DOUBLE VARIATEUR DE VITESSE POUR BATEAUX

JFS59@2019

Réalisation d'un variateur double pour bateau et d'un ensemble de 8 voies supplémentaires.

Matériel :

Arduino pro mini , afficheur barre de led, réseau de résistance, ULN 2005, Bornes de sorties pour circuit imprimé, convertisseur DC/DC 12V → 5V , IBT-4 (pont en H).

Principe :

L'Arduino décode les signaux RC PPM (4 canaux) et génère les pwm nécessaires à la commande de deux IBT-4 (pont en H) à partir des canaux 1 et 2.

Si fonctionnement sur un seul canal des deux moteurs :

Le canal 2 est affecté au gouvernail et permet suivant le mode de fonctionnement de fonctionner en moteur différentiel.

Le canal 3 est décodé pour activer le canal 4 en sortie ou en demande de bruiteur,

Le canal 4 est décodé pour commander 8 sorties ULN 2005 selon le principe d'extension de voie. (8 voies sur un canal). En fonction du canal 3 il permet de commander 8 bruits différents, (corne, cloche, diesel)

L'ensemble est configurable par liaison Bluetooth a l'aide d'un programme spécifique développé pour Android. Un mode simulateur est disponible.

Sont configurables : (En plus du mode de fonctionnement)

Limite haute télécommande Canal 1 : 2000 par défaut.

Limite basse télécommande Canal 1 : 1000 par défaut.

Neutre Canal 1 : 1500 par défaut.

Limite haute télécommande Canal 2 : 2000 par défaut.

Limite basse télécommande Canal 1 : 1000 par défaut.

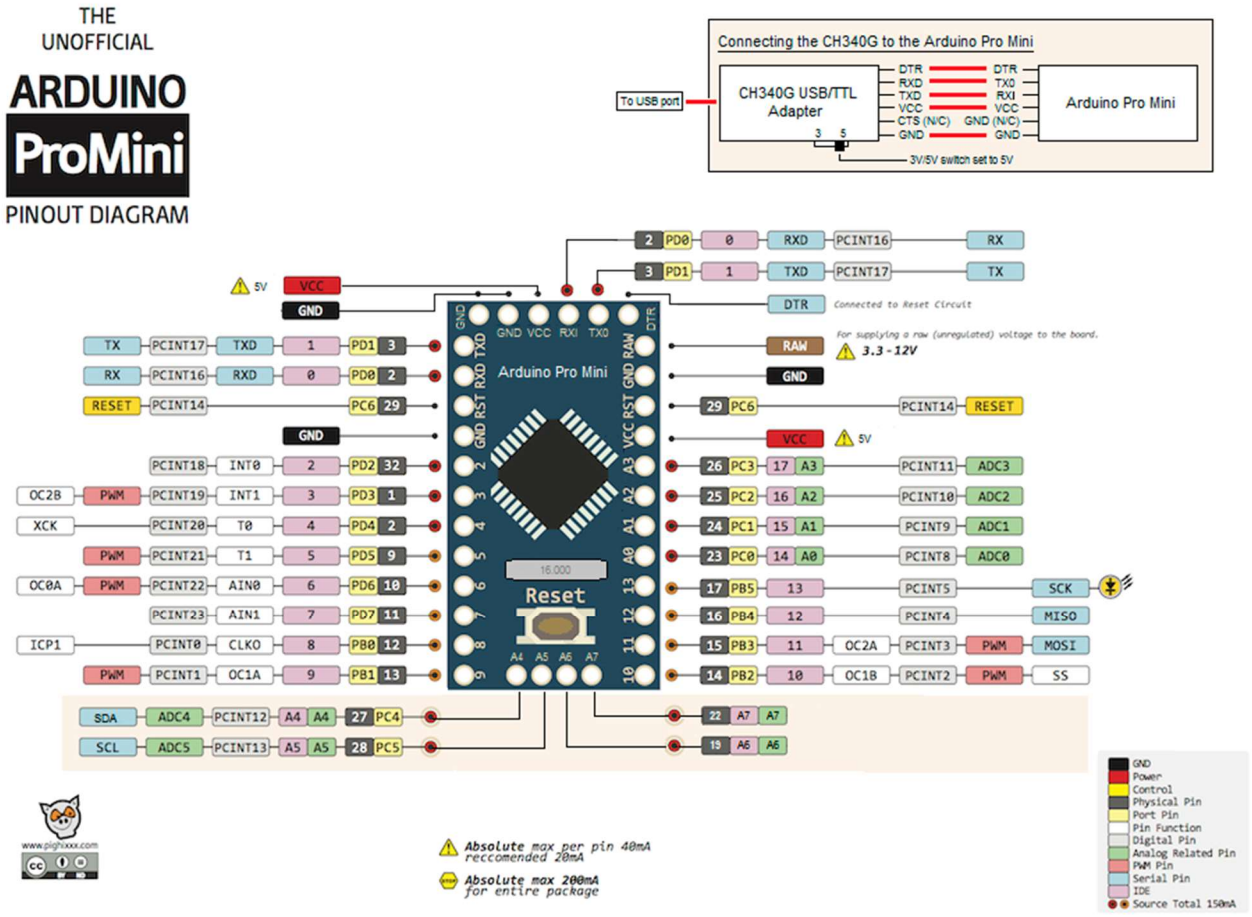
Neutre Canal 2 : 1500 par défaut.

