

Dimer 12 voies DMX
9 x 20A dimmer
+3 x 20A extended

Basé sur un processeur
"Arduino" MEGA 2560

Table des matières

| | |
|---|----|
| Pré requis..... | 3 |
| Utilisation des ressources : | 4 |
| Secteur | 4 |
| Entrées / sorties arduino Mega2560 | 5 |
| Etude électronique | 6 |
| Assemblage des cartes | 6 |
| Interface puissance dimmer..... | 8 |
| Broche μ P | 8 |
| Interface extended | 9 |
| Protection du système, de l'utilisateur | 10 |
| Interface DMX..... | 11 |
| Alimentation BT et détection 0V secteur..... | 12 |
| Carte façade | 14 |
| Etude programmation..... | 15 |
| Réception DMX..... | 15 |
| Traitement des données DMX reçues | 18 |
| Calculs des retards de phases | 19 |
| Méthode de mesure de la fréquence du réseau | 19 |
| Méthode de mesure des retards de phase | 19 |
| Choix de la configuration des timers pour le dimmer | 22 |
| Interruption du passage à 0 secteur | 24 |
| ANNEXES | 25 |
| Carte Alimentation | 25 |
| Carte Puissance A, B, C et D | 30 |
| Carte DMX | 34 |
| Carte Extended..... | 36 |
| Carte μ P | 37 |
| Carte Façade..... | 39 |
| Adresses des registres | 42 |

Pré requis

Etant donné le besoin en rapidité du code, ce dernier est optimisé et fait donc appel aux registres internes du processeur. Passer par les fonctions de l'IDE arduino ralenti bien trop le programme. Cette optimisation permet également de savoir exactement où on met les mains et de s'assurer qu'aucun "sketch" n'utilise une ressource déjà prise.

Les différents schémas et typons des cartes du système ne sont pas à échelle, pour cela, se référer aux fichiers joints (PROTEUS 4.1 et BMP 300dpi). les fichiers BMP sont les vues côté composants, soit la face imprimée contre le cuivre.

Les schémas de l'étude électronique peuvent différer des schémas finaux en annexes. Les schémas finaux sont fonctionnels, ils sont basés sur l'étude et affinés par la pratique. Certaines résistances de valeur 1 ou 0 sont des pontages placés pour garder un circuit simple face.

Il en est de même pour la programmation. les bouts de codes ne sont qu'une étude théorique. L'assemblage des fonctions impose certaines modifications du code, seul le code fourni (fichiers PDE) est valable. Comme pour toute application non commercialisée, certains bugs peuvent apparaître. Malgré toutes les précautions qu'il préconise, l'auteur ne peut en aucun cas être tenu responsable de dégâts des biens ou personnels.

Etant données les puissances mises en jeu, et l'accès au potentiel secteur, toute utilisation ou modification du système est aux risques et périls de l'utilisateur. L'assemblage électronique et la manipulation ne devrait être fait que par une personnes parfaitement au courant des risques qu'elle encourt.

Utilisation des ressources :

Secteur

Trois phases qui peuvent être câblées comme on veut : tri, mono, biphasé...

Lors de la mise en route, le système devra reconnaître les phases et le décalage des phases 2 et 3 par rapport à la phase 1 (phase de référence). L'utilisation des signaux redressés, on ne s'intéressera qu'à une demi période du courant secteur (soit 180°).

Pour des raisons de câblage, les voies du bloc puissance sont réparties ainsi sur les phases :

| Phase | Voies |
|-------|------------|
| 1 | 01, 04, 07 |
| 2 | 02, 05, 08 |
| 3 | 03, 06, 09 |

Possibilités de câblage :

| Câblage | Description | Décalage phase 2 | Décalage phase 3 |
|-----------|---|------------------|------------------|
| Monophasé | Les trois phases sont reliées entre elles | 0° | 0° |
| Triphasé | Chaque phase est reliée à une phase du réseau | 60° | 120° |
| | | 120° | 60° |
| Biphasé | Deux phases sont reliées entre elles | 0° | 60° |
| | | 0° | 120° |
| | | 60° | 0° |
| | | 60° | 60° |
| | | 120° | 0° |
| | | 120° | 120° |
| Partiel | Une phase (ou deux) n'est (ne sont) pas câblée(s). On se retrouve dans l'une des configuration ci-dessus, avec la possibilité d'une phase qui ne "répond pas" | | |

Sources

Le système étant destiné aux applications scéniques, il convient de prendre en compte le fait qu'il se peut que la source de courant soit un groupe électrogène, possédant une fréquence moins stable que le réseau de distribution filaire. Une mesure de fréquence devra être prévue.

Bloc puissance DMX 12 voies

Entrées / sorties arduino Mega2560

| Opt | Broche | Port | Pin | Type | E/S | Fonction | Détail |
|-----|--------|------|-----|-----------------------|-------------------|----------|--|
| | OC1A | B5 | 11 | Sorties timers | S | OUT 01 | Commande Optotriac Circuit 01 |
| | OC1B | B6 | 12 | | S | OUT 02 | Commande Optotriac Circuit 02 |
| | OC1C | B7 | 13 | | S | OUT 03 | Commande Optotriac Circuit 03 |
| | OC3A | E3 | 5 | | S | OUT 04 | Commande Optotriac Circuit 04 |
| | OC3B | E4 | 2 | | S | OUT 05 | Commande Optotriac Circuit 05 |
| | OC3C | E5 | 3 | | S | OUT 06 | Commande Optotriac Circuit 06 |
| | OC4A | H3 | 7 | | S | OUT 07 | Commande Optotriac Circuit 07 |
| | OC4B | H4 | 8 | | S | OUT 08 | Commande Optotriac Circuit 08 |
| | OC4C | H5 | 9 | | S | OUT 09 | Commande Optotriac Circuit 09 |
| | OC5A* | L3 | 44 | | S | OUT 10 | Commande Circuit 10 |
| | OC5B* | L4 | 45 | | S | OUT 11 | Commande Circuit 11 |
| | OC5C* | L5 | 46 | | S | OUT 12 | Commande Circuit 12 |
| | INT0 | D0 | 20 | | Passage 0 phase 1 | E | PH1 |
| | INT1 | D1 | 19 | Passage 0 phase 2 | E | PH2 | Depuis détecteur 0V ph2 |
| | INT2 | D2 | 18 | Passage 0 phase 3 | E | PH3 | Depuis détecteur 0V ph3 |
| X | RX0 | E0 | 0 | Contrôle par USB | E | USB in | Via contrôleur FTDI |
| x | TX0 | E1 | 1 | | S | USB out | |
| | RX2 | H0 | 17 | Entrée DMX | E | DMX IN | Depuis circuit adaptateur DMX |
| X | TX2 | H1 | 16 | Sortie DMX | S | DMX OUT | Vers circuit adaptateur DMX |
| | AIN0 | F0 | A0 | Entrées analogiques | E | IN 01 | Depuis potars face AV et / ou entrées 0-10V pour les voies 01 à 09. Commandes "avancées" pour les voies 10 à 12 |
| | AIN1 | F1 | A1 | | E | IN 02 | |
| | AIN2 | F2 | A2 | | E | IN 03 | |
| | AIN3 | F3 | A3 | | E | IN 04 | |
| | AIN4 | F4 | A4 | | E | IN 05 | |
| | AIN5 | F5 | A5 | | E | IN 06 | |
| | AIN6 | F6 | A6 | | E | IN 07 | |
| | AIN7 | F7 | A7 | | E | IN 08 | |
| | AIN8 | K0 | A8 | | E | IN 09 | |
| | AIN9 | K1 | A9 | | E | IN 10 | |
| | AIN10 | K2 | A10 | | E | IN 11 | |
| | AIN11 | K3 | A11 | | E | IN 12 | |
| X | 22 | A0 | | Affichage LCD | S | | |
| X | 23 | A1 | | | S | | |
| X | 24 | A2 | | | S | | |
| X | 25 | A3 | | | S | | |
| X | 26 | A4 | | | S | | |
| X | 27 | A5 | | | S | | |
| X | 28 | A6 | | | S | | |
| X | 29 | A7 | | | S | | |
| | 41 | G0 | | Leds panneau AV | S | Ready | Vert |
| | 40 | G1 | | | S | DMX | Orange |
| | 39 | G2 | | | S | ERROR | Rouge |
| X | 37 | C0 | | Clavier configuration | E | | |
| X | 36 | C1 | | | E | | |
| X | 35 | C2 | | | E | | |
| X | 34 | C3 | | | E | | |
| X | 33 | C4 | | | S | | |
| X | 32 | C5 | | | S | | |
| X | 30 | C6 | | | S | | |
| X | 31 | C7 | | | S | | |
| | | | | | | | |
| | | | | | | | |

