



VARIATEUR DE VITESSE POUR BATEAUX



Disponible sur <https://github.com/jfs59/Mini-variateur>

Merci de bien lire le descriptif.

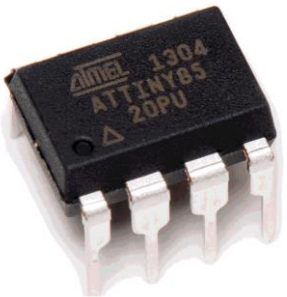
Toujours charger la dernière version de ce document sur le GitHub !

Mise en garde : Bien que ne comportant pas de difficultés majeures la réalisation n'est pas facile ! Si vraiment grand débutant faites-vous aider.

EN COURS D'ELABORATION !

JFS59 @ 2022-2023

Table des matières

Réalisation d'un variateur simple avec sortie 5V.....	3
Préambule :	3
Principe :	3
Schémas	3
ATTINY 85 5V / 8MHz DIP8.....	3
	
.....	3
Circuit imprimé	4
Composants et sous-ensembles.	4
Convertisseur DC/DC 12 V/ 5 V (BEC)	4
Borniers a vis clipsables,	5
ATTiny 85 en boitier DIL 8	5
Chronologie de câblage :	5
Compilation et Flash du Bootloader.....	6
Repérage des connecteurs de la mini pro :	6
Structure et extraits du programme AtTiny	6
Variables de configurations :	6
Configuration par défaut :	6
Gestion des entrées RC par interruption :	6
Interconnexions :	7
Chronogrammes :	8

Réalisation d'un variateur simple avec sortie 5V.

Préambule :

Principe :

L'AtTiny décode le signal RC PPM et génère les pwm nécessaires à la commande du pont en H.

L'ensemble est configurable l'aide d'un programme spécifique développé pour Windows.

Sont configurables

Neutre Canal : 1600 par défaut. (L'attiny a tendance à sous-estimer la valeur)

Delta avant : 500 par défaut.

Delta arrière : 500 par défaut.

Largeur zone neutre : +/- 30 par défaut.

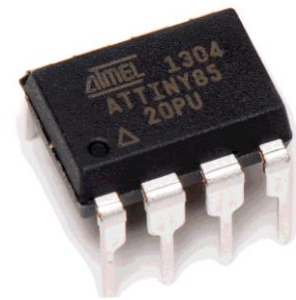
Pourcentage AV Pourcentage AR Canal 1 : 100% par défaut.

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

Taux de lissage : 5 par défaut.

