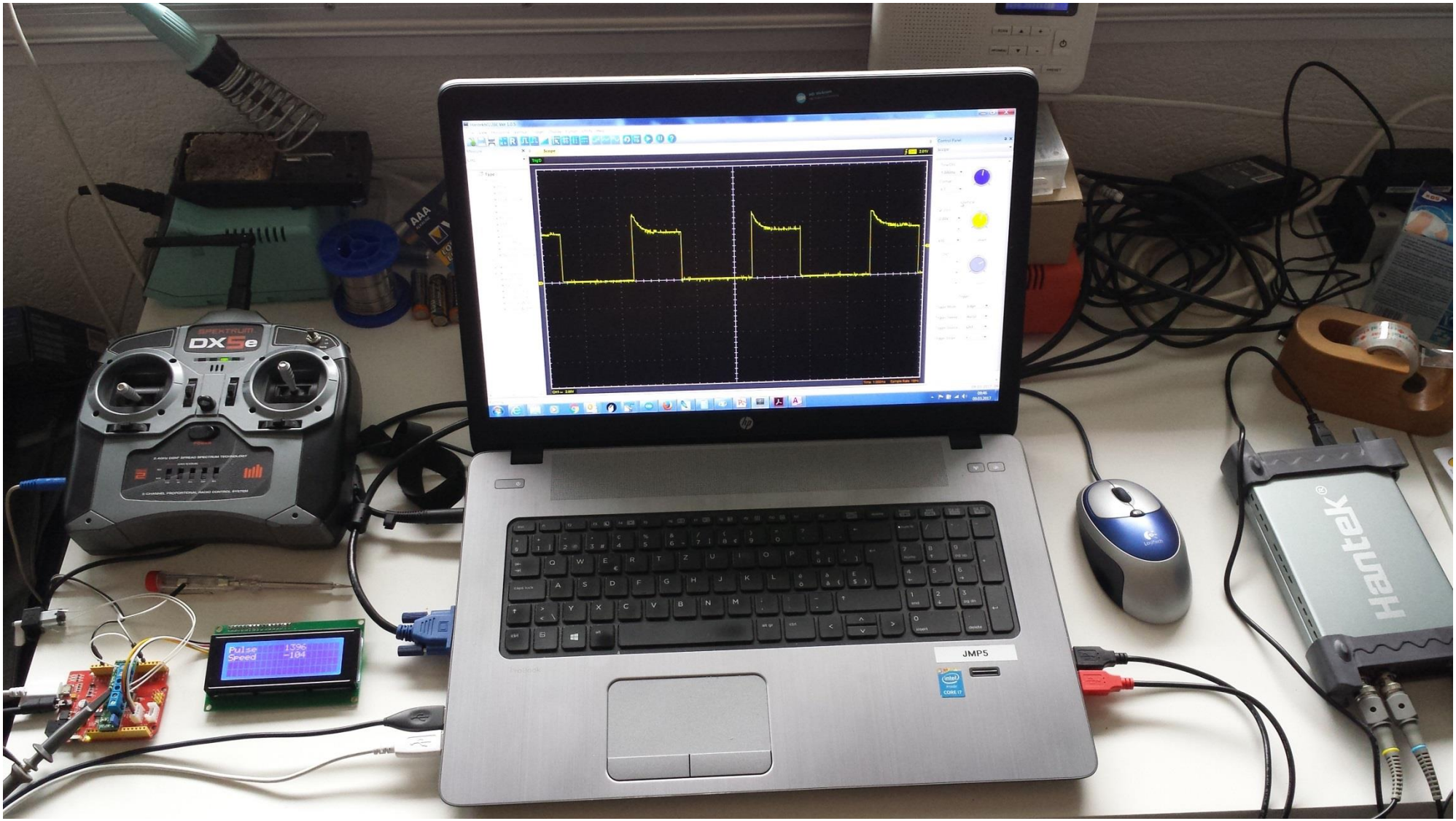
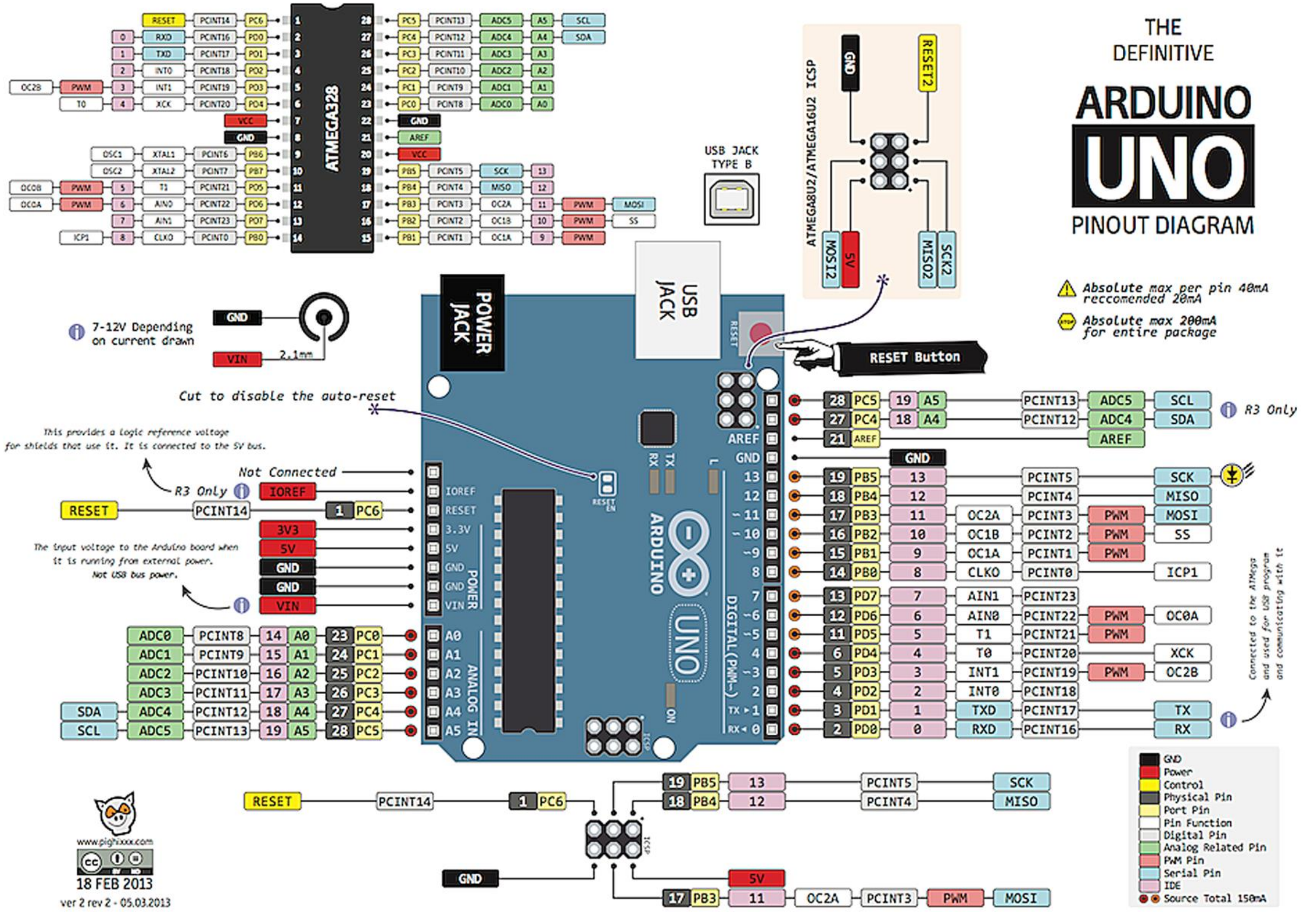
DC motors, DRV8835 et Arduino