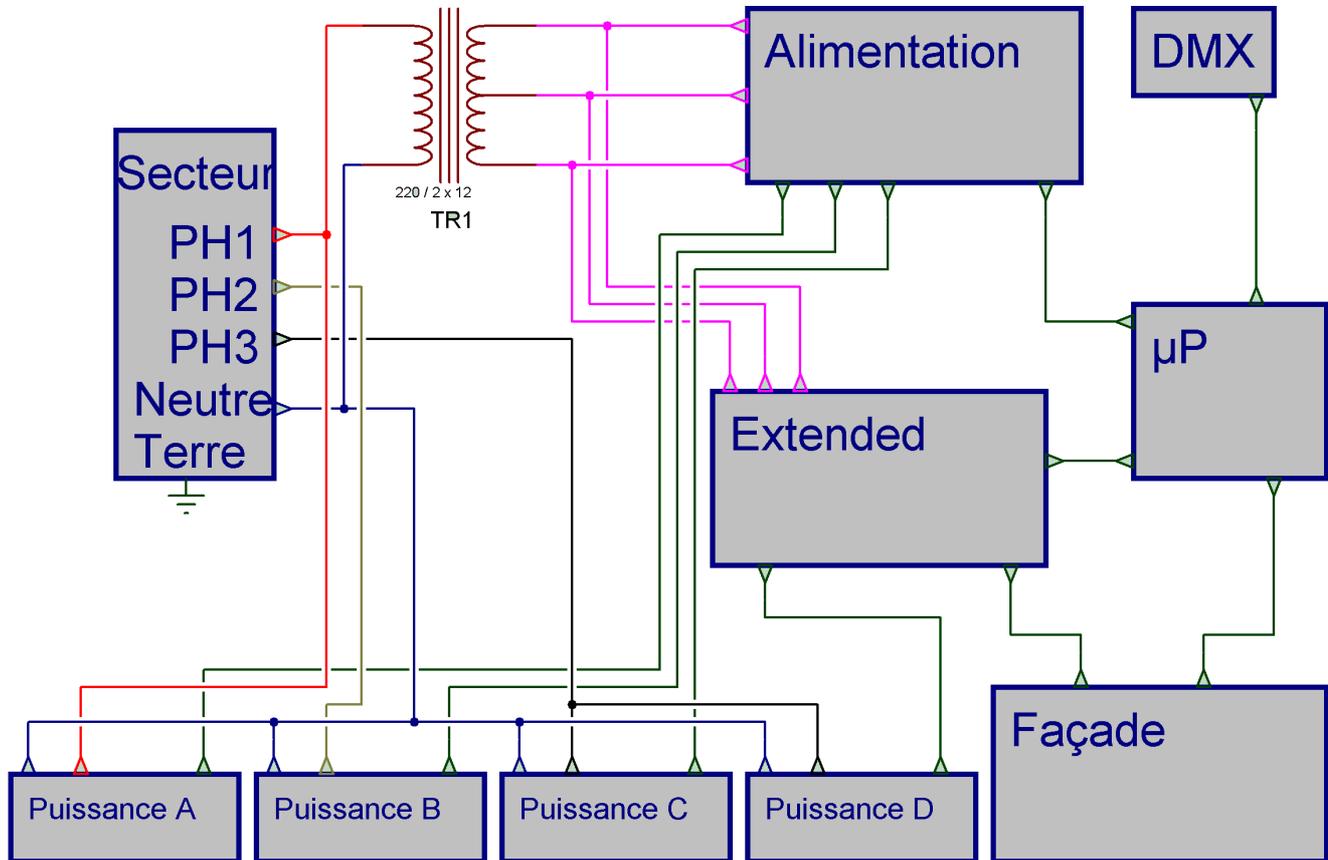
* : Utilisations à venir

Opt : en option, les branchements seront prévus mais non programmés ni utilisés pour l'instant.

Etude électronique

Assemblage des cartes

Schéma structurel



L'ensemble du système est constitué de cartes permettant le remplacement rapide d'une carte défectueuse.

Liaisons entre cartes

| Carte 1 | Carte 2 | Nombre contacts | Type connecteur | Signaux transportés | |
|--------------|--------------|-----------------|-----------------|-------------------------|-----------------|
| | | | | 1 -> 2 | 2 -> 1 |
| Secteur | Puissance A | 2 | | 220V / 60A | |
| Secteur | Puissance B | 2 | | 220V / 60A | |
| Secteur | Puissance C | 2 | | 220V / 60A | |
| Secteur | Puissance D | 2 | | 220V / 60A | |
| Secteur | TR1 | 2 | | 220V / 0,5A | |
| TR1 | Alimentation | 3 | Cosses | 2 x 12V AC | |
| TR1 | Extended | 3 | Cosses | 2 x 12V AC | |
| Alimentation | Puissance A | 8 | 2 x 4 | +5V Commande 3 voies | Passage à 0 ph1 |
| Alimentation | Puissance B | 8 | 2 x 4 | +5V Commande 3 voies | Passage à 0 ph2 |
| Alimentation | Puissance C | 8 | 2 x 4 | +5V Commande 3 voies | Passage à 0 ph3 |