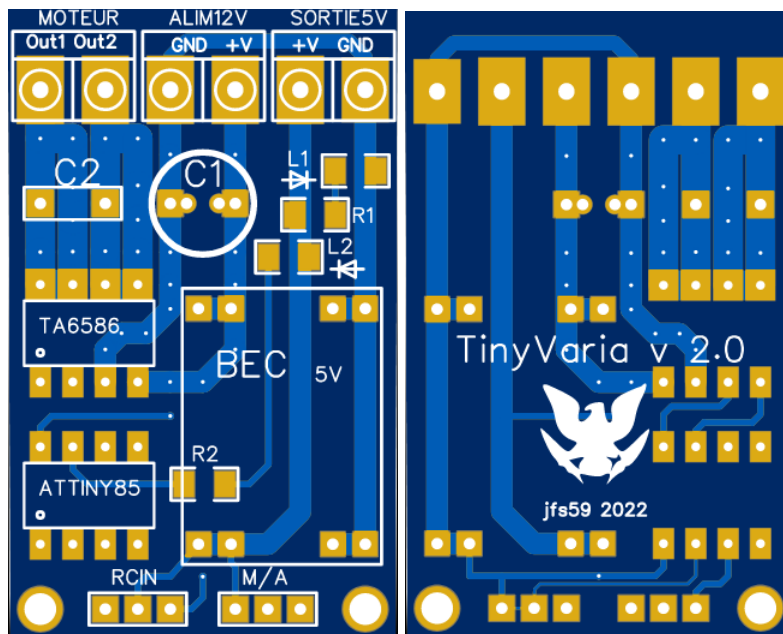
Schémas

ATTINY 85 5V / 8MHz DIP8



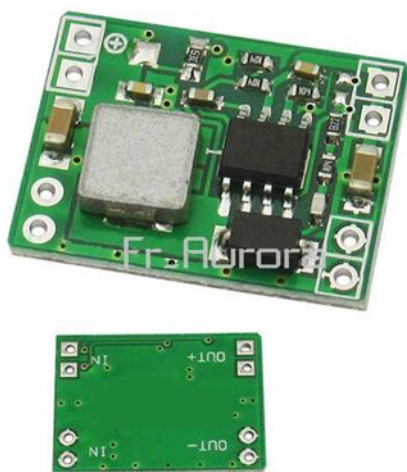
Circuit imprimé

Circuit double face, trous métallisés, vernis épargne, et sérigraphie. Dessiné et routé par mes soins. Réalisé par une entreprise. (Le résultat donne un CI professionnel) Le routage et le design sont donnés ci-dessous mais les fichiers Gerber sont et resteront ma propriété. Le circuit imprimé est disponible pour la somme de 2.5 € couvrant les frais de réalisation et transport. (Me contacter au club de Raismes.)



Composants et sous-ensembles.

Convertisseur DC/DC 12 V/ 5 V (BEC)



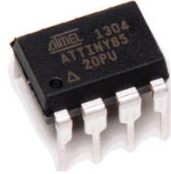
Utilisé pour alimenter l'AtTiny et les diodes de visualisation. Une sortie 5 V est prévue pour alimenter un montage externe.

Le convertisseur est positionné composants au-dessus.

Borniers a vis clipsables.



Par 2 ou 3 il en faut 6 : 2 X3 ou 3 X2



ATTiny 85 en boîtier DIL 8

(avec ou sans bootloader peu important)

Chronologie de câblage :

Souder les deux résistances cms 820 Ohm et vérifier en testant sur le circuit. (Consulter un tuto de soudage cms)

Souder les deux LED cms , Vérifier qu'elles s'allument en injectant du 5v dans le bon sens !

Souder la platine alimentation BEC.

Souder la ligne de bornier.

Brancher une tension >6 V sur Gnd, +V et vérifier la présence du + 5 V. (! la LED s'allume !)

Souder les condensateurs.

Souder le support de l'attiny.(Support nécessaire pour pouvoir le reflasher)

Charger le sketch (programme) dans l'Attiny.

Fixer l'attiny dans le support.(vérifier le sens !)

Mettre sous tension entre Gnd et +V (tension > 6V). Le programme démarre et on doit observer le clignotement de la deuxième led.

Débrancher.

Souder le TA6586. (Vérifier le sens !)

Souder les connecteurs de l'entrée RCin

Brancher un moteur et un générateur RC (ou un récepteur)

Faire varier le signal le moteur doit accélérer et changer de sens suivant la valeur du signal.

Compilation et Flash du Bootloader

Repérage des connecteurs de la mini pro :

Se référer à la procédure de flash soit avec un ISP soit avec un Arduino câblé en ISP et le programme (Arduino as ISP).

Structure et extraits du programme AtTiny

Le code complet ne sera pas publié par contre je laisse libre accès au fichier compilé et donc au fichier hexadécimal à programmer.

Je peux sous certaines conditions faire pour vous la mise à jour.

Variables de configurations :

```
static const unsigned long STRUCT_FLAG = 11223344;

struct __attribute__((packed)) RC_Config {
    unsigned int  Val_Neutre;
    unsigned int  Delta_Avant;
    unsigned int  Delta_Arriere;
    uint8_t Plage_Neutre;
    uint8_t PourcentAvant;
    uint8_t PourcentArriere;
    uint8_t Lissage;

    unsigned long Flag; // contrôle d'intégrité sauvegarde
};
```

