CH-1092 Belmont,  Switzerland
Tel +41 21 728-6156,  Fax 728-6157
Email info@didel.com,  www.didel.com

www.didel.com/Boree.pdf

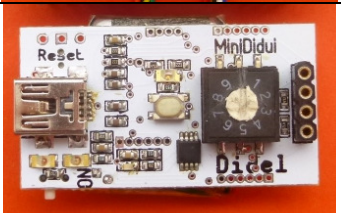
# Boree  DiduiPoRot

*Boree was the god of wind (rotswitch looks like a compass)*

*≪ under work ≫*

The Boree is suitable for many applications that require small size and autonomy.

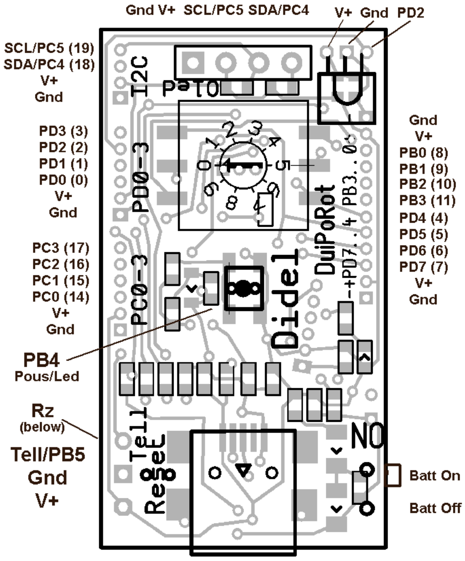
- Smaller connectors, but the Molex cables makes it easy to connect your devices.
- Lipo of any size and built-in LiPo charger
- Connector to add a cheap Oled, 64x128 or 32x128.
- Three-pin connector for a strip of WS2801 or WS2802 smart Leds.
- Built-in 16-position rotary switch, convenient to select a program and/or parameters.
- Led on pin 13 as usual, but with a convenient push-button in parallel.

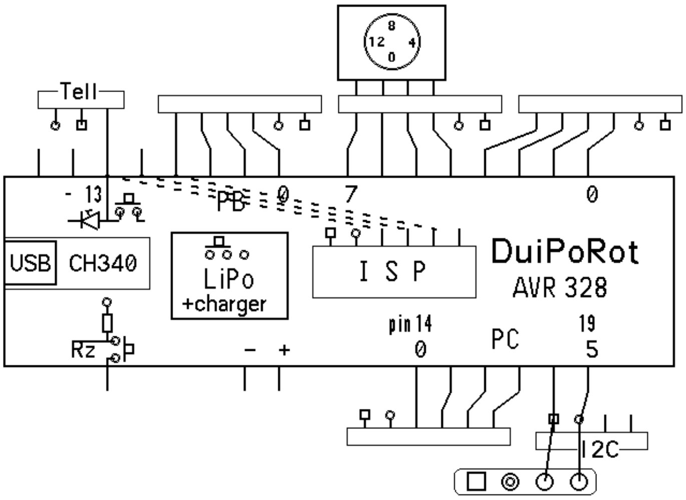


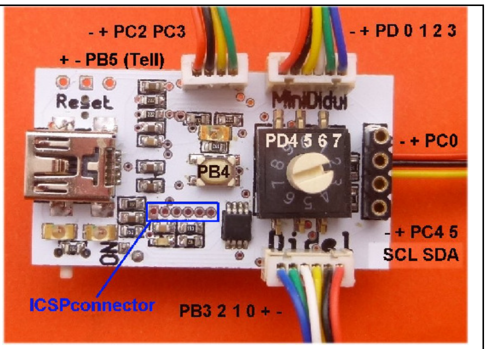
The DuiPo is compatible Arduino Duemilanove and include on request a 200mAh LiPo. Any Lipo is easy to solder.
Remember that you will develop the software at 5V and execute in autonomous mode from 4V down to 3.3V. Our experience (Xbot robot uses many sensors) is that very many components are compatible, but you have to check.

AVR328 pins

| Pin | Port | | Pin | Port | |
|---|---|---|---|---|---|
| 0 | PD0 | Rx PdConn | 10 | PB2 | PbConn |
| 1 | PD1 | Tx   id | 11 | PB3 | PbConn |
| 2 | PD2 | Int0  idD | 12 | PB4 | Pous/Led |
| 3 | PD3 | Int1  id | 13 | PB5 | Tell conn |
| 4 | PD4 | RotSw 1 | 14 | PC0 | RGB conn |
| 5 | PD5 | RotSw 2 | 15 | PC1 | nc |
| 6 | PD6 | RotSw 4 | 16 | PC2 | PcConn |
| 7 | PD7 | RotSw 8 | 17 | PC3 | PcConn |
| 8 | PB0 | PbConn | 18 | PC4 | I2C conn SCL |
| 9 | PB1 | PbConn | 19 | PC5 | I2C conn SDA |

