



It is a fully compatible Arduino board, using as the Arduino Mini an external USB/Serial adapter. A Bahoma Lipo power the Witty in a snap, no hard to insert small connector. Voltage is monitored with 2 leds.



Load www.didel.com/Witty.html to click on all our documentation.

General specs

Size	80 x 32 x 32 mm
Processor	AtMega 328P-AU
Motor and wheels	Vigor Bo-30 1:96 with Didel 32mm weels
Motor drivers	CS 7721
IR module	CHQ0038
Programming connectors	ISP 6 holes 1.27mm pich Gaia female 5 pins 1.27mm
Extension connectors	Gy521 2.54mm pitch A2A3 + - on 1.27mm holes
Switches	2x push button 1x SPD on/off switch
Voltage and current	3.0V/40mA-5.5V/80mA (both motors free running) 3.7V/200mA (both motors blocked)

Power control

See www.didel.com/Bahoma.pdf for details.

Check if the Bahoma magnets are clean before inserting. It may happen that the contact resistor is high. Interaction with the push button selecting a demo is good. But if the motor starts, the voltage drop is too high and the micro restarts.

With the 110 mAh Lipo, one can expect for more than 1 hour of action.



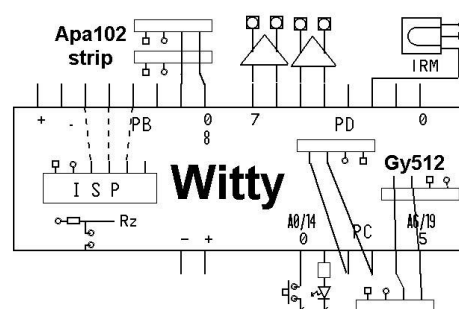
Power connection and pins

The Gaia programming adapter calls for a 340 driver, easy to install. At power-up with the driver, there is a 5 second delay for establishing the communication.

Witty microcontroller is an AtMega 328, initialized with the Duemilanove loader.

AVR328 pins

Pin	Port		Pin	Port	
0	PD0	Rx	11	PB3	(s2 prog)
1	PD1	Tx	12	PB4	(s1 prog)
2	PD2	IRmodule	13	PB5	(Tell prog)
3	PD3	–	14	PC0	Pous active low
4	PD4	bRecG	15	PC1	Led active high
5	PD5	bAvG	16	PC2	(Ana pin4)
6	PD6	bAvD	17	PC3	(Ana pin3)
7	PD7	bRecD	18	PC4	I2C SCL pin3
8	PB0		19	PC5	I2C SDA pin4
9	PB1				
10	PB2				



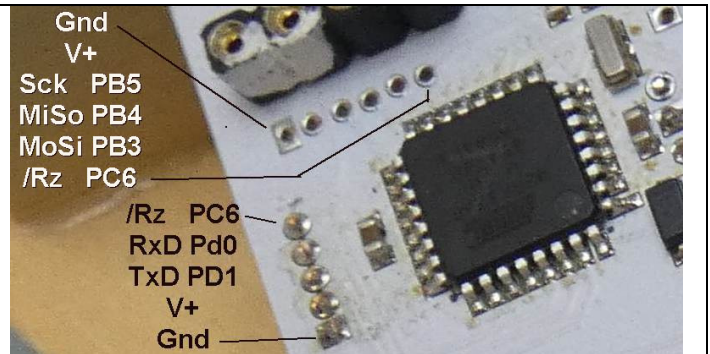
	Pin assignment

Reprogramming the AtMega328

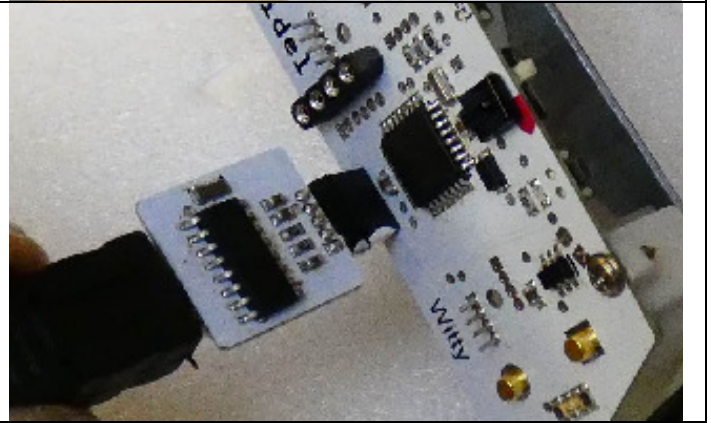
All Didel cards using an Atmel processor have a set of five 0.6mm holes, 1.25mm pitch, close to the processor.

Downloading programs

Downloading programs from the IDE environment uses a 5-pin connector compatible with the Gaia USB to 328 adapter.



Gaia always have male pins and microcontroller cards being programmed unfrequently do not need a connector; bending pins in holes is reliable.



Definition file beginning (see WittySoft.pdf)

```
// Witty.h C only
#define bLed 1 // PORTC
#define LedOn bitSet (PORTC,bLed)
#define LedOff bitClear (PORTC,bLed)
#define LedToggle (PINC^=(1<<bLed))
#define bPous 0 // actif à zero
#define PousOn (!(PINC&(1<<bPous)))

void SetupWitty () {
  DDRC = 0b000010 ; //Led out
  PORTC = 0b000001; // pullup on bPous
}
```

```
// Witty.h Arduino
#define Led 15
#define LedOn digitalWrite (Led,HIGH)
#define LedOff digitalWrite (Led,LOW)
#define Button 14
#define PushOn !(digitalRead (Button))

void SetupWitty{
  pinMode (Led,OUTPUT);
  pinMode (Button,INPUT_PULLUP);
}
```

Note: You may not like macros as we do. But we use them only at the lowest level, to give a better name to the function given to a pin. Note it is not recommended to use the low level digitalWrite, pinMode, etc inside the program; it destroys legibility and portability. See WittySoft.pdf and WittyLib for a more efficient way of programming the Witty.

Pythie

Replacing the Gaia module by the Pythie allows to connect an Oled on the Tx/Rx pins and use TerOled for displaying the program status in a more convenient way than on the screen terminal (data stays in place).

See www.didel.com/Pythie.pdf for more details.