Bloc puissance DMX 12 voies

| | | | | | |
|--------------|-------------|----|--------|--------------------------------------|--------------------------------|
| Alimentation | μP | 10 | 1 x 10 | +5V, +12V, -12V Passage 0 ph1,2,3 | |
| | | 10 | 1 x 10 | | Commande 9 voies |
| μP | DMX | 4 | 1 x 4 | +5V DMX OUT | DMX IN |
| μP | Façade | 16 | 2 x 8 | +5V | Commande 9 voies analogique |
| | | 10 | 1 x 10 | | Commande 9 voies |
| | | 24 | 2 x 13 | LCD Clavier | Clavier |
| μP | Extended | 8 | 2 x 4 | +5V, Commande 3 voies | Compteur T5 |
| Extended | Puissance D | 8 | 2 x 4 | +5V, Commande 3 voies | Passage à 0 ph3 |
| Extended | Façade | 4 | 1 x 4 | Commande 3 voies | |
| Façade | LCD | 16 | 1 x 16 | +5V Commandes LCD | |
| | | | | | |
| | | | | | |
| | | | | | |

Bloc puissance DMX 12 voies

Interface puissance dimmer

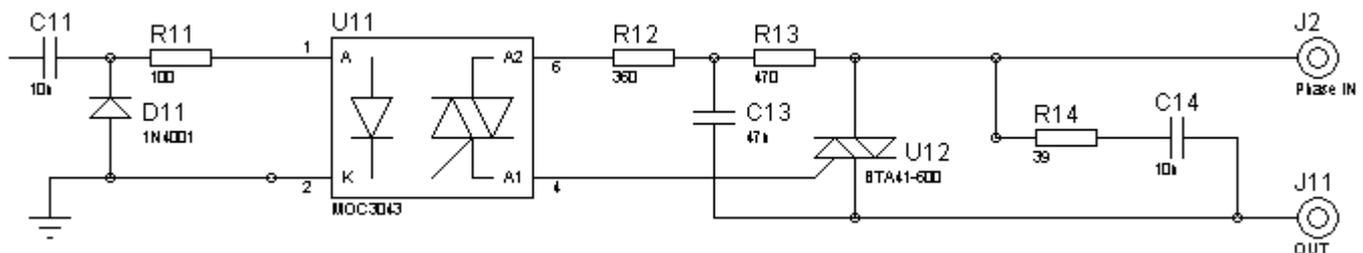
L'interface se compose de 3 cartes (A, B, C) comprenant chacune 3 circuits de puissance identiques. Une carte par phase.

Implantation des voies sur les cartes :

| Carte | Phase | Circuit 1 | | Circuit 2 | | Circuit 3 | |
|----------|----------|-----------|----------------|-----------|----------------|-----------|----------------|
| | | Voie | Broche μ P | Voie | Broche μ P | Voie | Broche μ P |
| A | 1 | 01 | OC1A | 04 | OC1B | 07 | OC1C |
| B | 2 | 02 | OC3A | 05 | OC3B | 08 | OC3C |
| C | 3 | 03 | OC4A | 06 | OC4B | 09 | OC4C |

Entrée : Commande numérique (0-5V), Secteur 220V

Sortie : Puissance 220V / 20A



Détail d'un circuit de puissance

Rôle des composants :

C11+R11+D11 : conversion de la commande en impulsions afin d'éviter tout retard de coupure. Le temps de l'impulsion utile est choisi à 5 μ s, avec une pointe de courant de 35mA (l'optotriac est donné pour un déclenchement à partir de 5mA pendant 100 μ s. Un pas DMX équivalent à 39 μ s, la réduction de l'impulsion demande d'augmenter la valeur moyenne du courant. Une impulsion supérieure au pas DMX risquerait de créer des interférences ou des flashes incontrôlés en sortie.

U11 : isolement galvanique entre la partie logique et la partie puissance.

R12 : limitation du courant dans l'optotriac

R13+C13 : filtrage pour l'optotriac.

R14+C14 : filtrage pour le triac.

U12 : triac 40A / 600V

Interface extended

En cours d'élaboration, toute option est envisageable.

Entrées : 3 signaux numériques de commande

Sorties : contacts de relais, signaux numériques...

A définir.

Protection du système, de l'utilisateur

Le système doit avant tout répondre aux normes actuelles. Il convient de respecter les points ci-dessous :

| Domaine | Caractéristiques | Norme | Respecter | A savoir |
|---------------------------|--|---|---|--|
| Alimentation | Secteur 240V 3 x 60A | Prise CE 5 pôles Mise à la terre de tout élément conducteur apparent | Prise 3 x 63A Câble d'alimentation 5x10mm ² | Le système doit être raccordé au réseau sur une prise répondant aux normes, protégée par un disjoncteur différentiel 3 x 60A / 30mA |
| Sorties électriques | 9 sorties monophasées 240V / 20A | Protection contre les surcharges et courts- circuits | Disjoncteur à double coupure 20A | Chaque sortie doit être protégée indépendamment. |
| | | | | |
| Accès aux branchements | Fiches spéciales | Accès restreint Protection contre l'arrachement des prises | Trappe d'accès aux branchements Système d'accrochage des câbles | Il est possible d'envisager la condamnation de la trappe par clé. |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

Bloc puissance DMX 12 voies

Interface DMX

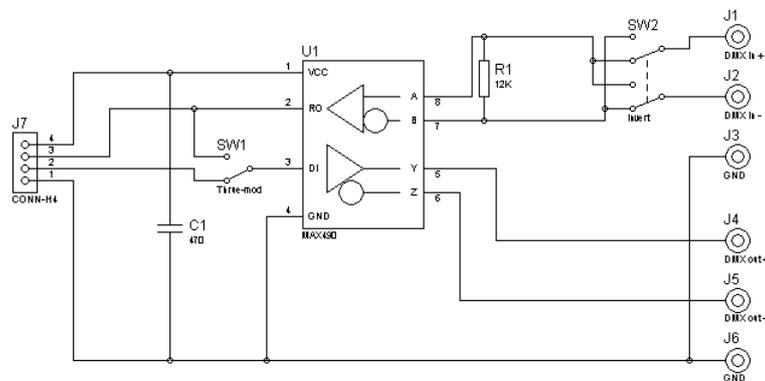
Plusieurs solutions se proposent à nous, car les signaux présents sur les deux fils de la liaison sont en opposition de phase. Le circuit MAX490 correspond tout à fait à ce que l'on cherche, et il permet de faire une recopie des signaux pour un nouveau départ DMX (régénération de signal).

Afin de maximiser les possibilités d'utilisation pour les futures versions, les fonctions suivantes sont prévues :

- Inversion par interrupteur des entrées DMX si le signal ne correspond pas.
- Recopie du signal DMX ou modification par le processeur (Opt).

