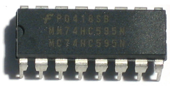
Project 16

Dual 8-Bit Binary Counters

Project 16 - Dual 8-Bit Binary Counters

In Project 16 we will daisy chain another 74HC595 IC onto the one used in Project 15 to create a dual binary counter.

What you will need

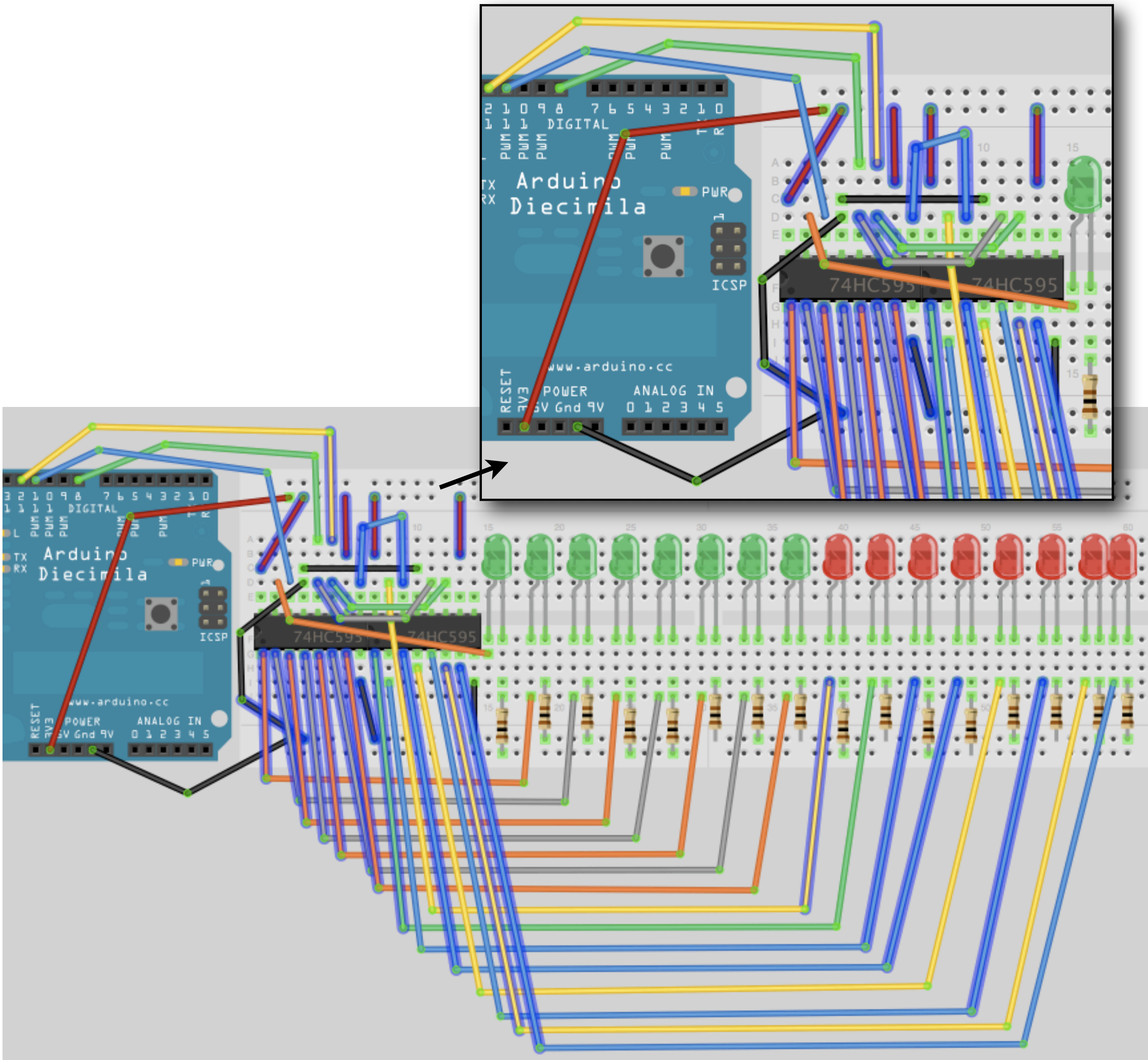
2 x 74HC595 Shift Registers	
8 x 240Ω Resistor	
8 x Red LED	

Connect it up

The first 595 is wired the same as in Project 15. The 2nd 595 has +5v and Ground wires going to the same pins as on the 1st 595. Then, add a wire from Pin 9 on IC 1 to Pin 14 on IC 2. Add another from Pin 11 on IC 1 to Pin 11 on IC 2 and Pin 12 on IC 1 to Pin 12 on IC 2.