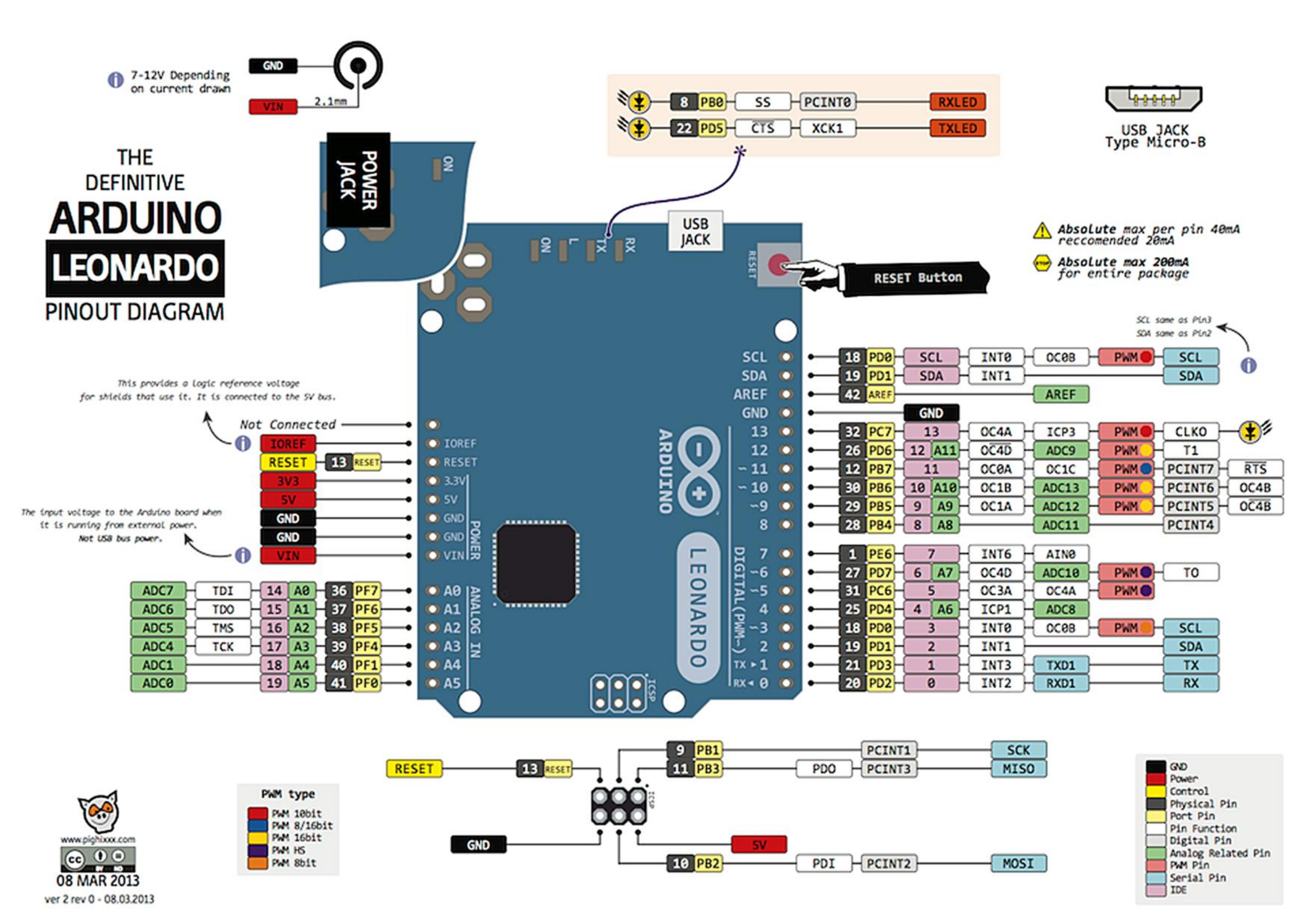
Jean-Marc Paratte
Mars 2017

Index

Arduino UNO	ATmega328P
Arduino Leonardo	ATmega32U4
Timer0	8-bit Timer/Counter
Timer1	16-bit Timer/Counter
PWM	Pulse Width Modulation
PFM	Pulse Frequency Modulation
DRV8835	Pololu Dual Motor Driver Shield for Arduino
jm_Scheduler	Scheduler Library
jm_CPPM	Combined Pulse Position Modulation Library
jm_LiquidCrystal_I2C	Revised LiquidCrystal_I2C Library
Wire	Revised Arduino Wire Library
Hantek6022BE	Low Cost USB Oscilloscope
10A ESC Brushed	Speed Controller For RC Car And Boat Without Brake


THE DEFINITIVE
ARDUINO UNO
PINOUT DIAGRAM
ATMEGA328
ATMEGA8U2/ATMEGA16U2 ICSP
USB JACK TYPE B
POWER JACK
USB JACK
RESET Button
7-12V Depending on current drawn
Cut to disable the auto-reset
This provides a logic reference voltage for shields that use it. It is connected to the 5V bus.
Not Connected
R3 Only
The input voltage to the Arduino board when it is running from external power. Not USB bus power.
Absolute max per pin 40mA reccomended 20mA
Absolute max 200mA for entire package
Connected to the ATMega and used for USB program and communicating with it
www.pighixxx.com
18 FEB 2013
ver 2 rev 2 - 05.03.2013
GND
Power
Control
Physical Pin
Port Pin
Pin Function
Digital Pin
Analog Related Pin
PWM Pin
Serial Pin
IDE
Source Total 150mA

