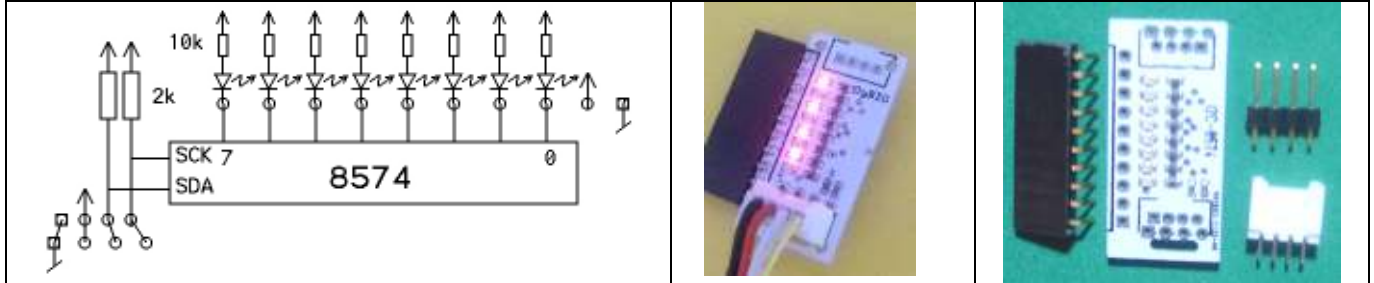


Dg8IO

I2C 8754A 8 IO Expander

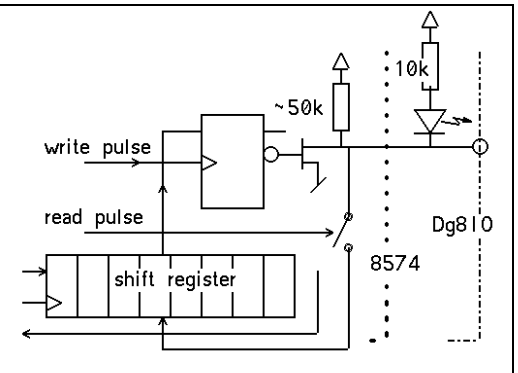


		<p>10-pins connector signals: gnd +5V 8 bits, LSB first.</p> <p>Dg8IO leds are ON if signal is LOW. On the picture, output byte is 0b00101100 = 0x2C</p>
<p>Need 16 I/O?</p>		

Pas de documentation en français prévue.

The bidirectional port of the 8754A is not similar to a microcontroller port, where the direction register decides if a pin is an input or output.

On the 8754A, the serial data send to the I2C register is inverted to control a transistor with a pull-up resistor. It is an open-collector output. A zero is active, the led on Dg8IO is on. A one is not active, that is if you program a one, you can set the voltage from outside without any risk of short. When you ask the I2C to read, it reads all pins. It is up to the programmer to remember that if a one is programmed, it may be because one have an input and the value read is 0 or 1. If it is an output, just be aware it is a weak signal you cannot charge.



This example shows how simple it is to write 8 bits on the Dg8IO using the Wire library, standard on Arduino (no installation).

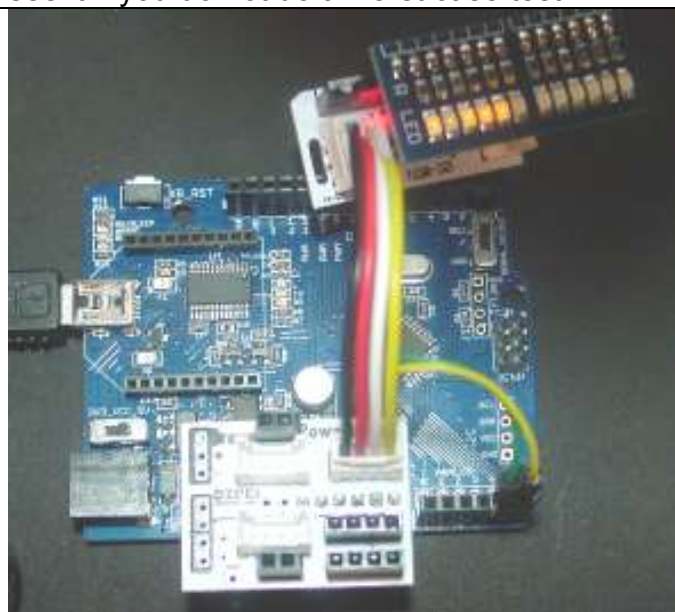
Do not be confused. The P8574 respond to 7-bit address 0x20 (32) while the P8754A used here has the address 0x38 (56).

```
// Dg8ioTestWrite.ino Count
#include <Wire.h>
#define P8574 56 // 0x70/2 for 8754A
void setup() {
  Wire.begin(); // I2C
}
byte data;
void loop() {
  Wire.beginTransmission(P8574);
  Wire.write(byte(data));
  Wire.endTransmission();
  data++;
  delay(100);
}
```

Reading is as easy but on have to prepare the output register with "ones"

```
// Dg8ioTestRead.ino read
#include <Wire.h>
#define P8574 56 // 0x70/2
void setup() {
  Wire.begin(); // I2C
  Serial.begin(9600);
  // write ones on output buffer
  Wire.beginTransmission(P8574);
  Wire.write(byte(0xFF));
  Wire.endTransmission();
}
byte data;
void loop() {
  Wire.requestFrom(P8574,1); // ask 1 byte
  while(Wire.available()) {
    data = Wire.read();
    Serial.print(data,HEX);
  }
  delay (1000);
}
```

The Boxtec Arduled connects 12 leds with the return of current on a single Gnd pin. Due to an Arduino idiosyncrasy, you can insert 7 pins of the Arduled into the Dg8IO connector. Knowing a little bit of hardware, you can make it work without a connection to Gnd. The resistors are 2k2, current "on" is hence $(5-1.5)/2.2 = \sim 2\text{mA}$. 6 leds on is 12 mA. The 8574 is good for sinking current, so just program the pin that has the Arduled Gnd signal, it will be a good enough Gnd. The leds may get 0.2V less voltage when all on, but you will not see it if you do not do a worst case test.



```
// TestI2CCompteArduLed.ino
// Arduled Gnd pin on 4th pin,
// Dg8IO connector is 3 pins unused, then
// Gnd, next 6 counts

#include <Wire.h>
#define P8574 56 // 0x70/2
void setup()
{
  Wire.begin(); // join i2c bus
  // (address optional for master)
  Serial.begin(9600); // start serial for
  // output
}
byte cnt,data;
void loop() {
  cnt++;
  data = cnt << 2; // 3 low bits are 0 Gnd
  Wire.beginTransmission(P8574); //Ad1307);
  Wire.write(byte(data));
  Wire.endTransmission();
  delay(100);
}
```

The schematic shows that the Leds are in serie with the pull-ups the circuit requires. The Leds are on when the signal (input or output) is zero.

