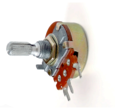
Project 6

Interactive LED Chase Effect

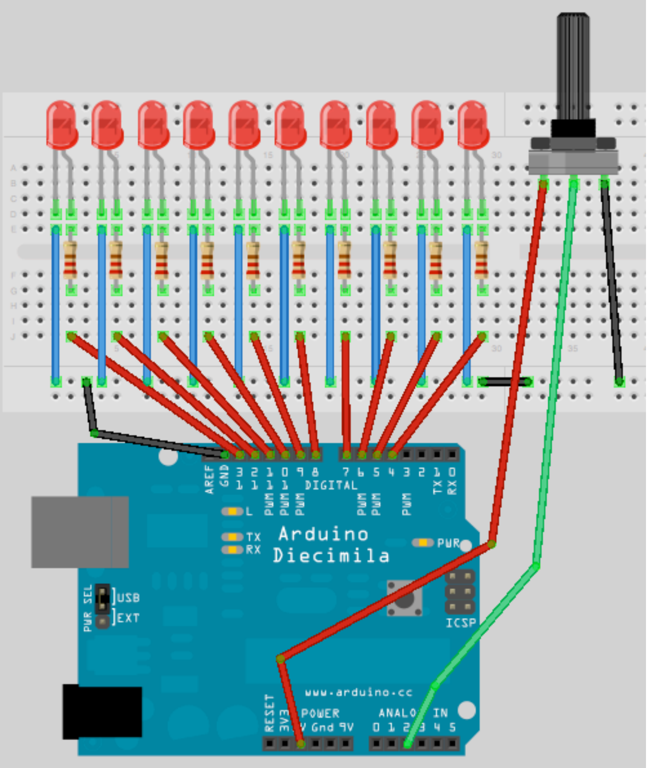
Project 6 - Interactive LED Chase Effect

We are now going to use a string of LED's (10 in total) to make an LED chase effect, similar to that used on the car KITT in the Knightrider TV Series and on the way introduce the concept of arrays.

What you will need

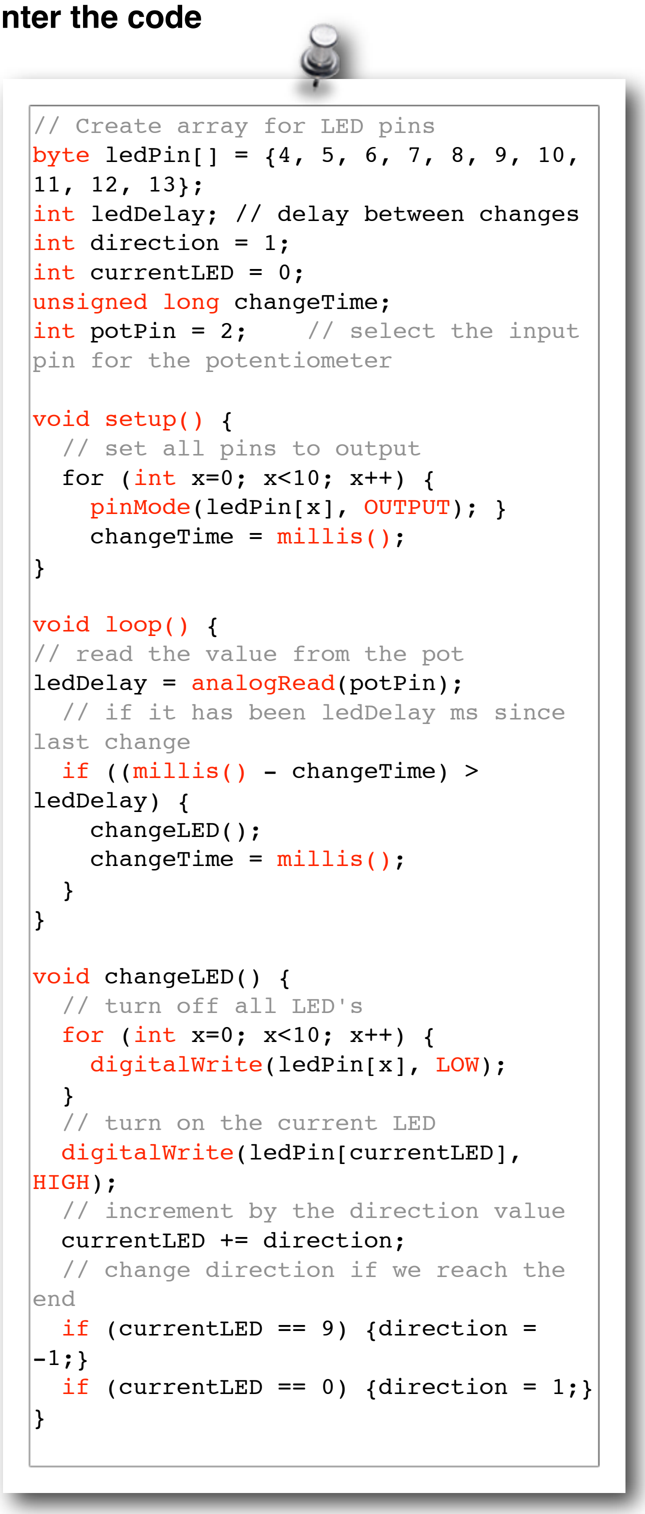
Parts from previous project plus....	
4K7 Potentiometer	

Connect it up



This is the same circuit as in Project 5, but we have simply added the potentiometer and connected it to 5v, Ground and Analog Pin 5.

