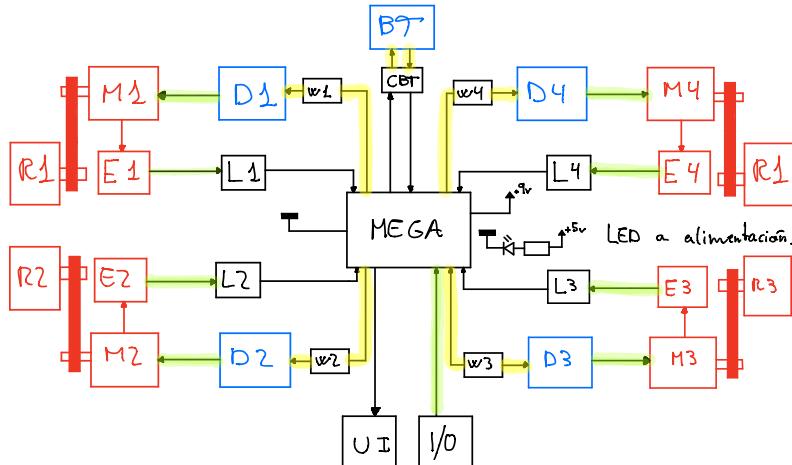


- Arduino mega2560 x 1
- Motor driver x 4 (Vhn2sp30)
- Motor x 4
- Bluetooth HC-05 x 1
- SNL554XKD SOIC x 4
- Displays 7 segmentos x 4
- LEDs x 26
- Resistencias...
- Condensadores...
- LED OK in stop
- LED RGB adafruit → Fast LED x2
- Potenciómetro
- Pulsadores x 5
- Display 7 segmentos ánodo común x4
- Microswitches de 4



● Motor
● Header
■ Mecánica

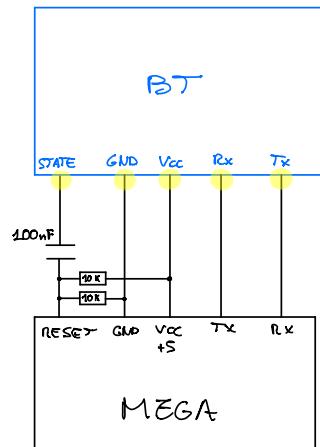
Mx Motor
 Rx Rueda
 Ex Encoder
 ■ Placa fabricada
 Dx Driver
 BT Bluetooth

■ Placa a diseñar
 Lx Lector del Encoder
 CBT Circuito Bluetooth
 UI Interfaz de usuario
 MEGA 2560
 I/O Comunicaciones.
 Wx LEDs en antiparalelo.

• Añadir FTDI para comunicaciones

CBT: este módulo nos permite tanto comunicarnos con la MEGA por bluetooth como subirla código de forma inalámbrica.

- Gastamos el Serial 0
- Perdemos la posibilidad de subir código por cable si el módulo está conectado.

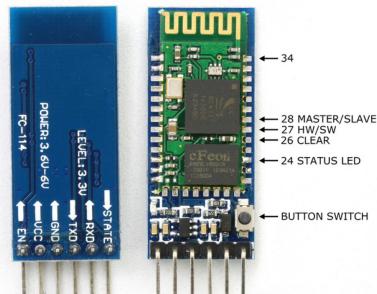


Si conectamos la MEGA al NUC podremos renunciar a este módulo.

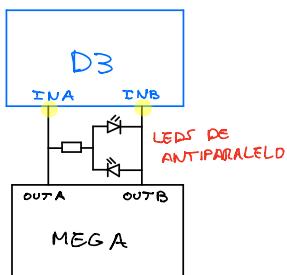
Comandos AT

```

AT+ORGL
AT+ROLE=0
AT+POLAR=1,0
AT+UART=115200,0,0
AT+INIT
  
```



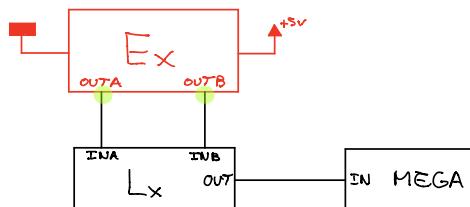
Wx: este módulo se añade a la comunicación entre la MEGA y los drivers vhn2sp30 para indicar en qué sentido gira cada rueda.