Largeur zone neutre : +/- 50 par défaut.

Pourcentage AV Pourcentage AR Canal 1 : 100% par défaut.

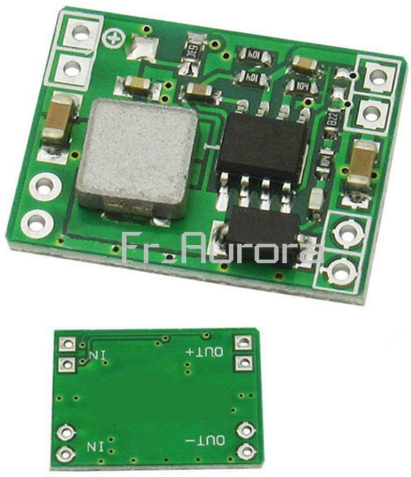
Pourcentage AV Pourcentage AR Canal 2 : 100% par défaut.

Mode de fonctionnement des 8 sorties : Monostables Bistables : 4 monostables, 4 bistables par défaut.



Remarque : l'utilisation de A6 et A7 en numérique implique des résistances de tirage au +. A6 et A7 étant purement analogique il faut utiliser une astuce de programmation.

Le bootloader de la carte devra être changé pour empêcher le clignotement au reste de la carte ! sinon il faut faire attention au rôle de la sortie S7 qui va s'allumer au reset !!



Convertisseur DC/DC 12 V/ 5 V

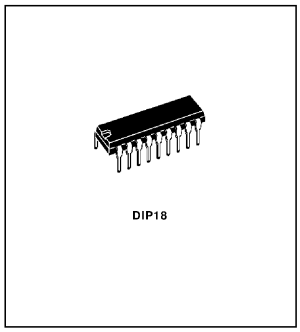
Utilisé pour alimenter l'Arduino et les diodes de visualisation. Une sortie 5 V est prévue pour alimenter un montage externe.

Sur la première version montage du convertisseur composant en dessous !

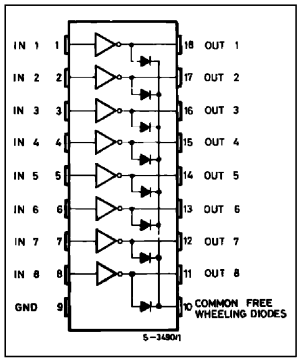
Version 3 le convertisseur est positionné composants au-dessus.

ULN 2005 interface de puissance

- EIGHT DARLINGTONS WITH COMMON EMITTERS
- OUTPUT CURRENT TO 500 mA
- OUTPUT VOLTAGE TO 50 V
- INTEGRAL SUPPRESSION DIODES
- VERSIONS FOR ALL POPULAR LOGIC FAMILIES
- OUTPUT CAN BE PARALLELED
- INPUTS PINNED OPPOSITE OUTPUTS TO SIMPLIFY BOARD LAYOUT



PIN CONNECTION (top view)



DESCRIPTION

The ULN2801A-ULN2805A each contain eight darlington transistors with common emitters and integral suppression diodes for inductive loads. Each darlington features a peak load current rating of 600mA (500mA continuous) and can withstand at least 50V in the off state. Outputs may be paralleled for higher current capability.

