

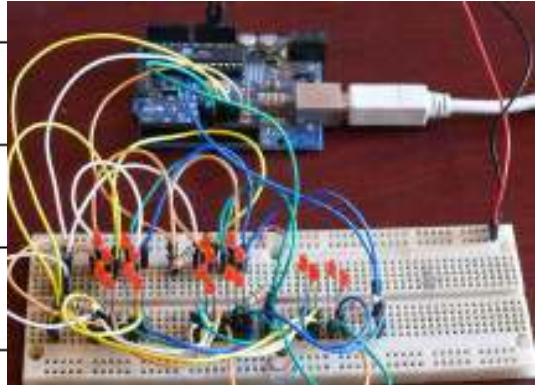
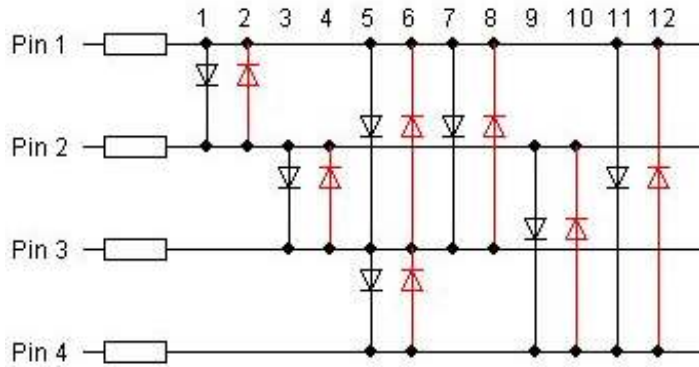


Doc in french under www.didel.com/diduo/Charlie.pdf

Kit Charlie for Arduino

The Charlieplexing trick is to use the input mode (floating) in order to increase the number of leds one can connect on a given number of pins. Several web links are available.

<https://en.wikipedia.org/wiki/Charlieplexing> The images below are taken from [riaancornelius](http://www.instructables.com/id/Controlling-20-Leds-from-5-Arduino-pins-using-Cha/)
<http://www.instructables.com/id/Controlling-20-Leds-from-5-Arduino-pins-using-Cha/>



Like this, it looks difficult to understand. Let us do the initial steps, assuming you know how to control a pin as an output and input (this mean floating, no action).

Let us consider 2, then 3 leds and have a look at the truth table..

What is this weak green mentioned at the bottom of the table? If the voltage is greater than 3.6V with pin3 high, pin2 floating and pin 1 low, there is enough voltage to start lighting the D1 and D2 green leds, serially connected (1.8V threshold).

If several leds are on, the light intensity varies since their currant must go through shared resistors.

For these reasons, one selects one led at a time, and scans the display to have several leds on within a 20ms frame.

Resistors are usually 200Ohm, in order to limit voltage drop and heat inside the microcontroller.

Pin2	Pin1	D1
0	0	off
0	1	green
1	0	red
1	1	off
in	0	off
in	1	offt
0	in	off
1	in	off
in	in	off

Pin3	Pin2	Pin1	D3 1-3	D2 2-3	D1 1-2
0	0	0	off	off	eteint
0	0	1	off	off	green
0	1	0	off	green	rouge
0	1	1	green	vert	off
1	0	0	rouge	rouge	off
1	0	1	eteint	rouge	green
1	1	0	rouge	eteint	rouge
1	1	1	off	off	off
in	0	0	off	off	off
in	0	1	off	off	vert
.	.	.			
0	in	1	green	weak green	weak green

Didel module has an interesting pedagogical aspect: the led arrangement makes it clear when you work with 2, 3, 4, 5 lines.

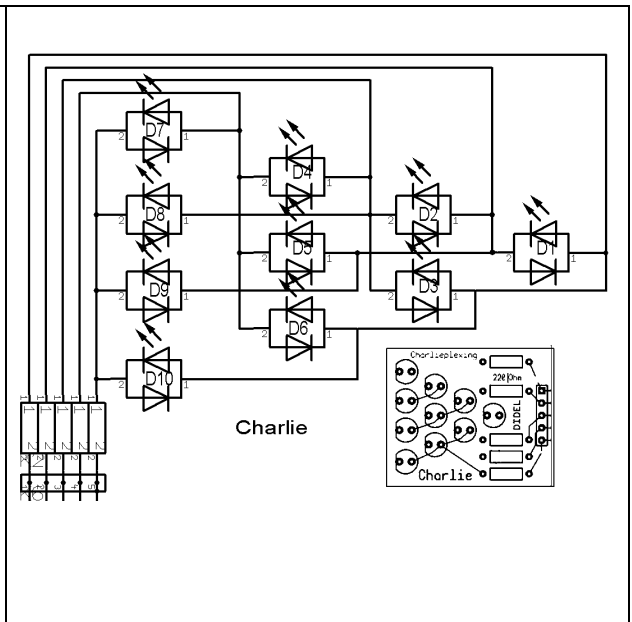
The kit Charlie for Arduino is powered by the control lines. Any microcontroller that can sink and source a currant of 20mA can be used. On the Arduino, the module can be connected in different ways on ports B,C,D. A board can be put upside down – guess what will be displayed !



The interest now is to define how to program. Lighting several leds at the same time is tricky and not all configurations are possible. Lighting one led at a time and avoiding flickering is also tricky. This makes the learning value of the Charlie kit.

Arduino test program

```
//TestCharlie.ino PortB bits 4 3 2 1 0 led13 bit 5
//          Charlie kit lines 1 2 3 4 5
//          D1 1>>2 green 2>>1 red
void setup () {
  DDRB = 0; // all in
  byte cx1;
void Led (byte biton, byte bitoff, int del) {
  DDRB = 0;
  PORTB = (1<<biton);
  DDRB = (1<<biton)|(1<<bitoff); // both lines active
  if (del) { delay (del) ;}
}
void Off (int del) { // all lines input
  DDRB = 0;
  if (del) { delay (del) ;}
}
#define tg 1 // green time on (efficient)
#define tr 2 // red time on (not so good)
void loop () {
  Led (3,4,tg); Led (2,4,tg); Led (2,3,tg);
  Off(0);
  Led (4,3,tr); Led (4,2,tr); Led (3,2,tr);
  Off(0);
}
}
```



The function `Led(,,)` sets one led at a time. The function `Off()` is used after the ON time and sets all lines as inputs, that is no led ON. This allows a compact code (if microseconds are used) for avoiding a visible blinking.

Charlie for Raspberry, Nucleo, nRF51 development boards

Output amplifiers take an huge space on a chip and modern microcontrollers prefer to put more transistors on the chip area. The kit Charlie for Raspberry includes currant amplifiers to be compatible. See www.didel.com/kits/CharlieRaspi.pdf