The same outputs as on the 1st 595 going to the first set of LED's go from the 2nd IC to the 2nd set of LED's.

Examine the diagrams carefully.



Enter the Code

Enter the following code and upload it to your Arduino.

When you run this code you will see the Red set of LED's count up (in Binary) from 0 to 255 and the Green LED's count down from 255 to 0 at the same time.

```
// Project 16

//Pin connected to Pin 12 of 74HC595 (Latch)
int latchPin = 8;
//Pin connected to Pin 11 of 74HC595 (Clock)
int clockPin = 12;
//Pin connected to Pin 14 of 74HC595 (Data)
int dataPin = 11;

void setup() {
  //set pins to output
  pinMode(latchPin, OUTPUT);
  pinMode(clockPin, OUTPUT);
  pinMode(dataPin, OUTPUT);
}

void loop() {
  //count from 0 to 255
  for (int i = 0; i < 255; i++) {
    //set latchPin low to allow data flow
    digitalWrite(latchPin, LOW);
    shiftOut(i);
    shiftOut(255-i);
    //set latchPin to high to lock and send data
    digitalWrite(latchPin, HIGH);
    delay(250 );
  }
}

void shiftOut(byte dataOut) {
  // Shift out 8 bits LSB first,
  // on rising edge of clock

  boolean pinState;

  //clear shift register read for
sending data
  digitalWrite(dataPin, LOW);
  digitalWrite(clockPin, LOW);

  // for each bit in dataOut send
out a bit
  for (int i=0; i<=7; i++)  {
      //set clockPin to LOW prior to
sending bit
    digitalWrite(clockPin, LOW);

    // if the value of DataOut and
(logical AND) a bitmask
    // are true, set pinState to 1
(HIGH)
    if ( dataOut & (1<<i) ) {
      pinState = HIGH;
    }
    else {
      pinState = LOW;
    }

    //sets dataPin to HIGH or LOW
    //depending on pinState
    digitalWrite(dataPin, pinState);
    //send bit out on rising edge of clock
    digitalWrite(clockPin, HIGH);
    digitalWrite(dataPin, LOW);
  }

  //stop shifting

  digitalWrite(clockPin, LOW);
}
```

Project 16 - Code & Hardware Overview

The code for Project 16 is identical to that in Project 15 apart from the addition of

```
shiftOut(255-i);
```

In the main loop. The shiftOut routine sends 8 bits, to the 595. In the main loop we have put 2 sets of calls to shiftOut. One sending the value of I and the other sending 255-i. We call shiftOut twice before we set the latch to HIGH. This will send 2 sets of 8 bits, or 16 bits in total, to the 595 chips before the latch is set HIGH to prevent further writing to the registers and to output the contents of the shift register to the output pins, which in turn make the LED's go on or off.

The 2nd 595 is wired up exactly the same as the 1st one. The clock and latch pins are tied to the pins of the first 595. However, we have a wire going from Pin 9 on IC 1 to Pin 14 on IC 2. Pin 9 is the data output pin and pin 14 is the data input pin.

The data is input to Pin 14 on the 1st IC from the Arduino. The 2nd 595 chip is 'daisy chained' to the first chip by Pin 9 on IC 1, which is outputting data, into Pin 14 on the second IC, which is the data input.

What happens is, as you enter a 9th bit and above, the data in IC 1 gets shunted out of its data pin and into the data pin of the 2nd IC. So, once all 16 bits have been sent down the data line from the Arduino, the first 8 bits sent would have been shunted out of the first chip and into the second. The 2nd 595 chip will contain the FIRST 8 bits sent out and the 1st 595 chip will contain bits 9 to 16.

An almost unlimited number of 595 chips can be daisy chained in this manner.

Exercise

1. Re-create the Knight Rider light effect using all 16 LED's