Five versions are available to simplify interfacing to standard logic families : the ULN2801A is designed for general purpose applications with a current limit resistor ; the ULN2802A has a 10.5kΩ input resistor and zener for 14-25V PMOS ; the ULN2803A has a 2.7kΩ input resistor for 5V TTL and CMOS ; the ULN2804A has a 10.5kΩ input resistor for 6-15V CMOS and the ULN2805A is designed to sink a minimum of 350mA for standard and Schottky TTL where higher output current is required.

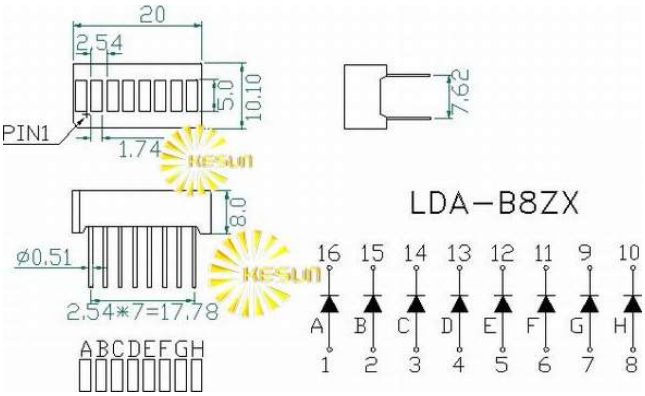
All types are supplied in a 18-lead plastic DIP with a copper lead frame and feature the convenient input-opposite-output pinout to simplify board layout.

Permet de relayer les sorties Arduino afin de disposer d'un courant de sortie plus important et d'une tension différente, peuvent se mettre en // pour augmenter le courant de sortie il suffit de les souder patte pour patte l'un au-dessus de l'autre,

Diode anti retour de protection communes a toutes les sorties.

Les sorties OUT sont mises à la masse quand un niveau haut (1) est présent à l'entrée correspondante.

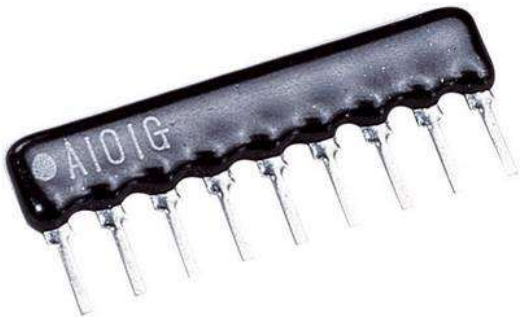
Bargraphe led



Visualise l'état des canaux et donc des sorties de puissance. Les cathodes sont reliées à la masse à travers un réseau des 8 résistances dont le point commun est à la masse.

Interrupteur DIL

Permettent de choisir le mode de fonctionnement de 0 à 15. Depuis la version Bluetooth le mode est configurable par le programme Android.