Entrées / sorties : signaux DMX / TTL

Détail du circuit d'adaptation DMX



Rôle des composants :

U1 : adaptation des signaux DMX \Leftrightarrow TTL

SW2 : inversion DMXin en cas de réception d'un signal inversé

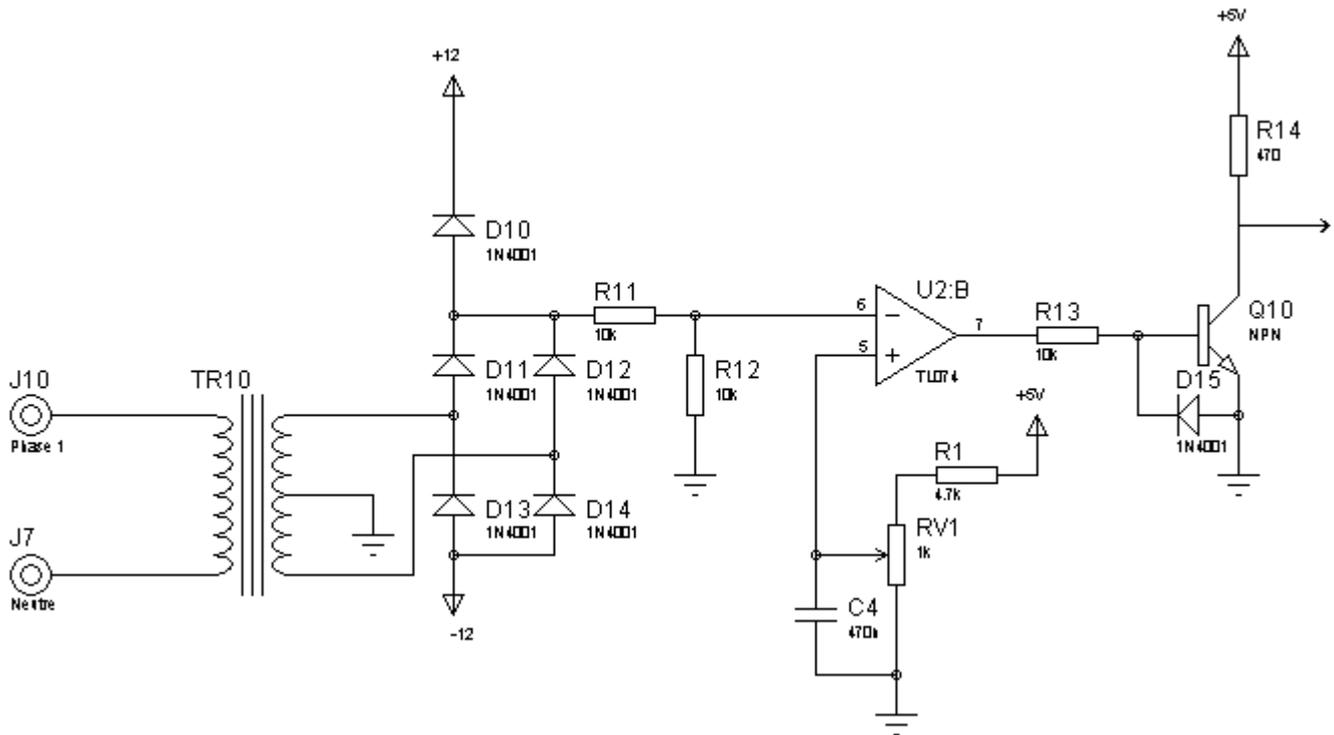
SW1 : Recopie du DMX ou transformation du signal par μ P (Opt.).

Alimentation BT et détection 0V secteur

Cette fonction doit permettre de générer une impulsion à chaque passage à 0 du secteur. Trois phases, donc trois circuits identiques.

Entrée : Sinus redressé mono alternance (un passage à 0 toutes les 10ms)

Sortie : Impulsion TTL positive (front descendant).



Détail d'un circuit de détection 0V secteur

Rôle des composants :

TR10 : transformateur 220V – 2 x 12V

D11 à D14 : pont de redressement double alternance

D10 : isolation du pont vers l'alimentation principale

R11+R12 : pont diviseur (divise par deux la tension en sortie de redresseur)

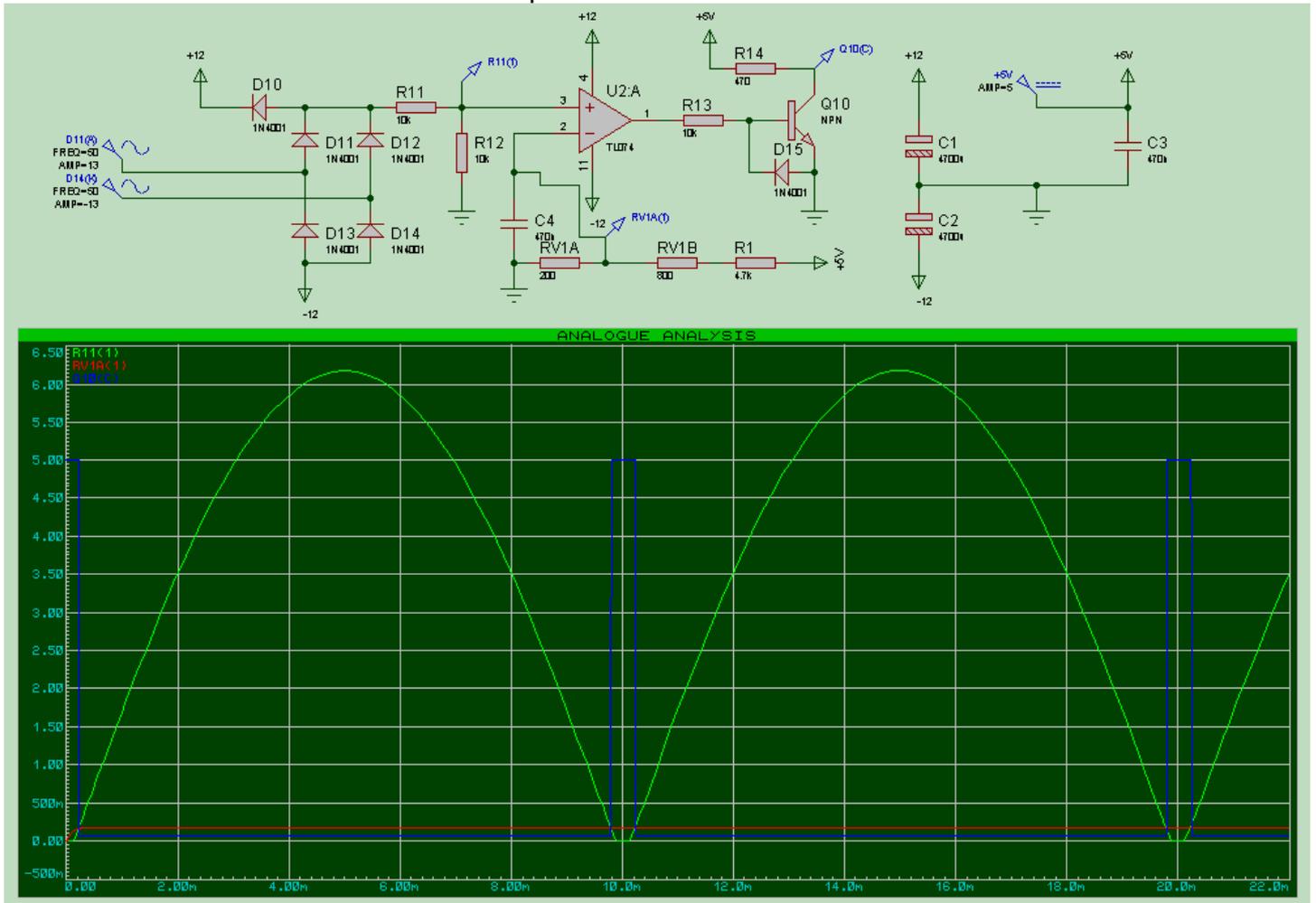
R1+RV1+C4 : tension de référence pour le comparateur