I/O: utilizaremos los puertos seriales del 1 al 3 para comunicarnos con y desde la MEGA al resto de placas. En principio solo haría falta un serial para comunicarnos con el NUC. Podría ser que solo estuviera el Serial 0 directo al NUC.

Lx: lee los encoders de los motores y los transforma a diferencia de potencial como si fuera una entrada analógica de 0 a 255 leída desde la MEGA.

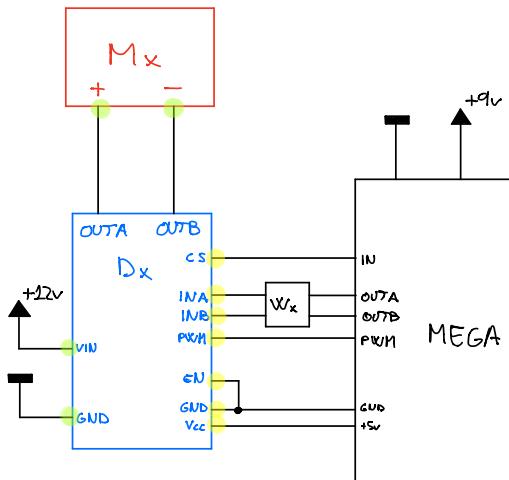
Los encoders estarán conectados al eje trasero del motor.



Consumimos 4 entradas de la MEGA:

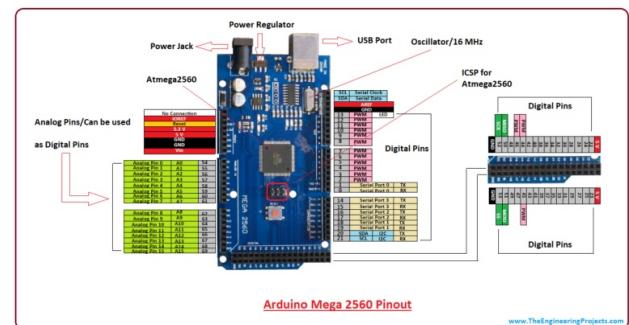
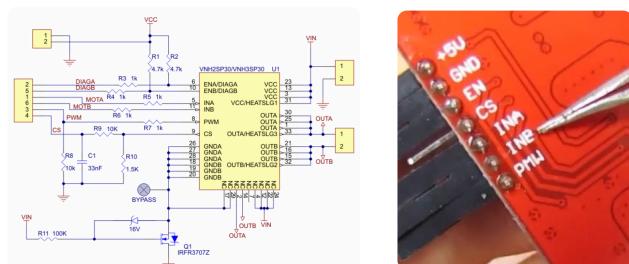
- L1
- L2
- L3
- L4

Dx: conexión entre los drivers, los motores y la MEGA.



Consumimos 16 entradas en la MEGA

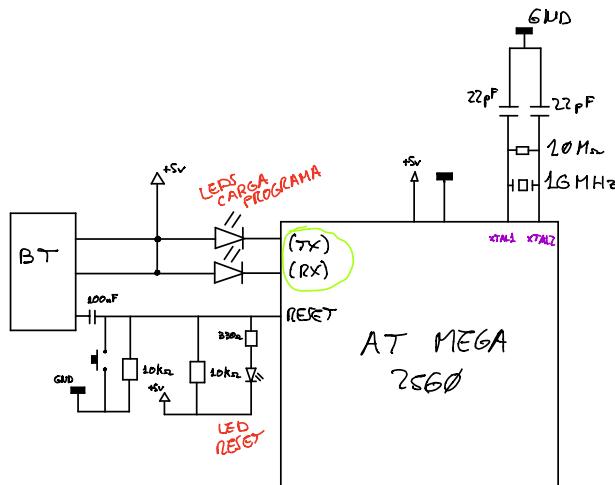
- OUTA 1
- OUTB 1
- PWM 1
- IN 1
- OUTA 2
- OUTB 2
- PWM 2
- IN 2
- OUTA 3
- OUTB 3
- PWM 3
- IN 3
- OUTA 4
- OUTB 4
- PWM 4



La salida CS del módulo nos proporciona información sobre el amperaje consumido por los motores.