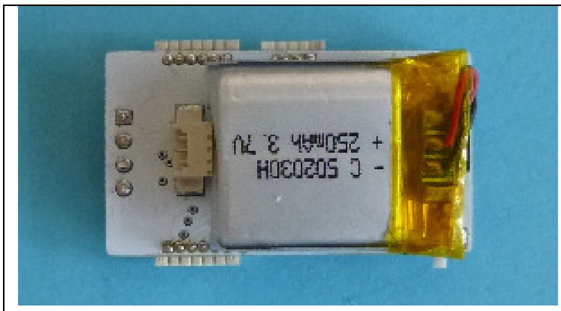


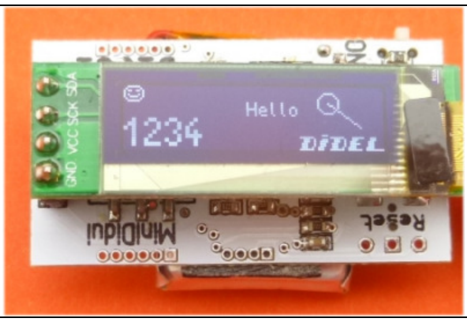


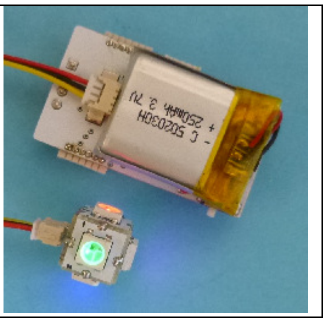


A typical application is to control a wearable strip of RGB leds or the Didel MiniCube.

Many applications that collect data need a control display. Use the PC23 connector for analogues signals, the I2C connector for smart sensors, Insert directly an Oled SSD1306 on its connector and you have a nice data acquisition system in a match box







.

## Specifications

AVR328 microcontroller 16 MHz, reset push button.
CH340 USB circuit. Easy to install, fast to be recognized at start-up.
Push button and Led on pin 13.. Pin13 is also available on the Tell connector to make debugging easy.
Rotary switch.(16 positions).
One 4-pin SIL connector for SSD1306, tree Molex connectors for AVR I/O pins

## Definition files

The files below and demo programs are available on http://www.didel.com/DuiPoRot.zip
Main definition file is DuiPoRot.h and include the instructions detailed below. We provide also tha Arduino way you may be more familiar with.

### Led (ignore the push-button).

AVR PORTB bit 45 or Arduino Pin 12

```
#define bLed 4   // Port B
#define  LedOn  bitSet   (PORTB,bLed)
#define  LedOff bitClear(PORTB,bLed)

void SetupLed { bitSet (DDRB,bLed); }
```

```
#define Led 12
#define  LedOn  digitalWrite (Led,HIGH)
#define  LedOff digitalWrite (Led,LOW)

void SetupLed { pinMode (Led,OUTPUT); }
```

### Led and Pushbutton

AVR PORTB bit 5 or Arduino Pin 13
By default, the pin is in Led mode (output). Depressing the switch in this mode does not lit the Led due to the protecting resistor.
In order to read the switch, one need to set the Input mode. The Led is On if the button is depressed. To make it easy, a macro (or a function) do 4 things: set the PushMode, wait few microseconds till the state is stabilized, read the switch, set the LedMode. This takes time, but human depressing switches are very slow!.

C

```
#define bLP 4  // LED/Push bit on PORTB
#define PushMode  bitSet (DDRB,bLP)
#define LedMode   bitClear  (DDRB,bLP)
#define  LEDOn    LedMode;  bitSet  (PORTB,bLP)
#define  LEDOfff  LedMode; bitClear(PORTB,bLP)
#define  Push  PushMode; Del60; PINB & (1<<bLP)
void SetupLed { bitSet (DDRB,bLed); }
```

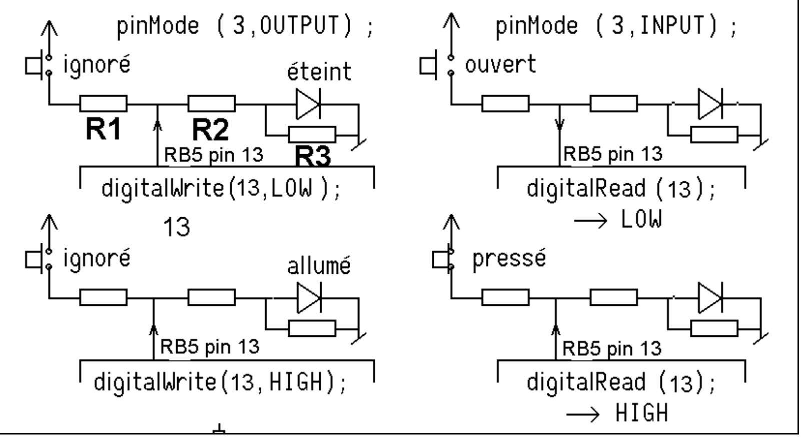
Arduino

```
#define LP 12 // LED/Push pin
#define PushMode  pinMode (LP,INPUT)
#define LedMode   pinMode (LP,OUTPUT)
#define  PushOn    PushMode; delayMicroseconds(60); digitalRead(LP);
#define  LedOn     LedMode; digitalWrite (LP,HIGH)
#define  LedOff    LedMode; digitalWrite (LP,LOW)
void SetupLed { pinMode (Led,OUTPUT); }
```