U2 : comparateur

R13+D15+Q10+R14 : adaptation du signal vers TTL

Simulations :

Bloc puissance DMX 12 voies



*Vert : le signal en entrée du comparateur. Sinus redressé double alternance.
 Rouge : tension de référence générée par RV1 (simulé ici par RV1A et RV1B)
 Bleu : impulsion résultante.*

RV1 permet de créer une avance de l'impulsion par rapport au signal secteur. En effet, les temps de traitement numérique et le retard induits par le transformateur peuvent être gênants. Le réglage de l'avance de l'impulsion permettra de palier à ces défauts. Du fait des seuils des diodes de redressement, l'avance minimum que l'on peut atteindre est de l'ordre de 200µs (soit entre 3° et 4°). Une correction logicielle permettra d'obtenir 0° pour une valeur DMX de 255.

Carte façade

Cette carte est l'interface "homme-machine". elle possède :

- 9 potentiomètres pour la commande en direct des 9 premières voies,
- 12 leds indiquant l'état des 12 voies
- 1 clavier 16 touches
- 1 afficheur LCD 20 x 4
- 5 leds d'état du système (power, dmx...)

C'est également elle qui impose le format 4U du boîtier, pour faire tenir toutes les commandes sur la face avant.

Etude programmation

Réception DMX

Bien que tendant à être remplacé dans l'avenir par l'Ethernet, ce protocole offre bien des avantages (longueur de transmission > 300m, rapidité, universel...). Les caractéristiques sont les suivantes :

Débit : 250Kb/s

Transmission : Asynchrone, 1 bit start, 2 bits stop, pas de parité

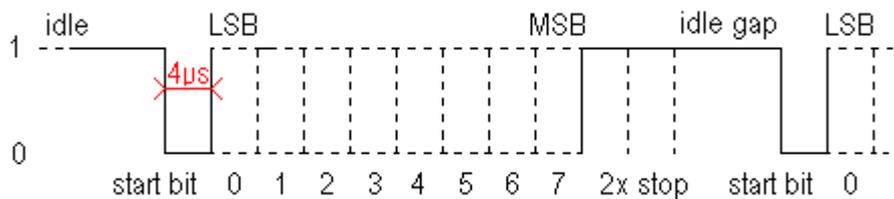
Nombre de voies : 512 (512 valeurs)

Durée d'une trame (envoi des 512 valeurs) : 22.668 ms

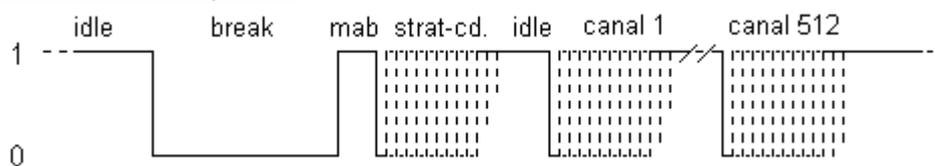
Fréquence des trames : de 1 à 44 trames par seconde maximum.

Transport : signal différentiel sur paire torsadée blindée.

Composition d'un octet :



Composition d'une trame complète :



Break (88µs), Mab (8µs) Start-code (44µs), données (512 x 44µs), Idle (??).

Analogie avec les liaisons séries : L'envoi des données est une liaison série classique, précédée d'une impulsion (break) de 88µs d'indication de départ de trame. Il suffit donc de démarrer la réception série dès la détection de l'impulsion "break".

Il est nécessaire d'utiliser un timer en plus d'un port série pour la détection du Break DMX (88µs).

Port série DMX in/out : USART2. Le USART0 est utilisé pour la programmation et communication (future option) USB. Les broches du USART1 sont utilisées par les interruption de détection de passage à zéro.

Timer : T2. Ce timer a la particularité d'être totalement indépendant des autres timers, donc ne craint pas les modifications apportées au prescaler commun.

Le code provient de Max Pierson. Il a été adapté au processeur MEGA2560 et traduit en français.

Le code suivant utilise directement l'interruption de réception de USART2, par conséquent, il faut supprimer (mettre en commentaire) la routine SIGNAL(SIG_USART2_RECV){} dans le fichier HardwareSerial.cpp du core arduino.

Code :

```

/***** Variables d'adressage *****/

```

Bloc puissance DMX 12 voies

```
#define NUMBER_OF_CHANNELS 12 //Le nombre de canaux que l'on désire recevoir (12 par défaut).
volatile word dmxaddress = 1; // adresse de départ de réception

#define RX_PIN 17 // PIN RX2
#define TX_PIN 16 // PIN TX2

/***** Variables DMX *****/

volatile byte i = 0; // indexeurs pour dmxvalue[]
volatile byte dmxreceived = 0;
volatile word dmxcurrent = 0; //compteur incrémenté à chaque réception de donnée.
volatile byte dmxvalue[NUMBER_OF_CHANNELS]; // tableau des valeurs DMX reçues
volatile boolean dmxnewvalue = false; // indique la reception de nouvelles données DMX
volatile byte zerocounter = 0; // Pour compter le nombre de zeros consécutifs
// reçus sur RX3. 22 zéros signifient un break (88µs).

void dmx_setup() {

    // Configuration du port série RX

    pinMode(RX_PIN, INPUT); // RX_PIN en entrée
    Serial2.begin(250000); //chaque bit DMX fait 4uS de long, soit 250Kbps
    cli(); //Désactivation des interruptions
    UCSR2B &= 0x7F; //Désactiver l'interruption de réception USART2

    // Timer2 : configuration

    TCCR2A = 0x02; // timer en mode 2, CTC. Le TOP sera OCRA.
    TCCR2B = 0x01; // pas de prescaler, 16Mhz
    OCR2A = 64; // Le timer 2 génèrera une interruption toutes les 4µs.
    TIMSK2 = 0x02; // autorise l'interruption sur OCR2A
    sei(); // Réactiver les interruptions.
}

void loop() {

    if (dmxnewvalue) { // Réception d'un flot de données : traitement.
        action(); // procédure de traitement des données DMX
        dmxnewvalue = FALSE; // Réinitialiser les variables de réception DMX
        zerocounter = 0;
        dmxcurrent = 0;
        i = 0;
        TIMSK2 |= 0x02; // relancer la recherche de trame DMX
    }
}

ISR(TIMER2_COMPA_vect) { // interruption appelée par timer2 toutes les 4µs
    if (PINH & 1) { // Détection d'un 1 : ce n'est pas un break.
        zerocounter = 0;
    } else {
        zerocounter++; // incrémentation du compteur de zéros
        if (zerocounter == 20) { // si 20 zéros reçus consécutivement (80uS : break)
            TIMSK2 &= 0xFD; // désactiver timer2, place au travail du USART2
            UCSR2B |= 0x80; // activation de USART2
        }
    }
}

ISR(USART2_RX_vect){ // interruption appelée par le port série.

    dmxreceived = UDR2;
    dmxcurrent++; //incrémentation du compteur d'adresse

    if(dmxcurrent >= dmxaddress) { // vérification de l'adresse de démarrage
        dmxvalue[i] = dmxreceived;
        i++;
        if(i == NUMBER_OF_CHANNELS) { // On a atteint le nombre de canaux à recevoir, On s'arrête là.
            UCSR2B &= 0x7F; // On désactive la réception DMX le temps d'exécuter
            dmxnewvalue = 1; // le traitement des données reçues
        }
    }
}
```

