```
//Some examples for the AM-2916 5V, Addressable LED strips http://www.andymark.com/product-p/am-2916.htm based on the new WS2812b chipset
//We ran this demo off of our AM-2287 Arduino Ethernet http://www.andymark.com/product-p/am-2287.htm
//http://arduino.cc/en/Main/ArduinoBoardEthernet
//For convenience, everything you need can be purchased in one kit here http://www.andymark.com/product-p/am-3010.htm

//The FastLED library we use here supports multiple chipsets
//This code requires that the fastspi library be put in your arduino\libraries folder
//Arduino info on how to install software libraries http://arduino.cc/en/Guide/Libraries
//AndyMark, Inc.
//CSK 12/3/2013, 3/17/2014, 3/20/2014, 2/12/2016, 7/1/2016


//***NOTE: This strip runs off of 5V MAX!!!. Applying much more than 5V will damage/destroy you LED strip!***
//***Handling note: Don't mess with the wiring while the power is on. This can cause voltage spikes ***
//***or sneak ground paths that can damage the LED strip ***

//DO NOT try to power the whole strip (150 LEDs) off the arduino 5v regulator.
//Use the AM-3068 10-30Vin to 5V 10A out stepdown converter http://www.andymark.com/product-p/am-3068.htm
//At full bright white, the strip can draw 4.5Amps or so.
//This would overheat or burnout the arduino regulator if you tried to drive it from the arduino only
//The BLACK wire is ground, RED is +5V, WHITE is data
//Make sure you connect the BLACK ground from the LED strip to the Arduino ground.
//Communications to the LEDs requires a common ground to work.

//If you are using the AndyMark AM-2297 Arduino Ethernet board then make sure
//you select Tools>Board>Arduino Ethernet from the Arduino IDE menu
//If you are new to working with Arduino a good place to start is here http://arduino.cc/en/Guide/HomePage
//Another new training resource provided by a 3rd party is here:
http://www.arduinoclassroom.com/index.php/arduino-101



//CSK 3/17/2013 Libraries new location
//https://github.com/FastLED/FastLED
//https://github.com/FastLED/FastLED/wiki/Overview

#include "FastLED.h"

#define COLOR_ORDER       GRB
#define MAX_BRIGHTNESS    255
//Tell it how many leds are in the strip. AndyMark's 2.5 meter strip has 150 leds
#define NUM_LEDS          265

// This is an array of leds. One item for each led in your strip
CRGB leds[NUM_LEDS];

//CSK 3/17/2014 I moved this to a pin that doesn't conflict with Ethernet functions in case you want to control LEDs via Ethernet
```

```
#define DATA_PIN        6 //White wire from the http://www.andymark.com/product-p/am-2917.htm power connector

//This function is used to setup things like pins, Serial ports etc.
//Here we specify which chipset our LEDs run off of by our choice of config function
void setup()
{

  // Uncomment one of the following lines for your leds arrangement.
  // FastLED.addLeds<TM1803, DATA_PIN, RGB>(leds, NUM_LEDS);
  // FastLED.addLeds<TM1804, DATA_PIN, RGB>(leds, NUM_LEDS);
  // FastLED.addLeds<TM1809, DATA_PIN, RGB>(leds, NUM_LEDS);
  //FastLED.addLeds<WS2811, DATA_PIN, RGB>(leds, NUM_LEDS);
  // FastLED.addLeds<WS2812, DATA_PIN, RGB>(leds, NUM_LEDS);
  //CSK 2/12/2016 This is the correct chipset for the am-2916 LED strip
  FastLED.addLeds<WS2812B, DATA_PIN, COLOR_ORDER>(leds, NUM_LEDS);
  // FastLED.addLeds<UCS1903, DATA_PIN, RGB>(leds, NUM_LEDS);

  //FastLED.addLeds<WS2801, RGB>(leds, NUM_LEDS);

  // FastLED.addLeds<SM16716, RGB>(leds, NUM_LEDS);
  // FastLED.addLeds<LPD8806, RGB>(leds, NUM_LEDS);

  //***This is the chipset in the AM-2640 LED strip***
  //CSK 3/17/2013 Changed to this function to allow direct data and clock pin specification
  //FastLED.addLeds<WS2801, DATA_PIN, CLOCK_PIN, RGB>(leds, NUM_LEDS);

  // FastLED.addLeds<SM16716, DATA_PIN, CLOCK_PIN, RGB>(leds, NUM_LEDS);
  // FastLED.addLeds<LPD8806, DATA_PIN, CLOCK_PIN, RGB>(leds, NUM_LEDS);
  FastLED.clear();
  FastLED.show();
  delay(250);
  //clear() turns all LEDs off
  FastLED.clear();
  FastLED.setBrightness(MAX_BRIGHTNESS);
  fill_solid( leds, NUM_LEDS /*number of leds*/, CRGB( 125, 125, 125) );
  FastLED.show();
  // start serial port at 9600 bps:
  Serial.begin(9600);
}

void loop()
{
  //This is kind of Arduino's equivalent to Main() in a standard C program
  //This, as the name implies, loops endlessly.