THE DEFINITIVE ARDUINO LEONARDO PINOUT DIAGRAM



08 MAR 2013

ver 2 rev 0 - 08.03.2013

Figure 14-1. 8-bit Timer/Counter Block Diagram

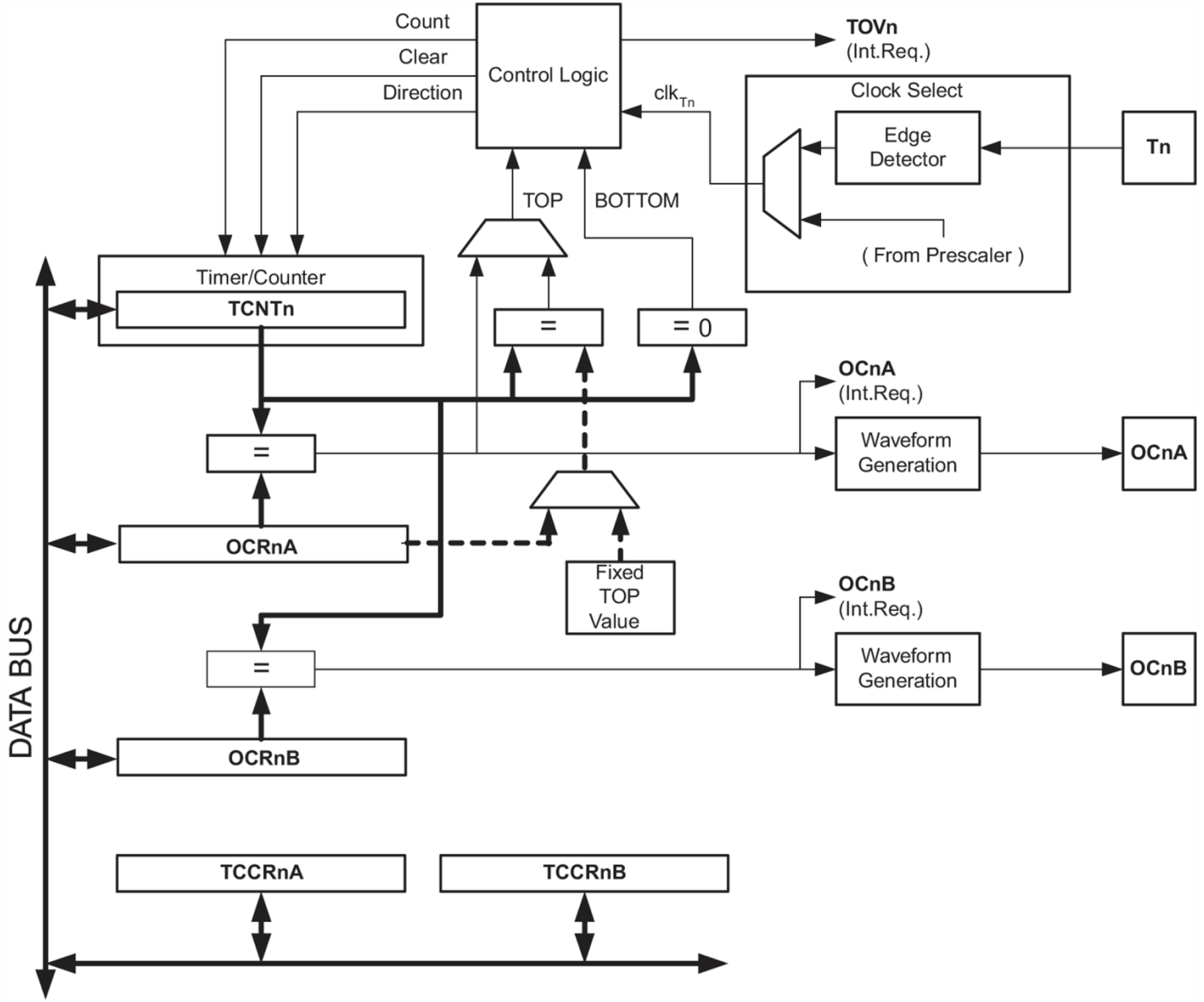
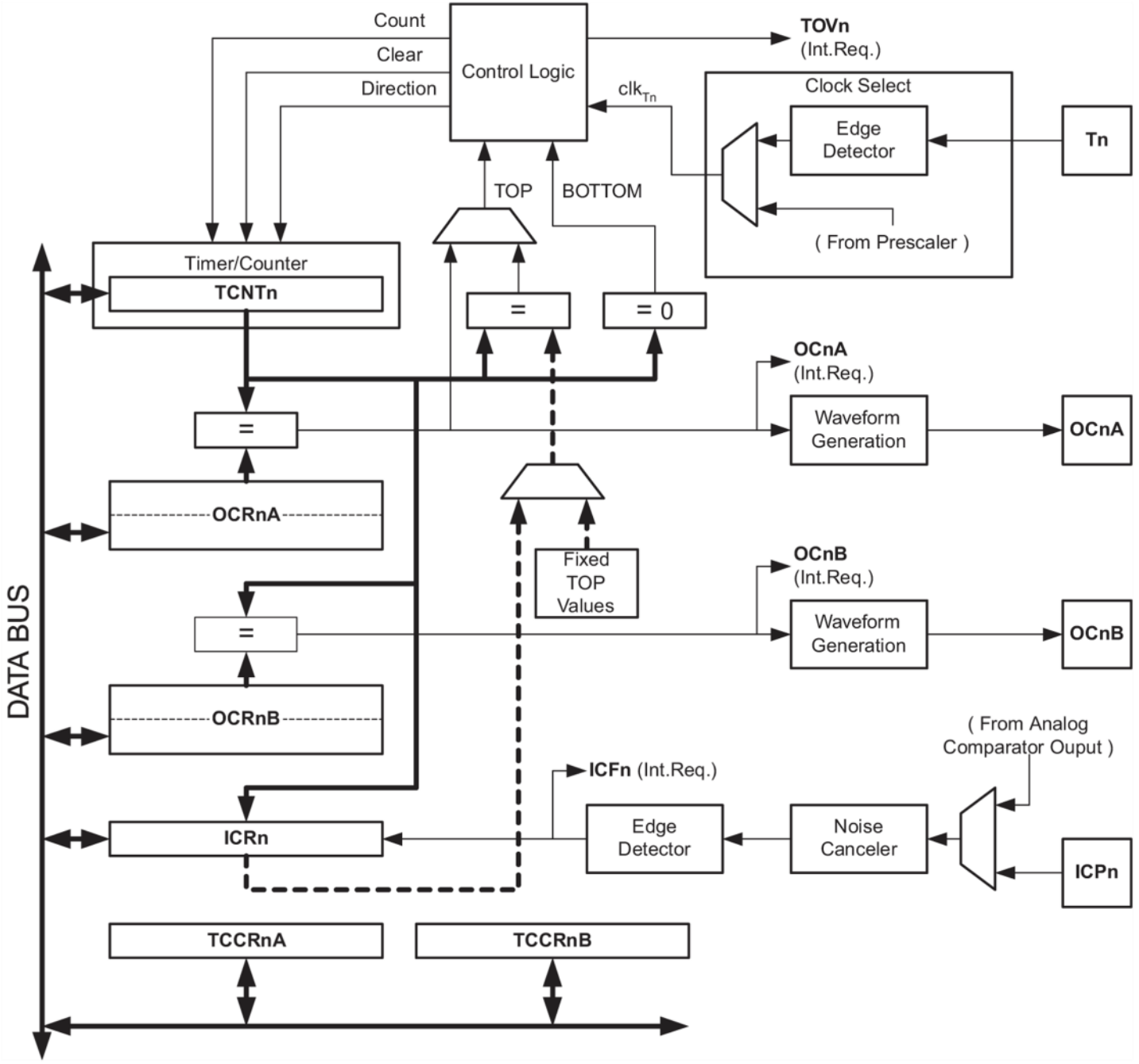


Figure 15-1. 16-bit Timer/Counter Block Diagram[(1)]



PWM - analogWrite()

https://www.arduino.cc/en/Reference/AnalogWrite

- **Description**
- Writes an analog value (PWM wave) to a pin. Can be used to light a LED at varying brightnesses or drive a motor at various speeds. After a call to **analogWrite()**, the pin will generate a steady square wave of the specified duty cycle until the next call to **analogWrite()** (or a call to **digitalRead()** or **digitalWrite()** on the same pin). The frequency of the PWM signal on most pins is approximately 490 Hz. On the Uno and similar boards, pins 5 and 6 have a frequency of approximately 980 Hz. Pins 3 and 11 on the Leonardo also run at 980 Hz.
- On most Arduino boards (those with the ATmega168 or ATmega328), this function works on pins 3, 5, 6, 9, 10, and 11. On the Arduino Mega, it works on pins 2 - 13 and 44 - 46. Older Arduino boards with an ATmega8 only support analogWrite() on pins 9, 10, and 11.
- The Arduino Due supports analogWrite() on pins 2 through 13, plus pins DAC0 and DAC1. Unlike the PWM pins, DAC0 and DAC1 are Digital to Analog converters, and act as true analog outputs.
- You do not need to call pinMode() to set the pin as an output before calling analogWrite().
- The *analogWrite* function has nothing to do with the analog pins or the *analogRead* function.

PWM - analogWrite()

https://www.arduino.cc/en/Reference/AnalogWrite

- **Syntax**
- analogWrite(pin, value)
- **Parameters**
- pin: the pin to write to.
- value: the duty cycle: between 0 (always off) and 255 (always on).
- **Returns**
- nothing
- **Notes and Known Issues**
- The PWM outputs generated on pins 5 and 6 will have higher-than-expected duty cycles. This is because of interactions with the millis() and delay() functions, which share the same internal timer used to generate those PWM outputs. This will be noticed mostly on low duty-cycle settings (e.g 0 - 10) and may result in a value of 0 not fully turning off the output on pins 5 and 6.