}

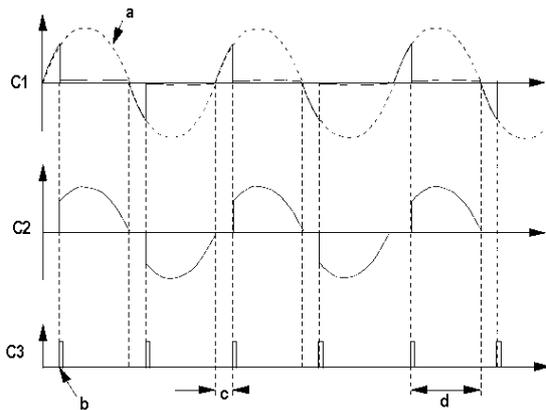
Traitement des données DMX reçues

Les données DMX se trouvent dans le tableau de variables dmxvalues[].

Chaque valeur (de 0 à 255) correspond à un angle d'attaque (de 0° à 180°) du triac de puissance.

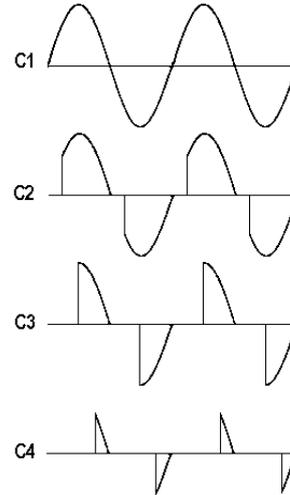
Les timers T1 à T4 comptent de 0 à 20 000 (en 10ms).

Détails impulsion / conduction du triac :



- C1 : tension aux bornes A2 et A1 du triac
- C2 : tension aux bornes de la charge
- C3 : impulsions de commande du triac
- a : pleine onde (tension secteur)
- b : impulsion de commande retardée
- c : retard ou angle d'attaque
- d : temps de conduction du triac

Forme des signaux à obtenir en sortie du triac :



- C1 : Angle d'attaque : 0° (valeur 255)
- C2 : Angle d'attaque : 39° (valeur 201)
- C3 : Angle d'attaque : 90° (valeur 128)
- C4 : Angle d'attaque : 138° (valeur 60)

Par conséquent, il convient d'appliquer la formule suivante permettant d'obtenir le bon angle d'attaque :

Code :

```
OCRnX = word((255 - dmxvalue[voie]) * FACTEUR);
```

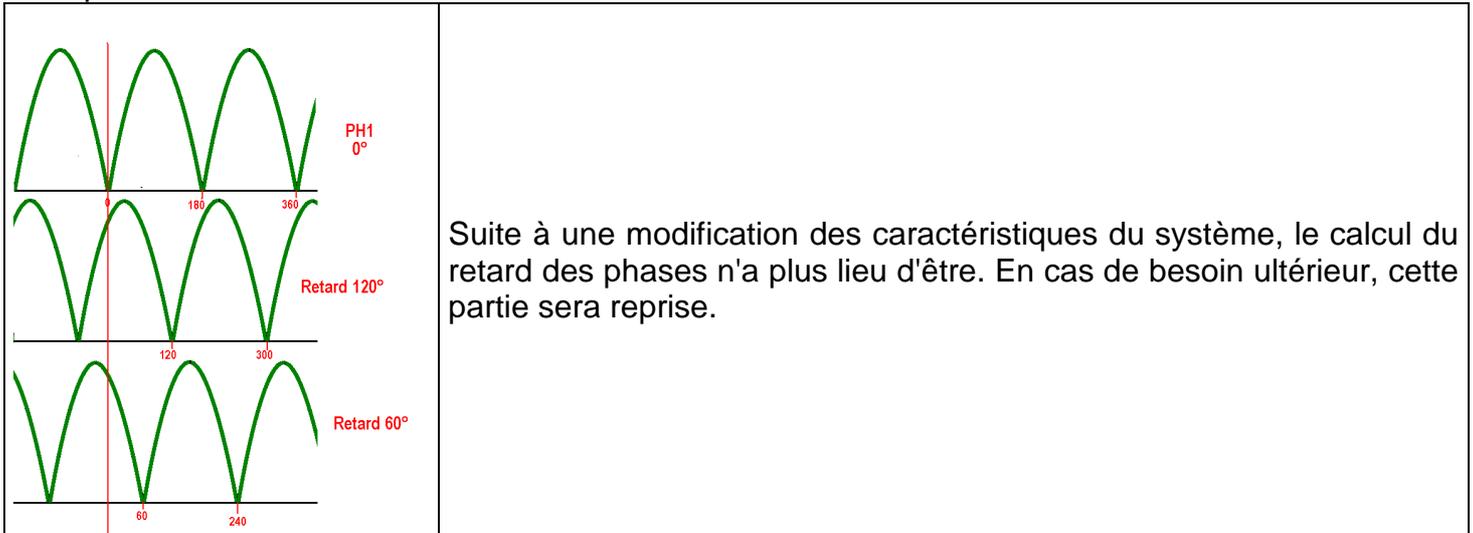
Mais auparavant, il faut inclure le retard de la phase sur laquelle la voie se trouve et la correction de l'avance du passage à zéro secteur. Les retards de phases sont inclus dans le tableau phase_retard[] contenant le retard à appliquer pour chaque voie (correction du retard passage à zéro). Nous simplifions donc le calcul ainsi :

Code :

```
OCRnX = (phase_retard[voie] - word(dmxvalue[voie] * FACTEUR));
```

Calculs des retards de phases

A la mise en route du système, on calcule le retard des phases 2 et 3. Deux valeurs de retard sont possibles :



Méthode de mesure de la fréquence du réseau

A la fréquence correspondra une variable "PERIODE" représentant le nombre qu'atteint un timer sur une demi alternance (180°). Cette variable permettra d'être plus précis dans le calcul des coefficients des angles d'ouverture.

Pour mesurer la fréquence, il suffit de démarrer un timer sur l'impulsion du détecteur de passage à 0, puis de lire la valeur du timer à l'impulsion suivante.

Méthode de mesure des retards de phase

Le passage à 0 des phases est fait sur les INT0, 1 et 2. Nous allons utiliser la phase 1 en référence, son passage à 0 déclenchera un timer, et la phase 2 arrêtera le timer. il suffira de lire la valeur du timer pour trouver le retard. Nous procéderons de même pour la phase 3. Si le timer fait un tour complet, c'est que la phase concernée n'est pas alimentée.