The following point may be skipped, a delay solve the problem, the library is same. Explaining the reason of this delay shows how important it is to understand levels and parasistic capacitances while handling low level instructions.  It is a good opportunity to anlyse systematically the states of such a simple system.

Pin 12 set as output controls the LED, switch is ignored.
Pin 12 set as input allows to read the switch state.
R1 limit the current if the switch is pressed when pin12 is LOW
R2 set the current in light
R3 is a high value resistor that improves the zero level in input mode.

See how to count push pulses on
www.didel.com/diduino/PushButton.pdf



Also, what happen is you write `LedOff; if(Push) { short action} LedOff;`?
If button is depressed, Led will be ON only for the duration of the `if{}` instruction, less than 1 microsecond. It is not a hardware bug! How to test if you believe it is a bug?

### Rotary switch

The 4 bits of the rotary switch are in binary order on PORTD, bits 4 to 7 for weight 1 2 4 8.
Signals on the pins are active low, software pull-up are required.
Note that if the rotary switch is not soldered or left on position zero, Pins 4 to 7 are available on connD connector can be used for any purpose.

C

```
#define MasRotSw 0b11110000
void SetupRotSw () { DDRD |= MasRotSw; PORTD |= MasRotSw; }

byte GetRotSw() { return  (~PIND>>4) & 0xF; }
```

Arduino

```
#define Rot1 4  // line of weight 1 on PD4
#define Rot2 5
#define Rot4 6
#define Rot8 7
void SetupRotSw () {
  pinMode (Rot1, INPUT_PULLUP);
  pinMode (Rot2, INPUT_PULLUP);
  pinMode (Rot4, INPUT_PULLUP);
  pinMode (Rot8, INPUT_PULLUP);
byte GetRotSw() {
  byte dd=0;
  if (digitalRead (Rot1,LOW) { dd+=1; }
  if (digitalRead (Rot2,LOW) { dd+=2; }
  if (digitalRead (Rot4,LOW) { dd+=4; }
  if (digitalRead (Rot8,LOW) { dd+=8; }
  return dd;
}
```

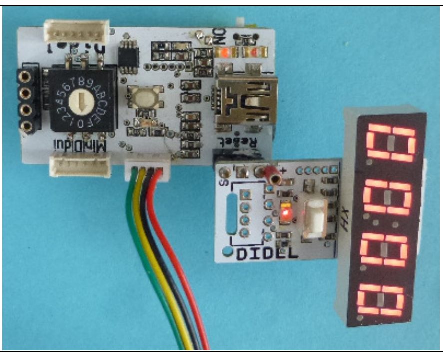
Arduino works at the pin level, with the advantage is wiring flexibility at the expense of programme size and execution time.

The GetRotSw() function is used in a switch/case statement, making the program clear and easy to modify.

```
  switch (GetRotSw()) {
    case 0:
     …
    break;
    case 1:
     …
    break;
.. etc
    default:
      // error, blink to show "no test"
  }  // end switch
```

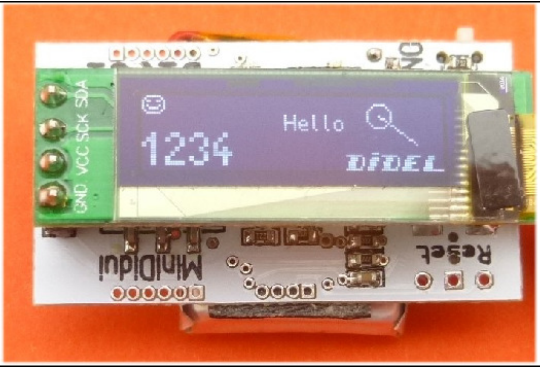
### Tell connector

We at Didel cannot design any software without having a Tell dislay connected. There are always analog value, counters to display, and the Arduino Terminal is not convenient, and not availablein autonomous mode..
Tell share the Led on pin13, you can even blink in the same time.
See https://www.didel.com/diduino/DiTell.pdf
for details and get a Tell from Boxtec, Tindie or Didel.