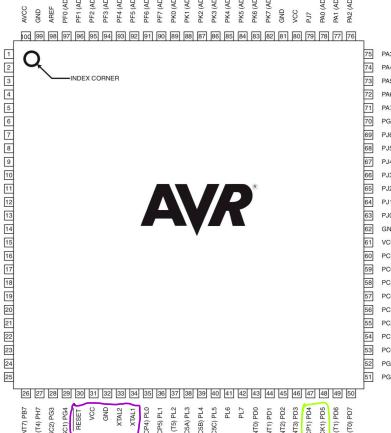
• IN 4

MEGA 2560

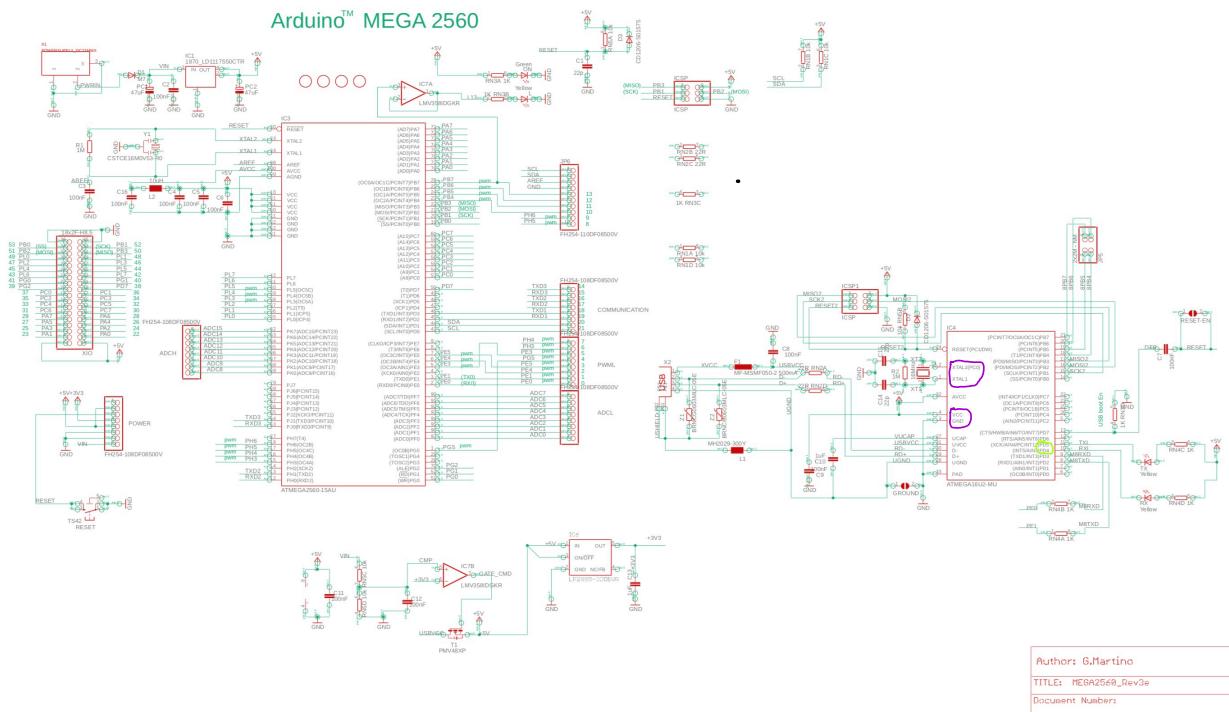


1. Pin Configurations

Figure 1-1. TQFP-pinout ATmega640/1280/2560



AVR

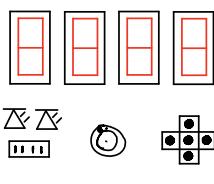


Author: G.Martino
TITLE: MEGA2560_Rev3e
Document Number:
Date: 24/08/18 12:19 | Sheet:

- Oscilador de cuarzo
- Botón de reset
- Regulador de potencia?
- Diodos de protección a la entrada?

U1: consumimos 29 pines.

- LED OK in setup
- LED RGB adafruit → Fast LED [x2]
- Potenciómetro
- Pulsadores x5
- Display 7 segmentos ánodo común [x4]
- Microswitches de 4



Servo para mostrar la dirección??