Timer utilisé : T5

Tempo de comptage : 0,5µs (Fclk/8 = 2MHz).

Débordement du compteur au bout de : 32,768ms.

Valeur du compteur pour un retard de :

| retard | 50Hz (+/- 1%) | | | 60Hz (+/- 1%) | | |
|---------------|-------------------------|--------|--------|---------------|--------|--------|
| | Min | TYP | Max | Min | TYP | Max |
| 0° ou 180° | 19 800 | 20 000 | 20 200 | 16 499 | 16 666 | 16 832 |
| 120° | 13 199 | 13 333 | 13 466 | 10 999 | 11 111 | 11 222 |
| 60° | 6 599 | 6 666 | 6 732 | 5 499 | 5 555 | 5 610 |
| Phase absente | débordement du compteur | | | | | |

Code :

```
volatile word PERIODE; // période du signal redressé.
volatile boolean ph1, ph2, ph3; // 1 = présence phase isolée
volatile boolean ph12, ph13, ph23, ph123; // indicateurs de couplage des phases
volatile boolean ph_mesure; // flag temp
volatile word ph_retard2, ph_retard3;
```

Bloc puissance DMX 12 voies

```
void test_phase () { // procédure de détection et mesure des phases
    int dummyw1, dummyw2, dummyw3;
    byte b_ph12, b_ph_13;

    // Timer5 : configuration pour les mesures des 3 phases

    cli(); // Désactiver les interruptions.
    TCCR5A = 0x00; // timer en mode 0, normal. Le TOP sera 0xFFFF
    TCCR5B = 0x00; // T1 désactivé

    // ***** Mesure de la fréquence secteur *****

    TIMSK5 = 0x01; // autorise l'interruption sur overflow
    TCNT5 = 0x0000; // initialisation du timer
    attachInterrupt(2, mesure_ph1, FALLING); // active l'interruption de passage à 0 secteur phase 1
    ph_mesure = FALSE;
    sei(); // Réactiver les interruptions.
    while(!ph_mesure); // attente de la fin de la mesure
    ph1 = HIGH;

    // ***** Mesure du retard de la phase 2 *****

    cli(); // Désactiver les interruptions.
    TIMSK5 = 0x01; // autorise l'interruption sur overflow
    TCNT5 = 0x0000; // initialisation du timer
    attachInterrupt(2, mesure_ph1, FALLING); // active l'interruption de passage à 0 secteur phase 1
    attachInterrupt(3, mesure_ph23, FALLING); // active l'interruption de passage à 0 secteur phase 2
    ph_mesure = FALSE;
    sei(); // Réactiver les interruptions.
    while(!ph_mesure); // attente de la fin de la mesure
    ph_retard2 = ph_retard3;

    // ***** Mesure du retard de la phase 3 *****

    cli(); // Désactiver les interruptions.
    TIMSK5 = 0x01; // autorise l'interruption sur overflow
    TCNT5 = 0x0000; // initialisation du timer
    attachInterrupt(2, mesure_ph1, FALLING); // active l'interruption de passage à 0 secteur phase 1
    attachInterrupt(4, mesure_ph23, FALLING); // active l'interruption de passage à 0 secteur phase 3
    ph_mesure = FALSE;
    sei(); // Réactiver les interruptions.
    while(!ph_mesure); // attente de la fin de la mesure

    // ***** Calcul du couplage des phases *****

    dummyw1 = PERIODE / 3;
    dummyw2 = PERIODE - ph_retard2;
    dummyw3 = PERIODE - ph_retard3;
    if (abs(dummyw2) < 2000) { // phase2 = 0°
        b_ph2 = 1;
    } else if (dummyw2 < (dummyw1 + 2000)) { // phase2 = 240°
        b_ph2 = 3;
    } else if (dummyw2 < ((2 * dummyw1) + 2000)) { // phase2 = 120°
        b_ph2 = 2;
    } else { // phase 2 absente
        b_ph2 = 0;
    }
    if (abs(dummyw3) < 2000) { // phase3 = 0°
        b_ph3 = 1;
    } else if (dummyw3 < (dummyw1 + 2000)) { // phase3 = 240°
        b_ph3 = 3;
    } else if (dummyw3 < ((2 * dummyw1) + 2000)) { // phase3 = 120°
        b_ph3 = 2;
    } else { // phase 3 absente
        b_ph3 = 0;
    }
    }
    ph12 = LOW;
    ph13 = LOW;
    ph23 = LOW;
    ph1 = HIGH;
    ph2 = (b_ph2 > 0); // phase 2 présente
}
```

Bloc puissance DMX 12 voies

```
ph3 = (b_ph3 > 0); // phase 3 présente
if ((b_ph2 == 1) && (b_ph3 == 1)){ // phases 1, 2, 3 couplées
    ph123 = HIGH;
    ph1 = LOW;
    ph2 = LOW;
    ph3 = LOW;
} else {
    ph123 = LOW;
    if ((b_ph2 == b_ph3) && (b_ph2 > 0)) { // phases 2, 3 couplées
        ph23 = HIGH;
        ph1 = HIGH;
        ph2 = LOW;
        ph3 = LOW;
    } elseif (b_ph2 == 1) { // phases 1, 2 couplées
        ph12 = HIGH;
        ph1 = LOW;
        ph2 = LOW;
    } elseif (b_ph3 == 1) { // phases 2, 3 couplées
        ph13 = HIGH;
        ph1 = LOW;
        ph3 = LOW;
    }
}
}

void mesure_freq1(){ // appelée par INT0 pour démarrer la mesures de la fréquence secteur
    TCCR5B = 0x02; // démarrer T5, prescaler = 1/8, 2MHz / 0.5µs
    detachInterrupt(2); // désactive l'interruption 0
    attachInterrupt(2, mesure_freq2, FALLING); // active l'int de passage à 0 secteur phase 1
}

void mesure_freq2(){ // appelée par INT0 pour terminer la mesures de la fréquence secteur
    TCCR5B = 0x00; // Stoppe T5
    TIMSK5 = 0x00; // désactive l'interruption sur overflow
    detachInterrupt(2); // désactive l'interruption 0
    PERIODE = TCNT5; // lecture du déphasage
    ph_mesure = TRUE; // mesure effectuée
}

void mesure_ph1(){ // appelée par INT0 pour démarrer les mesures de phases
    TCCR5B = 0x02; // démarrer T5, prescaler = 1/8, 2MHz / 0.5µs
    detachInterrupt(2); // désactive l'interruption 0
}

void mesure_ph23(){ // appelée par INT1 ou 2 pour les mesures de phases
    TCCR5B = 0x00; // Stoppe T5
    TIMSK5 = 0x00; // désactive l'interruption sur overflow
    detachInterrupt(3); // désactive l'interruption INT1
    detachInterrupt(4); // désactive l'interruption INT2
    ph_retard3 = TCNT5; // lecture du déphasage
    ph_mesure = TRUE; // mesure effectuée
}

ISR(TIMER5_OVF_vect){ // interruption appelée si phase absente (débordement compteur T5)
    TCCR5B = 0x00; // Stoppe T5
    TIMSK5 = 0x00; // désactive l'interruption sur overflow T5
    detachInterrupt(2); // désactive l'interruption INT0
    detachInterrupt(3); // désactive l'interruption INT1
    detachInterrupt(4); // désactive l'interruption INT2
    ph_retard3 = 0; // lecture du déphasage
    ph_mesure = TRUE; // mesure effectuée
}
```

Choix de la configuration des timers pour le dimmer

Les timers retenus T1, T3 et T4 possèdent tous quatre le même fonctionnement et trois comparateurs.