  //https://code.google.com/p/fastspi/wiki/CRGBreference
        FastLED.clear();
        FastLED.show();
        delay(500);
  //CSK 3/20/2014 I added a rainbow function just for grins
        rainbow(1);
        cylon(CRGB::Red, 1, 1);
```

```
   cylon(CRGB::Green, 1, 1);
   cylon(CRGB::Blue, 1, 1);
   color_chase(CRGB::Red, 5);
   color_chase(CRGB::DarkOrange, 5);
   color_chase(CRGB::Yellow, 5);
         color_chase(CRGB::Green, 5);
   color_chase(CRGB::Blue, 5);
         color_chase(CRGB::Violet, 5);
         missing_dot_chase(CRGB::White, 5);
         missing_dot_chase(CRGB::Red, 5);
         missing_dot_chase(CRGB::Yellow, 5);
         missing_dot_chase(CRGB::Green, 5);
         missing_dot_chase(CRGB::Cyan, 5);
         missing_dot_chase(CRGB::Blue, 5);
         missing_dot_chase(0x3000cc, 5) ;
}

//These are the functions we have defined to do chase patterns. They are actually called inside the loop() above
//They are meant to demonstrate things such as setting LED colors, controlling brightness
void color_chase(uint32_t color, uint8_t wait)
{
  FastLED.clear();
  //The brightness ranges from 0-255
  //Sets brightness for all LEDS at once
  FastLED.setBrightness(MAX_BRIGHTNESS);
  // Move a block of LEDs

  for(int led_number = 0; led_number < NUM_LEDS - 5; led_number++)
  {
    // Turn our current led ON, then show the leds
    leds[led_number] = color;
    //CSK 4/22/2016 Make it multiple dots on
    leds[led_number + 1] = color;
    leds[led_number + 2] = color;
    leds[led_number + 3] = color;
    leds[led_number + 4] = color;
    leds[led_number + 5] = color;

    // Show the leds (only one of which is has a color set, from above
    // Show turns actually turns on the LEDs
    FastLED.show();

    // Wait a little bit
    delay(wait);

    // Turn our current led back to black for the next loop around
    //CSK 4/22/2016 Turn the dots off
    leds[led_number] = CRGB::Black;
  }
  return;
}
```

```
//Move an "empty" dot down the strip
void missing_dot_chase(uint32_t color, uint8_t wait)
{
  //Step thru some brightness levels from max to 10. led_brightness/=2 is a cryptic shorthand way of saying
led_brightness = led_brightness/2
  //      for (int led_brightness = MAX_BRIGHTNESS; led_brightness > 10; led_brightness/=2)
  {
    //FastLED.setBrightness(led_brightness);
    //CSK 4/22/2016 Turn brightness down to save batteries since almost all leds are on
    FastLED.setBrightness(25);

    // Start by turning all pixels on:
    //for(int led_number = 0; led_number < NUM_LEDS; led_number++) leds[led_number] = color;
    //https://github.com/FastLED/FastLED/wiki/Controlling-leds
    fill_solid(leds, NUM_LEDS, color);

    // Then display one pixel at a time:
    for(int led_number = 0; led_number < NUM_LEDS - 5; led_number++)
    {
      leds[led_number] = CRGB::Black; // Set new pixel 'off'
      //CSK 4/22/2016
      leds[led_number + 1] = CRGB::Black; // Set new pixel 'off'
      leds[led_number + 2] = CRGB::Black; // Set new pixel 'off'
      leds[led_number + 3] = CRGB::Black; // Set new pixel 'off'
      leds[led_number + 4] = CRGB::Black; // Set new pixel 'off'
      leds[led_number + 5] = CRGB::Black; // Set new pixel 'off'
      if( led_number > 0 && led_number < NUM_LEDS)
      {
        leds[led_number-1] = color; // Set previous pixel 'on'
      }
      FastLED.show();
      delay(wait);
    }
  }
  return;
}

//Cylon - LED sweeps back and forth, with the color, delay and number of cycles of your choice
void cylon(CRGB color, uint16_t wait, uint8_t number_of_cycles)
{
  FastLED.setBrightness(MAX_BRIGHTNESS);
  for (uint8_t times = 0; times<=number_of_cycles; times++)
  {
    // Make it look like one LED is moving in one direction
    for(int led_number = 0; led_number < NUM_LEDS; led_number++)
    {
      //Apply the color that was passed into the function
      leds[led_number] = color;
      //Actually turn on the LED we just set
      FastLED.show();
      // Turn it back off
      leds[led_number] = CRGB::Black;
```

```
      // Pause before "going" to next LED
      delay(wait);
    }

    // Now "move" the LED the other direction
    for(int led_number = NUM_LEDS-1; led_number >= 0; led_number--)
    {
      //Apply the color that was passed into the function
      leds[led_number] = color;
      //Actually turn on the LED we just set
      FastLED.show();
      // Turn it back off
      leds[led_number] = CRGB::Black;
      // Pause before "going" to next LED
      delay(wait);
    }
  }
  return;
}

void rainbow(uint8_t wait)
{

  uint16_t hue;
  FastLED.clear();

  for(hue=10; hue<255*3; hue++)
  {

    fill_rainbow( &(leds[0]), NUM_LEDS /*led count*/, hue /*starting hue*/);
    FastLED.show();
    delay(wait);
  }
  return;
}
```