- Oled → I2C

Conector para cargar el Bootloader → ICSP



ORDERING INFORMATION

SN54LSXXJ Ceramic
SN74LSXXN Plastic
SN74LSXXD SOIC

Enable	Anodo Común						
	A	B	C	D	E	F	G
0	0	0	0	0	0	0	1
1	1	0	0	1	1	1	1
2	0	0	1	0	0	1	0
3	0	0	0	0	1	1	0
4	1	0	0	1	1	0	0
5	0	1	0	0	1	0	0
6	0	1	0	0	0	0	0
7	0	0	0	1	1	1	1
8	0	0	0	0	0	0	0
9	0	0	0	0	1	0	0

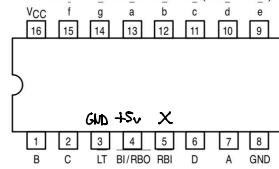


NUMERICAL DESIGNATIONS – RESULTANT DISPLAYS

TRUTH TABLE

DECIMAL FUNCTION	INPUTS							OUTPUTS							NOTE
	L	T	R	B	C	D	A	B1R1B0	a	b	c	d	e	f	
0	H	H	L	L	L	L	H	L	L	L	L	L	L	H	A
1	H	X	L	L	L	H	H	H	L	L	L	L	L	H	A
2	H	X	L	L	H	H	H	H	L	L	L	L	L	H	A
3	H	X	L	L	H	H	H	H	L	L	L	L	L	H	L
4	H	X	L	L	L	H	H	H	H	L	L	H	H	L	L
5	H	X	L	H	L	H	H	H	H	H	H	H	H	L	L
6	H	X	L	H	L	H	H	H	H	H	H	H	H	L	L
7	H	X	L	H	H	H	H	H	L	L	L	H	H	H	H
8	H	X	H	L	L	H	H	H	L	L	L	L	L	H	L
9	H	X	H	L	L	H	H	H	L	L	L	L	L	H	L
10	H	X	H	H	L	H	H	H	H	H	H	H	H	H	L
11	H	X	H	H	H	H	H	H	L	L	H	H	H	H	L
12	H	X	H	H	L	L	H	H	L	H	H	H	H	L	L
13	H	X	H	H	L	H	H	H	L	H	H	H	H	L	L
14	H	X	H	H	H	H	H	H	L	H	H	H	H	L	L
15	H	X	H	H	H	H	H	H	H	H	H	H	H	H	H
16	X	X	X	X	X	X	X	X	L	H	H	H	H	H	B
17	H	L	L	L	L	L	L	L	H	H	H	H	H	H	C
18	L	X	X	X	X	X	X	X	H	H	H	H	H	H	D

CONNECTION DIAGRAM DIP (TOP VIEW)



GND +5v X

- Sensor de temperatura

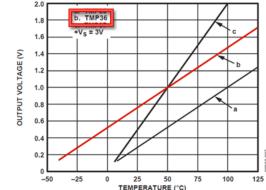
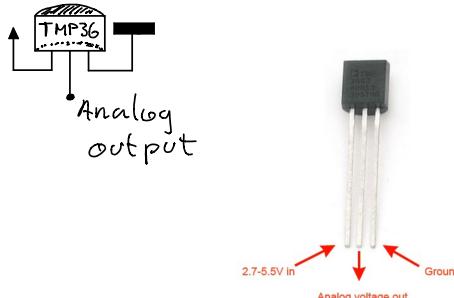


Figure 6. Output Voltage vs. Temperature

To convert the voltage to temperature, simply use the basic formula:

$$\text{Temp in } ^\circ\text{C} = [(V_{out} \text{ in mV}) - 500] / 10$$

- WiFi ESP

esp8266 / ESP-01



1 - GND	10	MODO CARGA	MODO EJECUCIÓN
2 - GPIO 2	~	+	+
3 - GPIO 0	~	+	+

Board Generic Module ESP8266
Flash Mode DIO
Flash Size 512K (64K SPIFFS)
Debug Port Disabled
Debug Level None
lwIP Variant v2 Lower Memory
Reset Method ck
Crystal Frequency 26 MHz
VTables Flash
Flash Frequency 40 MHz
CPU Frequency 80 MHz
Built-in LED 2
Upload Speed 115200
Erase Flash Only Sketch
Programmer AVRISP mkII → Arduino: Arduino as ISP