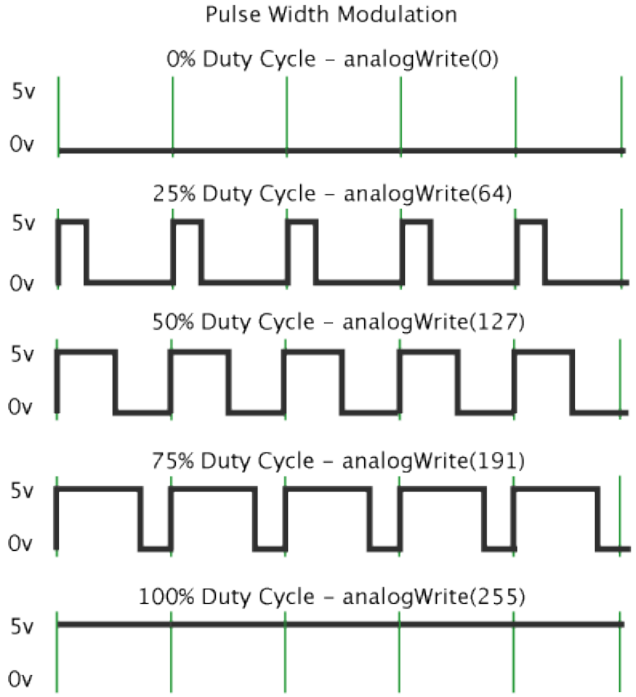
PWM - Pulse Width Modulation

https://www.arduino.cc/en/Tutorial/PWM

- **PWM**
- The Fading example demonstrates the use of analog output (PWM) to fade an LED. It is available in the File->Sketchbook->Examples->Analog menu of the Arduino software.
- Pulse Width Modulation, or PWM, is a technique for getting analog results with digital means. Digital control is used to create a square wave, a signal switched between on and off. This on-off pattern can simulate voltages in between full on (5 Volts) and off (0 Volts) by changing the portion of the time the signal spends on versus the time that the signal spends off. The duration of "on time" is called the pulse width. To get varying analog values, you change, or modulate, that pulse width. If you repeat this on-off pattern fast enough with an LED for example, the result is as if the signal is a steady voltage between 0 and 5v controlling the brightness of the LED.
- In the graphic below, the green lines represent a regular time period. This duration or period is the inverse of the PWM frequency. In other words, with Arduino's PWM frequency at about 500Hz, the green lines would measure 2 milliseconds each. A call to analogWrite() is on a scale of 0 - 255, such that analogWrite(255) requests a 100% duty cycle (always on), and analogWrite(127) is a 50% duty cycle (on half the time) for example.

PWM - Pulse Width Modulation

https://www.arduino.cc/en/Tutorial/PWM

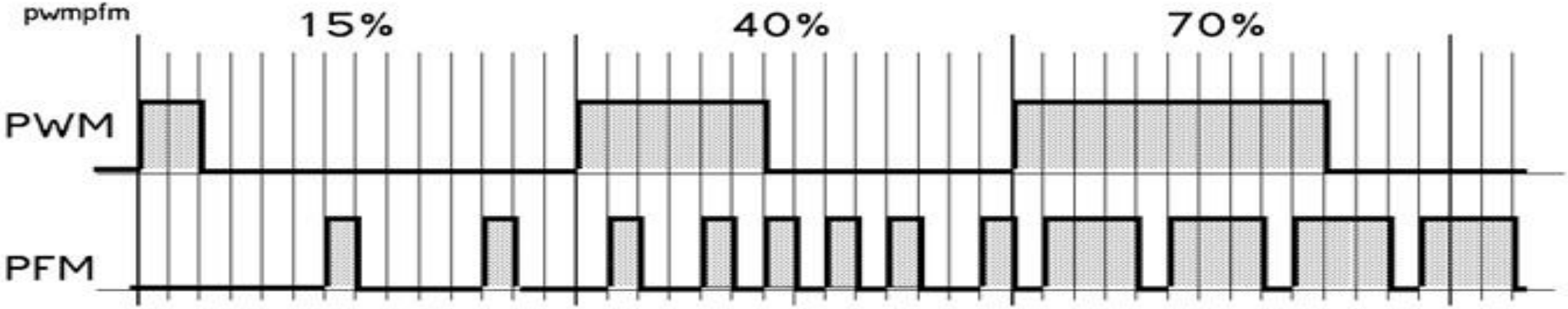


- Once you get this example running, grab your arduino and shake it back and forth. What you are doing here is essentially mapping time across the space. To our eyes, the movement blurs each LED blink into a line. As the LED fades in and out, those little lines will grow and shrink in length. Now you are seeing the pulse width.
- *Written by Timothy Hirzel*

PFM - Pulse Frequency Modulation

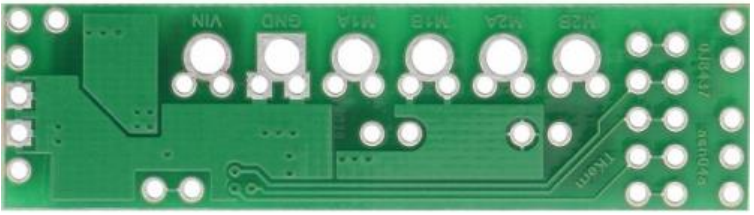
https://www.zigobot.ch/fr/fiches-pratiques/44-programmation-calm/110-programmer-le-robot-bimo-en-calm.html