**I2C connector for SSD1306**

The Oled SSD 1306, 32x128 or 64x128 are so cheap and so easy to use, why do without it ?
Since it is I2C, one can connect any I2C sensor or display. Just check the pinout and the voltage.
We will find a corner to add a Molex I2C connector on serie PCBs.
Our doc is easy to click on
https://www.didel.com/Oled.html

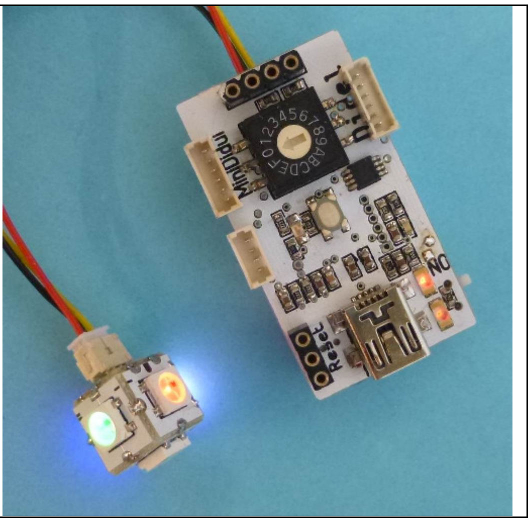


**3 and 4-pin connector** on PC0 and PC2 PC3

These 2 connectors are powered and can receive analogue channels.
The 3-pin uses a 3-pin Molex and is below the circuit to make the thickness minimum in case of wearable applications

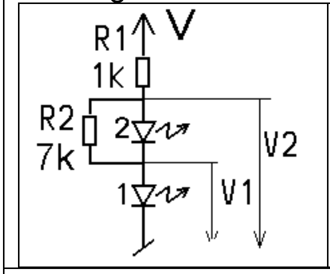
Our doc on strips and cube:
https://www.didel.com/affichages/rgb-strips/

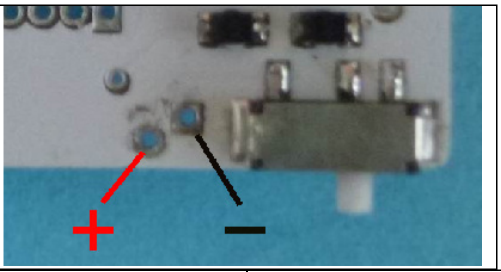
The 4-pin connector is on top.

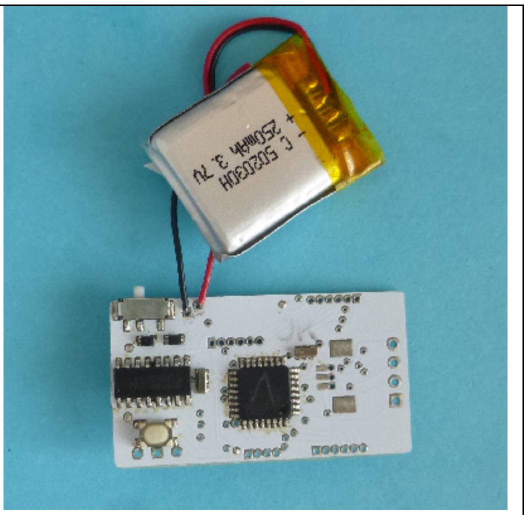


**Lipo connection**

The connection point are close to the switch. Square pads are always Gnd.
Charge current is set at ~30mA, automatic cut-off when The Lipo voltage exceed 4.2V followed by trickle.
One of the two LED **is Off** at 3V and **On** at 3.7V. It is not precise, but at a glance, you know if you need to recharge.

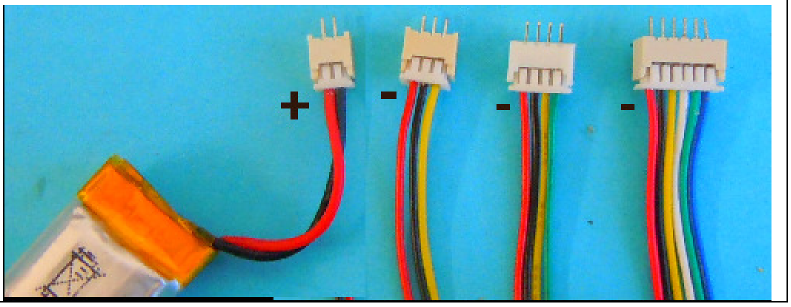






**Warning : Molex cable harness**

Pin 1 in most designs is Gnd and we apply that rule with inevitable exceptions. Molex has a red wire on pin 1 and black on pin 2. This is not what we would have liked.
The worse is Lipos refer to the color, pin1 is +, pin2 is -. We have to follow that anomaly. Be carefull.
Our other exception is the Tell connector.



jdn 170711 /180519