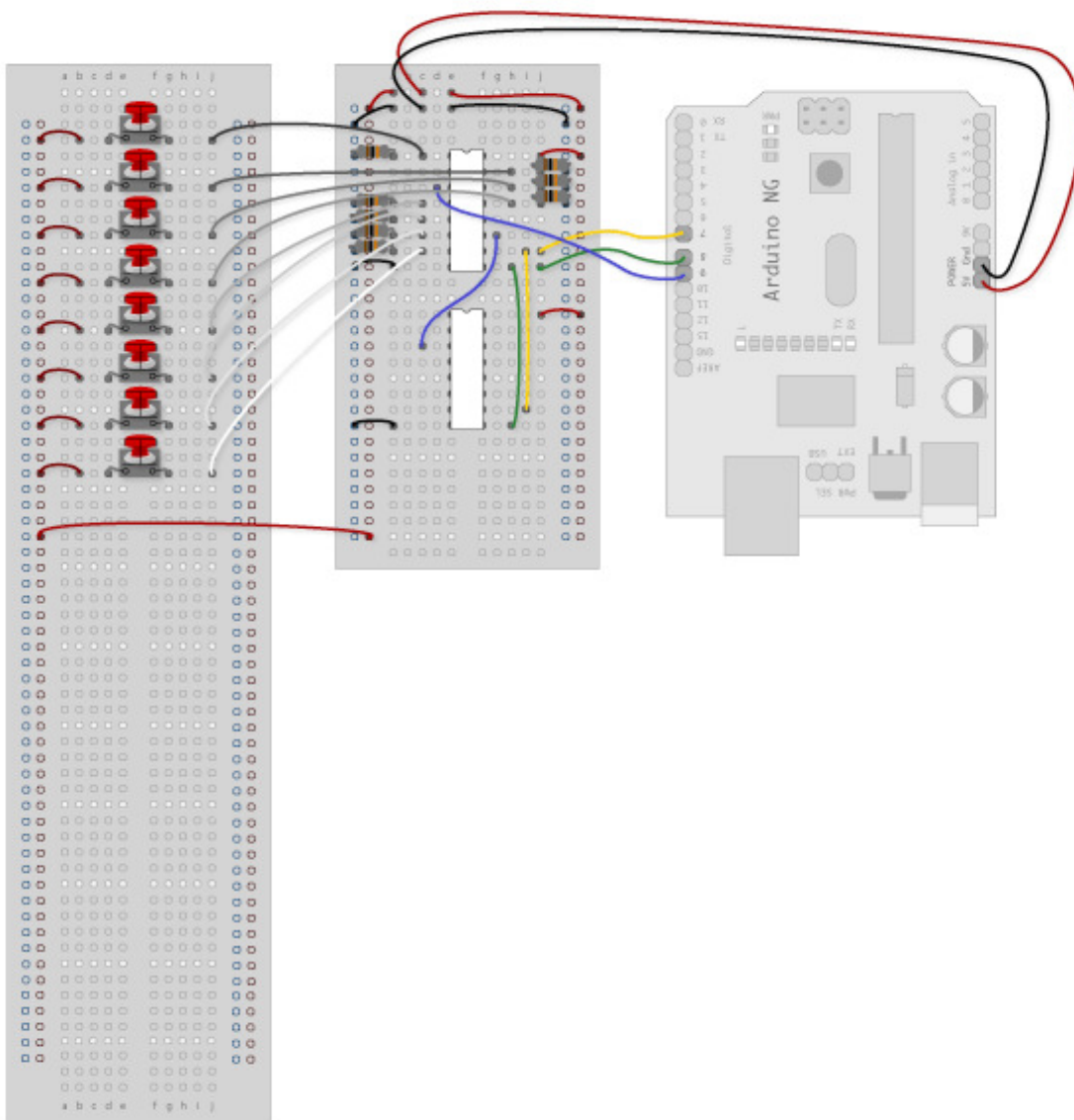
The Circuit

1. Add a second shift register.



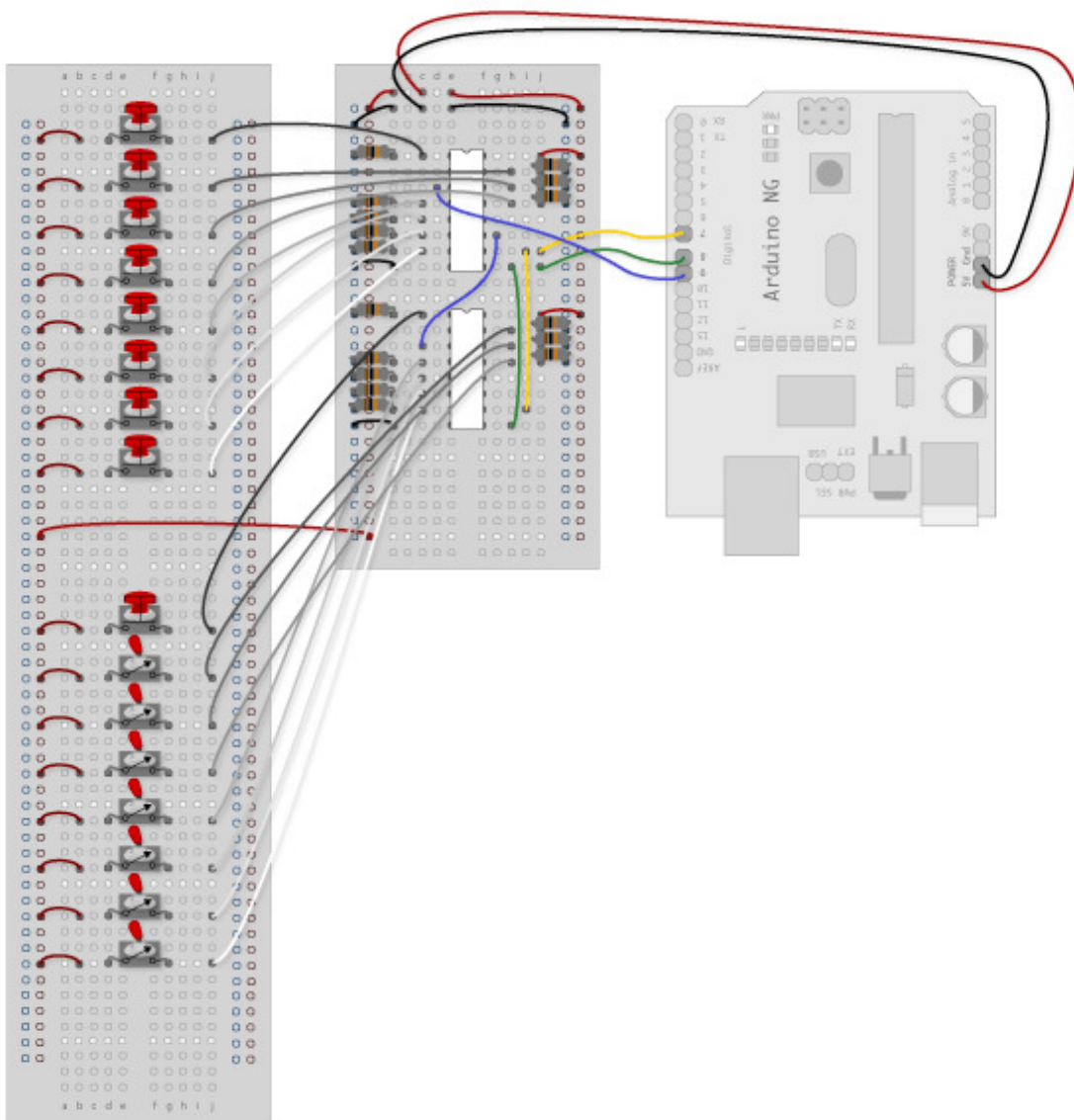
2. Connect the 2 registers.

Two of these connections simply extend the same clock and latch signal from the Arduino to the second shift register (yellow and green wires). The blue wire is going from the serial out pin (pin 9) of the first shift register to the serial data input (pin 14) of the second register.