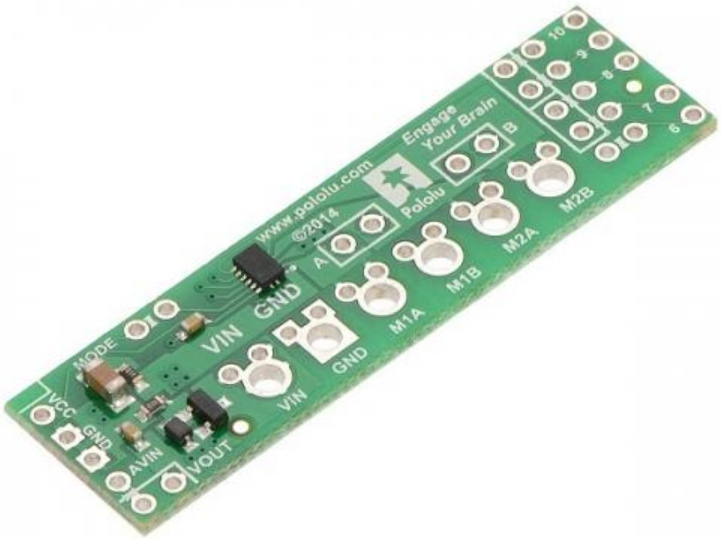
- Pour agir sur la vitesse du moteur (et sur l'intensité d'une LED) le principe est d'envoyer des impulsions, à une fréquence suffisante pour que l'inertie du moteur ou la persistance lumineuse donne l'effet de continuité.
- Avec le **PWM** (Pulse Width Modulation), la période est constante et le pourcentage actif varie.
- Avec le **PFM** (Pulse Frequency Modulation), la durée d'excitation du moteur est constante et l'espacement des impulsions varie.
- Le PFM est très intéressant avec des moteur de basse qualité ou qui tournent trop vite. On choisit une durée d'impulsion suffisamment longue pour que le moteur démarre (1-10ms). On attend ensuite en fonction de la vitesse de rotation moyenne voulue. A faible vitesse, les a-coups sont visibles, mais on obtient des faibles vitesses que le PWM ne permet pas.

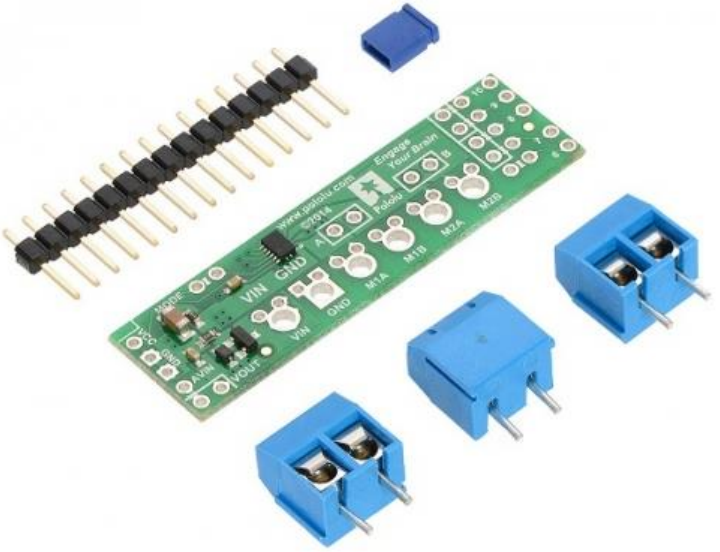


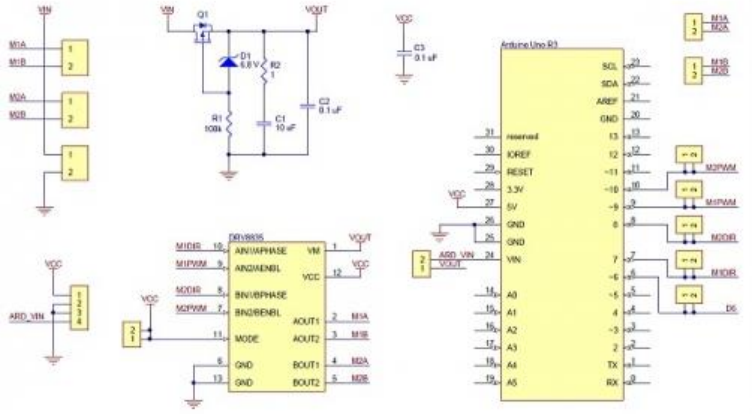
DRV8835 Dual Motor Driver Shield for Arduino