Configuration par défaut :

```
void DefinirConfigDefaut() {

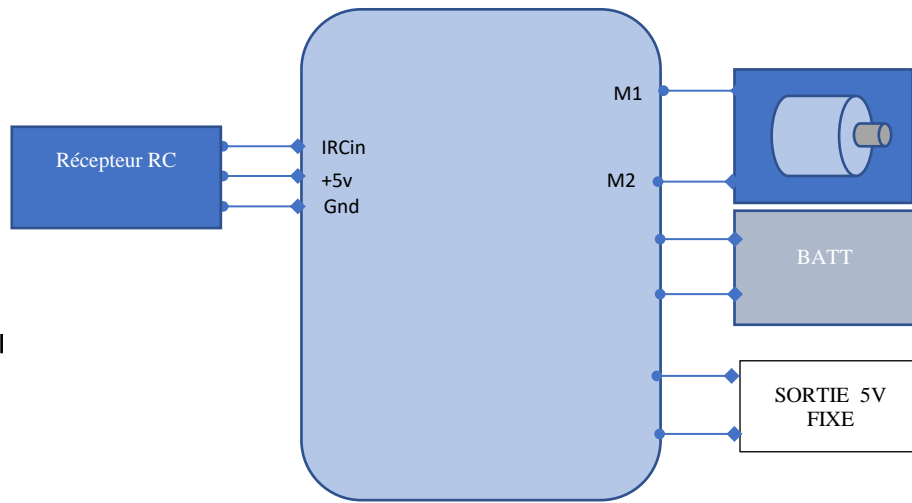
    Configuration.Val_Neutre = 1600;
    Configuration.Delta_Avant = 500;
    Configuration.Delta_Arriere = 500;
    Configuration.Plage_Neutre = 30;
    Configuration.PourcentAvant = 100;
    Configuration.PourcentArriere = 100;
    Configuration.Lissage = 5;

    Configuration.Flag = 01223344;
}
```

Gestion des entrées RC par interruption :

Interconnexions :

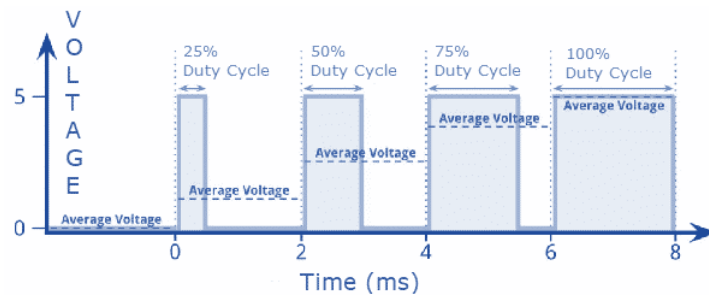
L'entrées RC alimente le récepteur en 5 V
attention de ne pas alimenter le récepteur par
une autre alimentation en même temps !



La sortie 5V peut alimenter les accessoires.

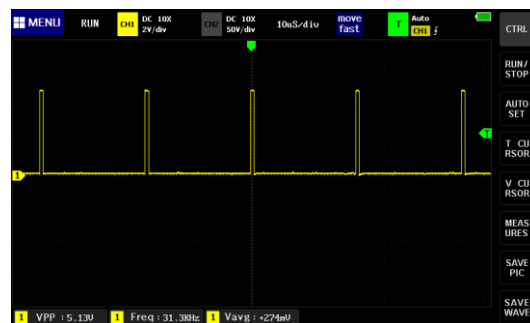
Chronogrammes :

Explication du PWM : Le signal peut varier de 0 à 100% en changeant le temps du signal à l'état haut. Cela correspond donc à alimenter avec une tension correspondant à la moyenne du temps à l'état haut par rapport à l'état bas (Duty cycle).

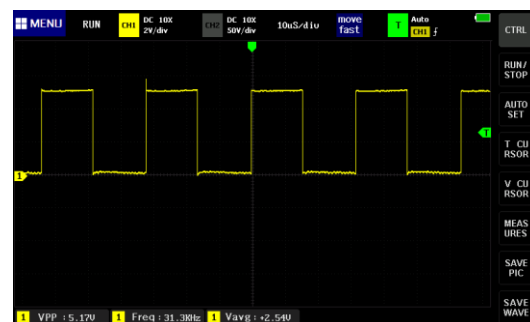


Le pont en H sera alimenté par une tension PWM déterminée par le processeur en fonction du signal RC.

Tension minimum le moteur tourne à peine.



Tension à 50% le moteur tourne à mi-vitesse.



Tension à 100% le moteur tourne à pleine vitesse.



La fréquence du PWM