Réseau de 8 résistances

Choisir la valeur entre 470 Ohm et 1 K Ohm,

Borniers a vis clipsables,

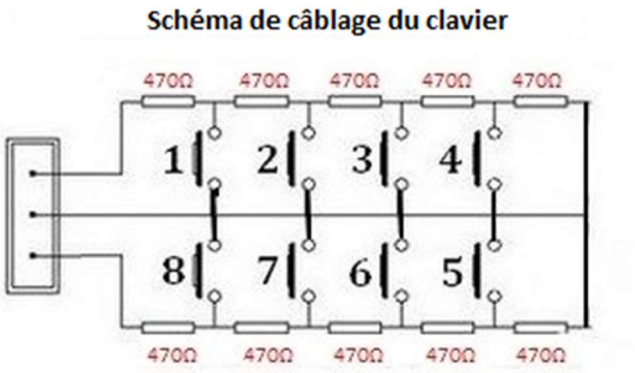
Par 2 ou 3 il en faut 12 : 2 X6 ou 3 X4

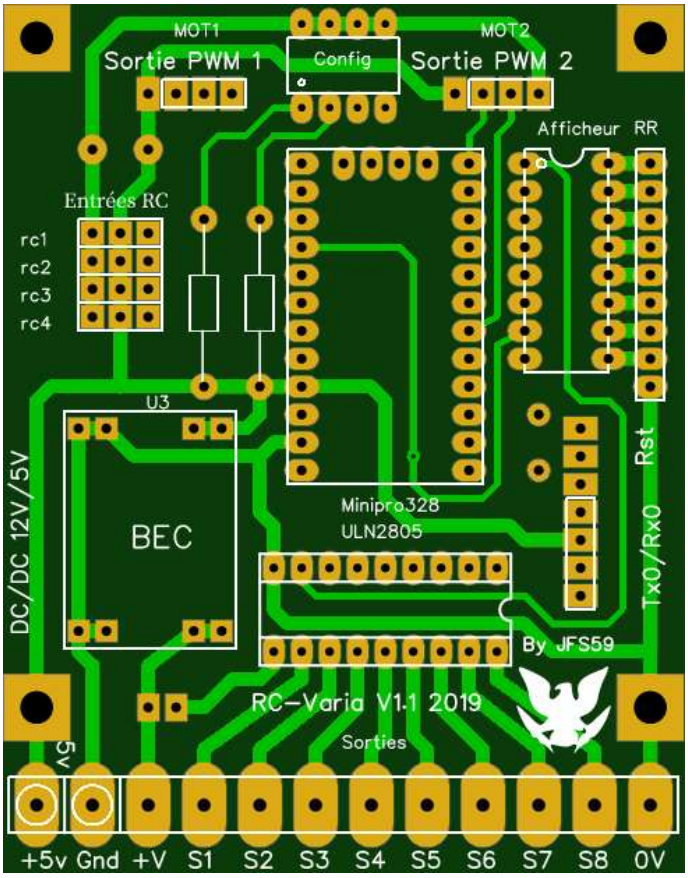
IBT4 Pont en H

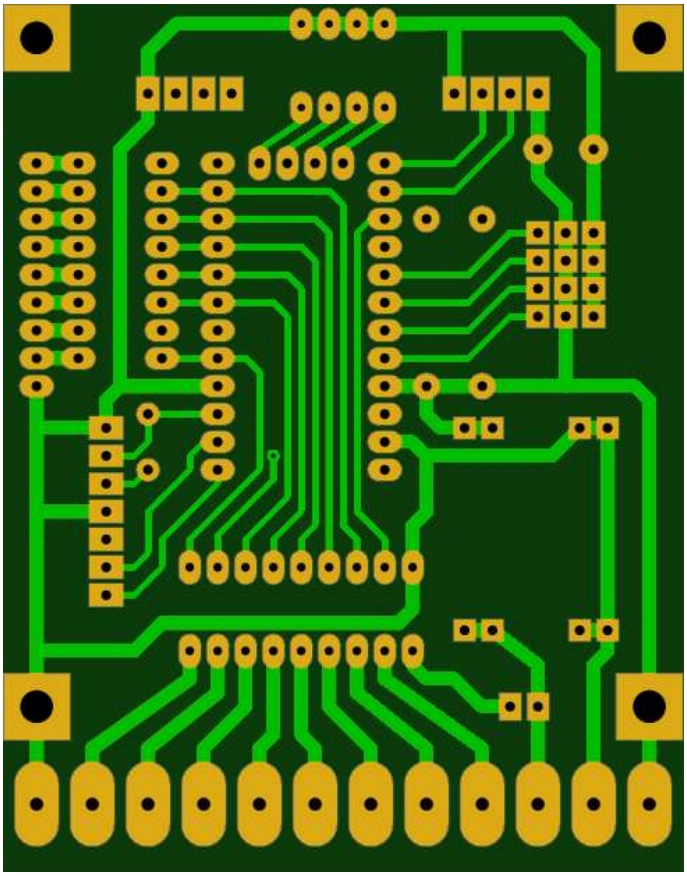
Caractéristiques :

- Module driver Pont H à courant élevé (50A) MOSFET
- Isolation galvanique du signal PWM.
- Rotation avant et arrière du moteur, deux entrées PWM au maximum 200kHz
- Alimentation 3.3V à 15V.