http://www.hobbytronics.co.uk/drv8835-dual-motor-driver-shield





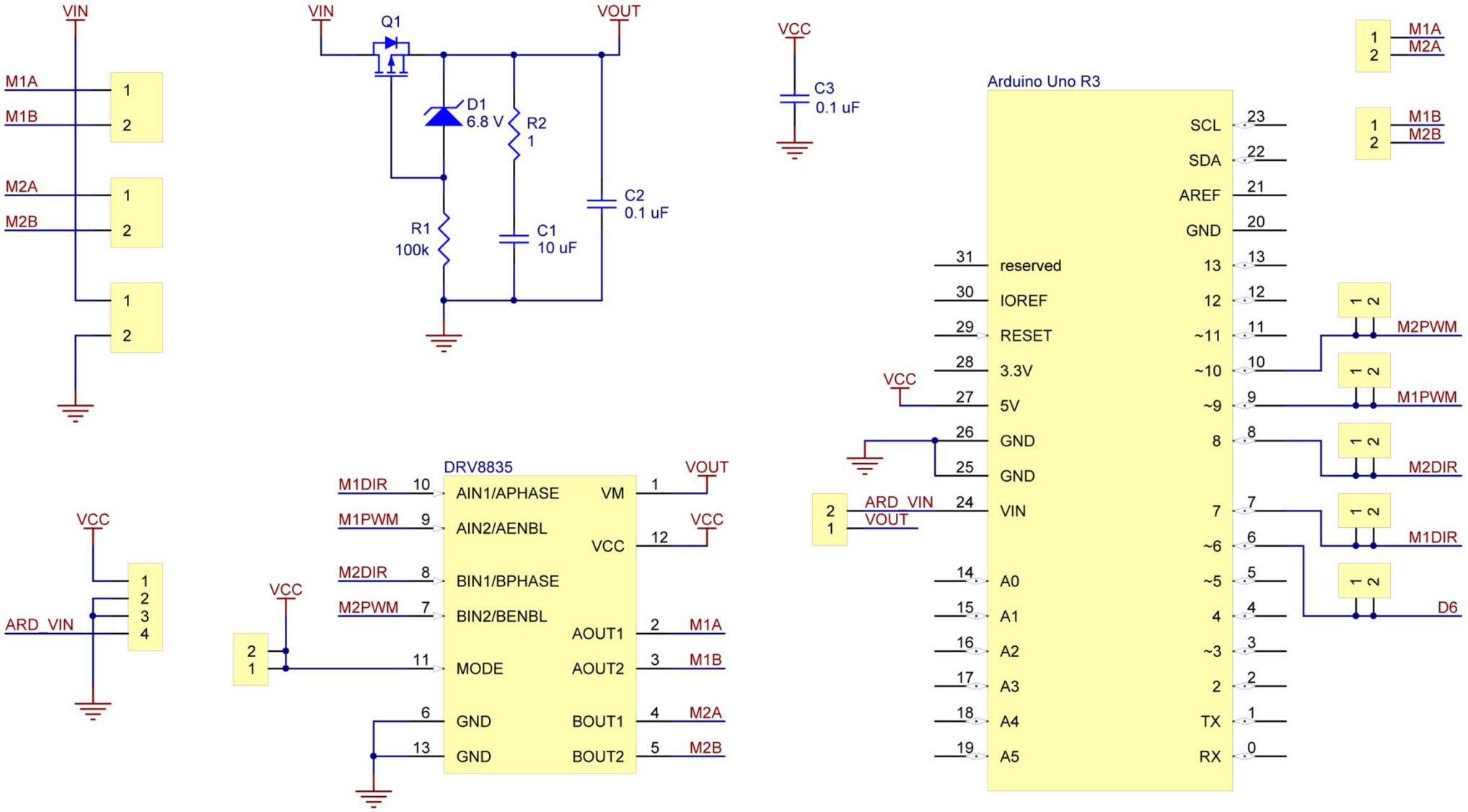




DRV8835 Dual Motor Driver Shield for Arduino

www.pololu.com

http://www.hobbytronics.co.uk/drv8835-dual-motor-driver-shield



DRV8835 Dual Motor Driver Shield for Arduino

http://www.hobbytronics.co.uk/drv8835-dual-motor-driver-shield

By default, the board operates in PHASE/ENABLE mode, in which a PWM signal applied to the ENABLE pin determines motor speed and the digital state of the PHASE pin determines direction of motor rotation. Arduino pins 9 and 7 are used to control the speed and direction, respectively, of motor 1, and pins 10 and 8 control the speed and direction of motor 2. The table below shows how the inputs affect the outputs in this mode:

Drive/brake operation in default PHASE/ENABLE mode				
xPHASE	xENABLE	MxA	MxB	operating mode
0	PWM	PWM	L	forward/brake at speed *PWM %*
1	PWM	L	PWM	reverse/brake at speed *PWM %*
X	0	L	L	brake low (outputs shorted to ground)

PHASE/ENABLE mode should be suitable for most applications.