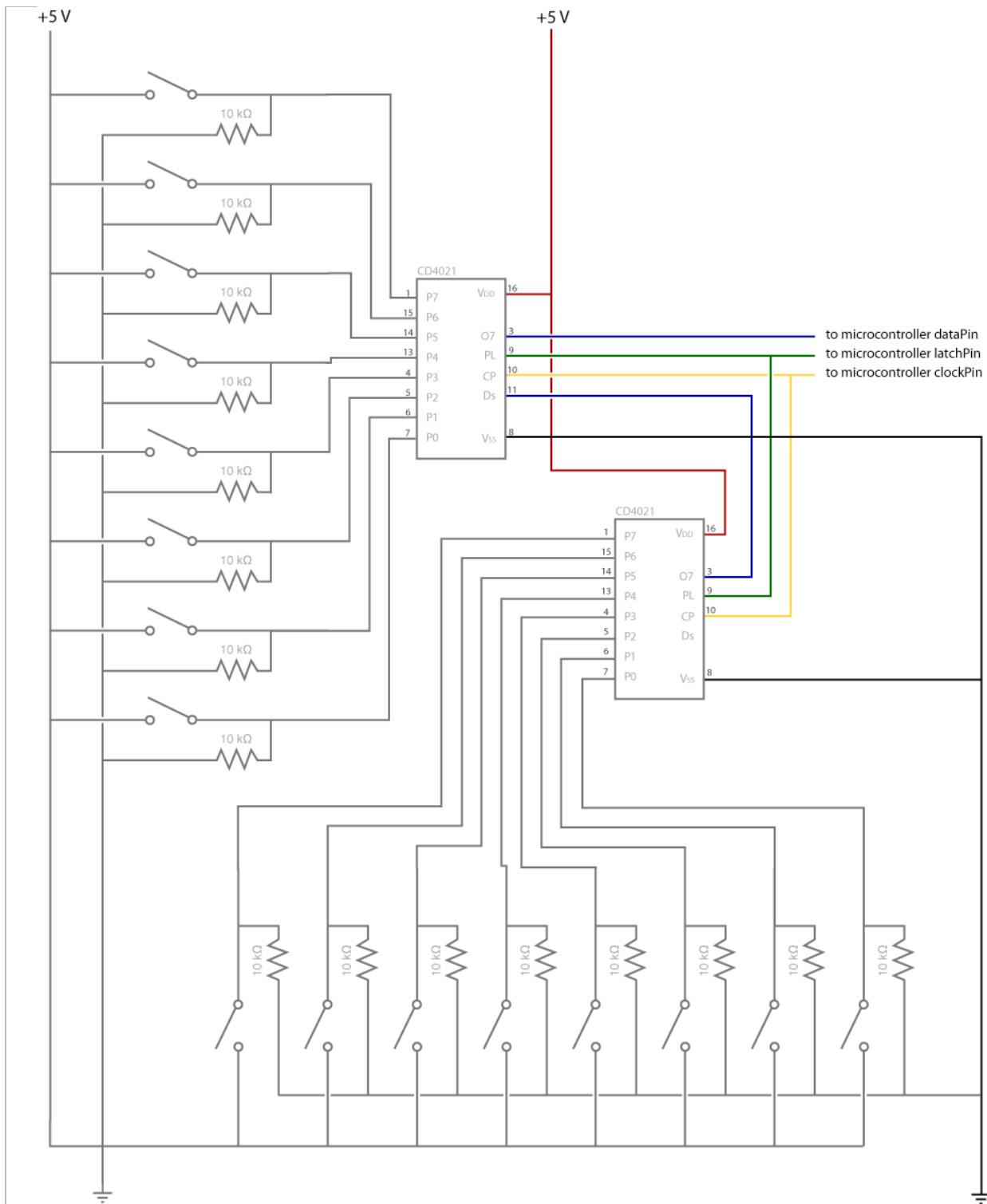


3. Add a second set of Switches.

Notice that there is one momentary switch and the rest are toggle switches. This is because the code examples will be using the switches attached to the second shift register as settings, like a preference file, rather than as event triggers. The one momentary switch will be telling the microcontroller that the setting switches are being changed.



Diagram



The Code

[Code Sample 2.1 – Hello World](#)

[Code Sample 2.2 – Using the second byte for settings, Print all](#)

[Code Sample 2.3 – Using the second byte for settings, Print on only \(uses sub-function\)](#)

Share |

[Edit Page](#) | [Page History](#) | [Printable View](#) | [All Recent Site Changes](#)