Enter the code



```
// Create array for LED pins
byte ledPin[] = {4, 5, 6, 7, 8, 9, 10,
11, 12, 13};
int ledDelay; // delay between changes
int direction = 1;
int currentLED = 0;
unsigned long changeTime;
int potPin = 2;     // select the input
pin for the potentiometer

void setup() {
  // set all pins to output
  for (int x=0; x<10; x++) {
    pinMode(ledPin[x], OUTPUT); }
    changeTime = millis();
}

void loop() {
// read the value from the pot
ledDelay = analogRead(potPin);
  // if it has been ledDelay ms since
last change
  if ((millis() - changeTime) >
ledDelay) {
    changeLED();
    changeTime = millis();
  }
}

void changeLED() {
  // turn off all LED's
  for (int x=0; x<10; x++) {
    digitalWrite(ledPin[x], LOW);
  }
  // turn on the current LED
  digitalWrite(ledPin[currentLED],
HIGH);
  // increment by the direction value
  currentLED += direction;
  // change direction if we reach the
end
  if (currentLED == 9) {direction =
-1;}
  if (currentLED == 0) {direction = 1;}
}
```

This time when verify and upload your code, you should now see the lit LED appear to bounce back and forth between each end of the string of lights as before. But, by turning the knob of the potentiometer, you will change the value of `ledDelay` and speed up or slow down the effect.

Let's take a look at how this works and find our what a potentiometer is.

Project 6 - Code Overview

```
// Create array for LED pins
byte ledPin[] = {4, 5, 6, 7, 8, 9, 10, 11,
12, 13};
int ledDelay; // delay between changes
int direction = 1;
int currentLED = 0;
unsigned long changeTime;
int potPin = 2;     // select the input pin
for the potentiometer

void setup() {
  // set all pins to output
  for (int x=0; x<10; x++) {
    pinMode(ledPin[x], OUTPUT); }
    changeTime = millis();
}

void loop() {
// read the value from the pot
ledDelay = analogRead(potPin);
  // if it has been ledDelay ms since last
change
  if ((millis() - changeTime) > ledDelay) {
    changeLED();
    changeTime = millis();
  }
}

void changeLED() {
  // turn off all LED's
  for (int x=0; x<10; x++) {
    digitalWrite(ledPin[x], LOW);
  }
  // turn on the current LED
  digitalWrite(ledPin[currentLED], HIGH);
  // increment by the direction value
  currentLED += direction;
  // change direction if we reach the end
  if (currentLED == 9) {direction = -1;}
  if (currentLED == 0) {direction = 1;}
}
```

The code for this Project is almost identical to the previous project. We have simply added a potentiometer to our hardware and the code has additions to enable us to read the values from the potentiometer and use them to adjust the speed of the LED chase effect.

We first declare a variable for the potentiometer pin

```
int potPin = 2;
```

as our potentiometer is connected to analog pin 2. To read the value from an analog pin we use the `analogRead` command. The Arduino has 6 analog input/outputs with a 10-bit analog to digital convertor (we will discuss bits later on). This means the analog pin can read in voltages between 0 to 5 volts in integer values between 0 (0 volts) and 1023 (5 volts). This gives a resolution of 5 volts / 1024 units or 0.0049 volts (4.9mV) per unit.

We need to set our delay using the potentiometer so we will simply use the direct values read in from the pin to adjust the delay between 0 and 1023 milliseconds. We do this by directly reading the value of the potentiometer pin into ledDelay. Notice that we do not need to set an analog pin to be an input or output like we need to with a digital pin.

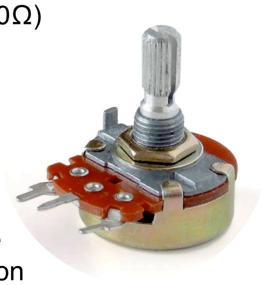
```
ledDelay = analogRead(potPin);
```

This is done during our main loop and therefore it is constantly being read and adjusted. By turning the knob you can adjust the delay value between 0 and 1023 milliseconds (or just over a second) and therefore have full control over the speed of the effect.

OK let’s find out what a potentiometer is and how it works.

Project 6 - Hardware Overview

The only additional piece of hardware used in this project was the 4K7 (4700Ω) potentiometer.



You have already come across a resistor and know how they work. The potentiometer is simply an adjustable resistor with a range from 0 to a set value (written on the side of the pot). In the kit you have been given a 4K7 or 4,700Ω potentiometer which means it's range is from 0 to 4700 Ohms.

The potentiometer has 3 legs. By connecting up just two legs the potentiometer becomes a variable resistor. By connecting all 3 legs and applying a voltage across it, the pot becomes a voltage divider. This is how we have used it in our circuit. One side is connected to ground, the other to 5v and the centre pin to our analog pin. By adjusting the knob, a voltage between 0 and 5v will be leaked from the centre pin and we can read the value of that voltage on Analog Pin 2 and use it's value to change the delay rate of the light effect.

The potentiometer can be very useful in providing a means of adjusting a value from 0 to a set amount, e.g. the volume of a radio or the brightness of a lamp. In fact, dimmer switches for your home lamps are a kind of potentiometer.

Exercises

1. Get the LED's at BOTH ends of the strip to start as on, then to both move towards each other, appear to bounce off each other and then move back to the end.

2. Make a bouncing ball effect by making the LED start at one end, 'drop' toward the other end, bounce back up, but to only go up 9 spaces, bounce, go up 8 spaces, then 7, then 6, etc. To give the effect it is a bouncing ball, getting bouncing up to a lower height on each bounce.