Clavier émetteur déconnectable







Structure du programme Arduino :

Variables de configurations :

```
struct __attribute__ ((packed)) RC_Config {
unsigned long Flag;
  byte STRUCT_VERSION;
  byte ModeFonctionnement;
  unsigned int  Voie_1_Neutre;
  unsigned int  Voie_1_Arriere;
  unsigned int  Voie_1_Avant;
  unsigned int  Voie_2_Neutre;
  unsigned int  Voie_2_Arriere;
  unsigned int  Voie_2_Avant;
  byte Plage_Neutre;
  byte PourcentAvantV1;
  byte PourcentArriereV1;
  byte PourcentAvantV2;
  byte PourcentArriereV2;
  byte ModeSorties[8];
  int  Seuil_Bouton[8];
};
```

Configuration par défaut :

```
void DefinirConfigDefaut() {
  Configuration.Flag = 123456789;
  Configuration.STRUCT_VERSION = 1;
  Configuration.ModeFonctionnement = 11 ;
  Configuration.Voie_1_Neutre = 1500;
  Configuration.Voie_1_Avant = 2000;
  Configuration.Voie_1_Arriere = 1000;
  Configuration.Voie_2_Neutre = 1500;
  Configuration.Voie_2_Avant = 2000;
  Configuration.Voie_2_Arriere = 1000;
  Configuration.Plage_Neutre = 50;
  Configuration.PourcentAvantV1 = 100;
  Configuration.PourcentArriereV1 = 100;
  Configuration.PourcentAvantV2 = 100;
  Configuration.PourcentArriereV2 = 100;
  for ( int n = 0; n < 8; n ++) {
    Configuration.ModeSorties[n] = 0 ;
  }
  for ( int n = 0; n < 8; n ++) {
    Configuration.Seuil_Bouton[n] = 1500 ;
  }
}
```

Gestion des entrées RC par interruption :

```
static   byte rcOld;        // Prev. states of inputs
volatile unsigned long rcRises[4]; // times of prev. rising edges
volatile unsigned long rcTimes[4]; // recent pulse lengths
volatile unsigned int  rcChange=0; // Change-counter
// Be sure to call setup_rcTiming() from setup()
void setup_rcTiming() {
  rcOld = 0;
  pinMode(A0, INPUT_PULLUP);  // pin 14, A0, PC0, for pin-change interrupt
  pinMode(A1, INPUT_PULLUP);  // pin 15, A1, PC1, for pin-change interrupt
  pinMode(A2, INPUT_PULLUP);
  pinMode(A3, INPUT_PULLUP);
  PCMSK1 |= 0x0F;       // Four-bit mask for four channels
  PCIFR  |= 0x02;       // clear pin-change interrupts if any
  PCICR  |= 0x02;       // enable pin-change interrupts
}
// Define the service routine for PCI vector 1
```

```
ISR(PCINT1_vect) {
  byte rcNew = PINC & 15;   // masquage 4 bits, A0-A3
  byte changes = rcNew^rcOld;   // verif changement bit
  byte channel = 0;
  unsigned long now = micros(); // micros() ok
  while (changes) {
    if ((changes & 1)) {  // Did current channel change?
      if ((rcNew & (1<<channel))) { // Check rising edge
        rcRises[channel] = now;     // Is rising edge
      } else {              // Is falling edge
        rcTimes[channel] = now-rcRises[channel];
      }
    }
    changes >>= 1;      // shift out the done bit
    ++channel;
    ++rcChange;
  }
  rcOld = rcNew;        // Save new state
}
```

Chronologie du câblage :

Souder les deux résistances cms 1K et vérifier en testant sur le circuit.

Souder la platine alimentation BEC.

Souder une ligne de bornier (possibilité de souder : +V, gnd, +5v uniquement)

Brancher une tension >6 V sur Gnd, +V et vérifier la présence du + 5 V.

Souder le connecteur Rx/Tx Bluetooth et le condensateur c3 de 100 nf. (pour permettre le flash de la carte Arduino)

Souder les pattes extrêmes du réseau et vérifier le sens et les valeurs. (Souder le reste des pattes)

Souder le support d'afficheur. Placer l'afficheur et vérifier chaque segment en injectant +5 v sur les pastilles correspondantes au niveau de la carte Arduino.

Souder la plaque Arduino MiniPro.

Charger un sketch (programme) dans l'Arduino (Eventuellement le bootloader no led)

Mettre sous tension entre Gnd et +V (tension > 6V). Le programme démarre est on doit observer a chaque reset un défilement « bargraphe » sur l'afficheur.

Souder le support ULN.

Placer un ULN et vérifier le bon fonctionnement de chaque sortie (S1 a S8). (Terminer les soudures borniers si nécessaire)

Souder l'interrupteur DIL éventuellement sur support.

Souder les connecteurs de sortie PWM.