jm_Scheduler, jm_CPPM, jm_LiquidCrystal_I2C, Wire

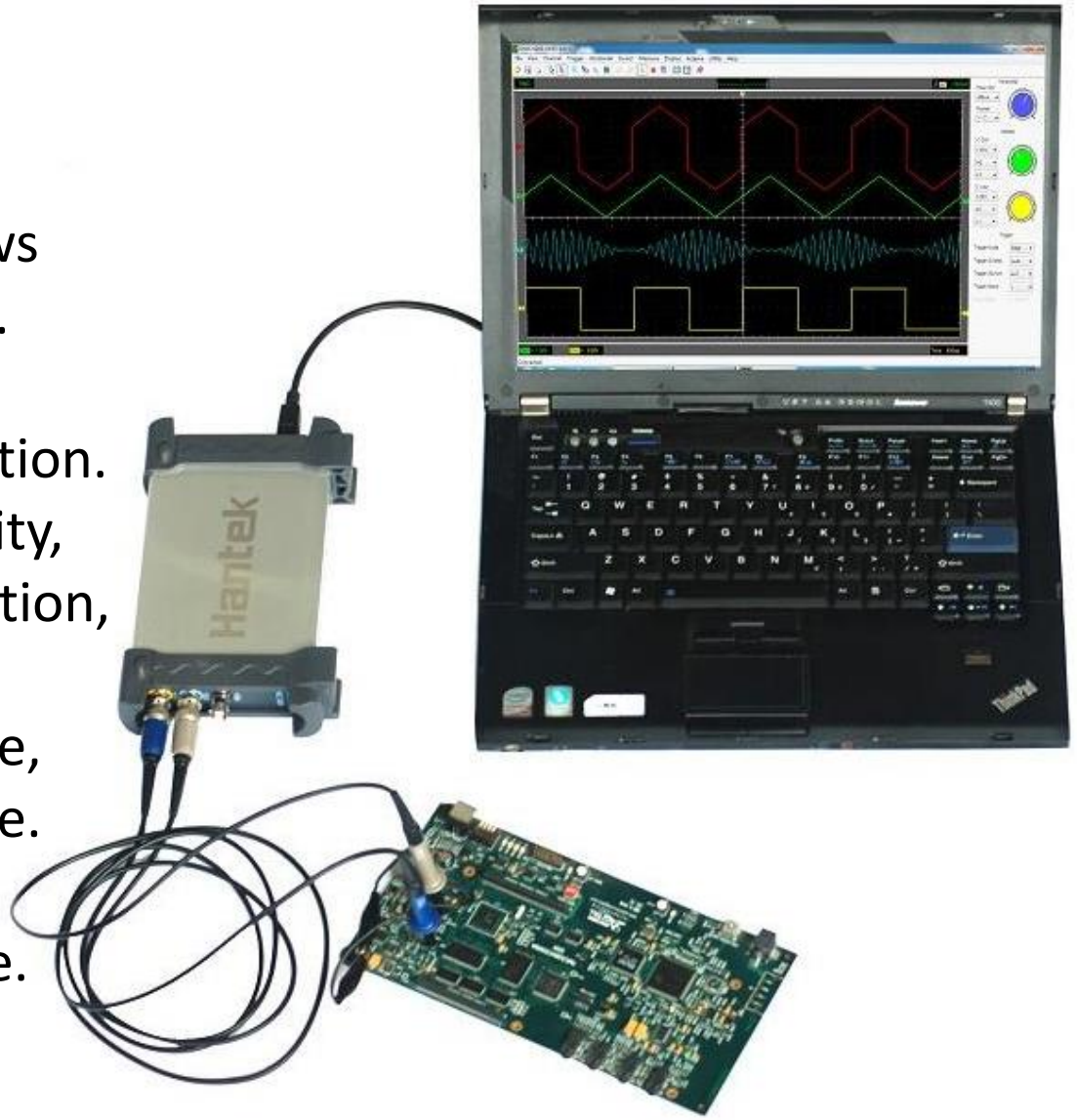
Switch to TextPad

Hantek6022BE

http://www.banggood.com/Hantek-6022BE-PCBased-USB-Digital-Storag-Oscilloscope-2-Channels-p-925842.html
http://www.hantek.com/en/ProductDetail_2_31.html

Features

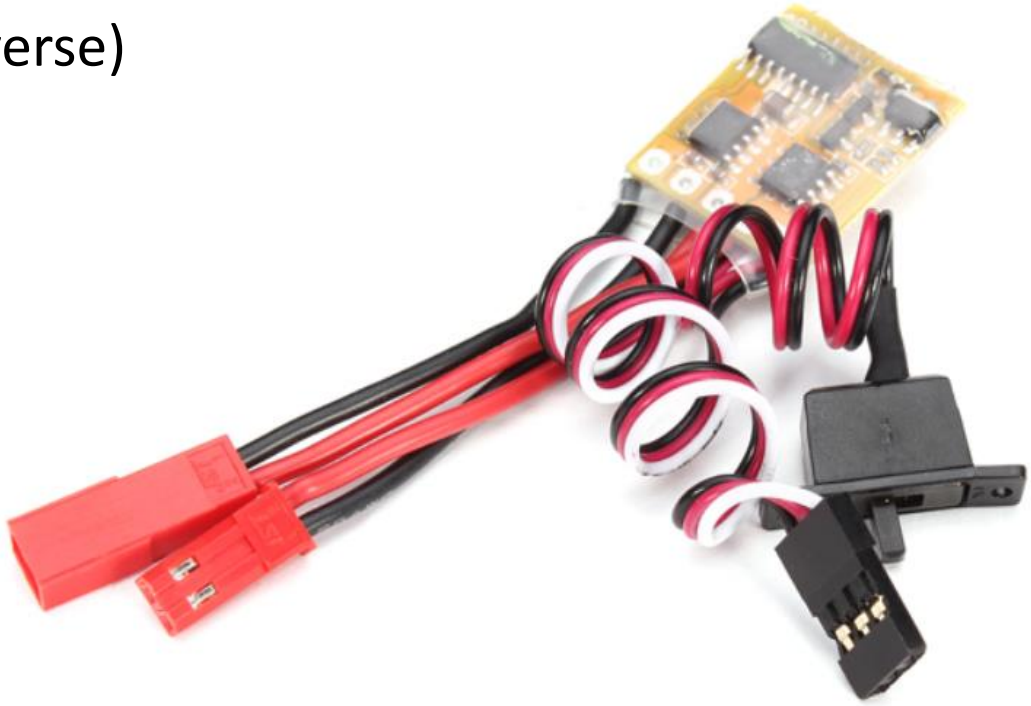
- High performance, 48MS/s real-time sampling, 20MHz bandwidth.
- Operating System: Windows 7, Windows NT, Windows 2000, Windows XP, VISTA.
- 23 measurement functions, PASS/FAIL Check, be suitable for technical application.
- Waveform average, persistence, intensity, invert, addition, subtraction, multiplication, division, X-Y plot.
- Save waveform in the following: text file, jpg/bmp graphic file, MS excel/word file.
- FFT
- Labview\VB\VC Second Design instance.



10A ESC Brushed Speed Controller For RC Car And Boat Without Brake

http://www.banggood.com/10A-ESC-Brushed-Speed-Controller-For-RC-Car-And-Boat-Without-Brake-p-966363.html

Description:



It`s Brushed bustophedon ESC.(forward,reverse)
Weight :9g
Dimensions(L*W*H):30*21*6.0mm
Current(A):10A
BEC :5V 1A
PWM:8K
Input signal :PPM
Driver frequency:2KHz
Li-Po :2S
Ni-Mh/Ni-cd :4-7cell
Constant current 10A Max 15A< 30s Pulsed 30A< 5s
With brake
For 1/16 1/18 1/24 car and boat