Organisation des voies sur les sorties des timers :

| Timer Tn | Sorties de comparaison | | |
|-----------|------------------------|------|------|
| | OCnA | OCnB | OCnC |
| T1 | 01 | 04 | 07 |
| T3 | 02 | 05 | 08 |
| T4 | 03 | 06 | 09 |

Orientation des ports de sortie :

| | | | | | | | | | |
|----------------------------|------|------|------|------|------|------|------|------|---------------|
| DDRB (T1) | DDB7 | DDB6 | DDB5 | DDB4 | DDB3 | DDB2 | DDB1 | DDB0 | |
| | 1 | 1 | 1 | x | x | x | x | x | = 0xE0 |
| DDRE (T3) | DDE7 | DDE6 | DDE5 | DDE4 | DDE3 | DDE2 | DDE1 | DDE0 | |
| | x | x | 1 | 1 | 1 | x | x | x | = 0x38 |
| DDRH (T4) | DDH7 | DDH6 | DDH5 | DDH4 | DDH3 | DDH2 | DDH1 | DDH0 | |
| | x | x | 1 | 1 | 1 | x | x | x | = 0x38 |

1 = broche en sortie

Afin de garder une précision honnête, nous choisirons le prescaler le plus bas possible (la fréquence la plus élevée).

Avec le prescaler à 1:8, les timers tourneront à 2MHz (soit la possibilité de compter pendant 32,768 ms). Ils seront remis à zéro toutes les 10ms par le passage à zéro de la phase de référence. Sans prescaler, les timers feraient un tour complet en 4,096ms, ce qui n'est pas assez.

Les timers fonctionneront en mode normal (comptage permanent), et la sortie OCnX sera mise à 1 quand le timer Tn arrivera à la valeur OCRnX.

Pour cela, les registres de configuration des timers seront :

WGMn3:0 = 14 : Fonctionnement de Tn : Fast PWM, TOP = ICRn.

COMnX1:0 = 3 : Mise à 1 de OCnX à TCNTn = OCRnX, mise à 0 de OCnX à TCNTn = 0x0000.

CSn2:0 = 2 : Prescaler à 1/8, Fclk = 16MHz/8 = 2MHz.

| | | | | | | | | | |
|---------------|--------|--------|--------|--------|--------|--------|-------|-------|-------------|
| TCCRnA | COMnA1 | COMnA0 | COMnB1 | COMnB0 | COMnC1 | COMnC0 | WGMn1 | WGMn0 | |
| | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0xFE |

| | | | | | | | | | |
|---------------|-------|-------|---|-------|-------|------|------|------|-------------|
| TCCRnB | ICNCn | ICESn | - | WGMn3 | WGMn2 | CSn2 | CSn1 | CSn0 | |
| | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0x1A |

Code :

```
volatile word AVANCE_0 = 5; // Paramètre d'avance RAZ timer, (2.5µs par défaut)
volatile word ICR1_1, OICR2_1, ICR3_1; // valeurs de RAZ des timers.
```

```
// Sous-entendu que les mesures de phases aient déjà été faites.
```

```
Void t134_setup() { // initialisation des timers 1, 3 et 4
```

```
    DDRB |= 0xE0; // T1 : voies 01, 04, 07
```

```
    DDRE |= 0x38; // T3 : voies 02, 05, 08
```

Bloc puissance DMX 12 voies

```
DDRH |= 0x38;           // T4 : voies 03, 06, 09
TCCR1A = 0xFC;         // Utilisation des trois comparateurs du timer 1
TCCR3A = 0xFC;         // Utilisation des trois comparateurs du timer 3
TCCR4A = 0xFC;         // Utilisation des trois comparateurs du timer 4
TCCR1B = 0x00;         // Préparation du timer 1, mise en attente
TCCR3B = 0x00;         // Préparation du timer 3, mise en attente
TCCR4B = 0x00;         // Préparation du timer 4, mise en attente
ICR1 = PERIODE + 500;  // Valeur arbitraire de TOP T1
ICR3 = PERIODE + 500;  // Valeur arbitraire de TOP T3
ICR4 = PERIODE + 500;  // Valeur arbitraire de TOP T4
ICR1_1 = ICR1 - AVANCE_0; // Valeur pour RAZ T1
ICR3_1 = ICR3 - AVANCE_0; // Valeur pour RAZ T3
ICR4_1 = ICR4 - AVANCE_0; // Valeur pour RAZ T4
}
```

Interruption du passage à 0 secteur

Les interruptions choisies sont INT0, INT1, INT2.

Mise à zéro des timers et des sorties OCnX :

Un timer par phase, donc chaque INT concerne un seul timer.

L'instant de remise à zéro des sorties dépend du branchement et du retard des phases. Les sorties OCnX sont réinitialisée quand le timer passe à 0 (de ICRn à 0). Pour garantir ce passage à 0 du timer, nous affecterons au timer la valeur $ICRn_1 = ICRn - A$, A étant un paramètre d'avance réglable. Afin d'éviter que deux interruptions soient appelées en même temps, nous allons écrire plusieurs fonctions que nous assignerons aux interruptions (INT0:2) en fonction du câblage détecté à la mise sous tension. Exemple : en monophasé, seule la fonction `raz_ph123` sera appelée.

Code :

```

void raz_ph1() { // phase de référence, mode phase isolée
    TCNT1 = ICR1_1; // RAZ T1
}

void raz_ph2() { // phase 2, mode phase isolée
    TCNT3 = ICR3_1; // RAZ T3
}

void raz_ph3() { // phase 3, mode phase isolée
    TCNT4 = ICR4_1; // RAZ T4
}

Void raz_ph12() { // phase de référence, 1 et 2 couplées
    TCNT1 = ICR1_1; // RAZ T1
    TCNT3 = ICR3_1; // RAZ T3
}

Void raz_ph13() { // phase de référence, 1 et 3 couplées
    TCNT1 = ICR1_1; // RAZ T1
    TCNT4 = ICR4_1; // RAZ T4
}

void raz_ph23() { // phase 2 et 3 couplées
    TCNT3 = ICR3_1; // RAZ T3
    TCNT4 = ICR4_1; // RAZ T4
}

Void raz_ph123() { // phase de référence, mode 3 phases couplées
    TCNT1 = ICR1_1; // RAZ T1
    TCNT3 = ICR3_1; // RAZ T3
    TCNT4 = ICR4_1; // RAZ T4
}

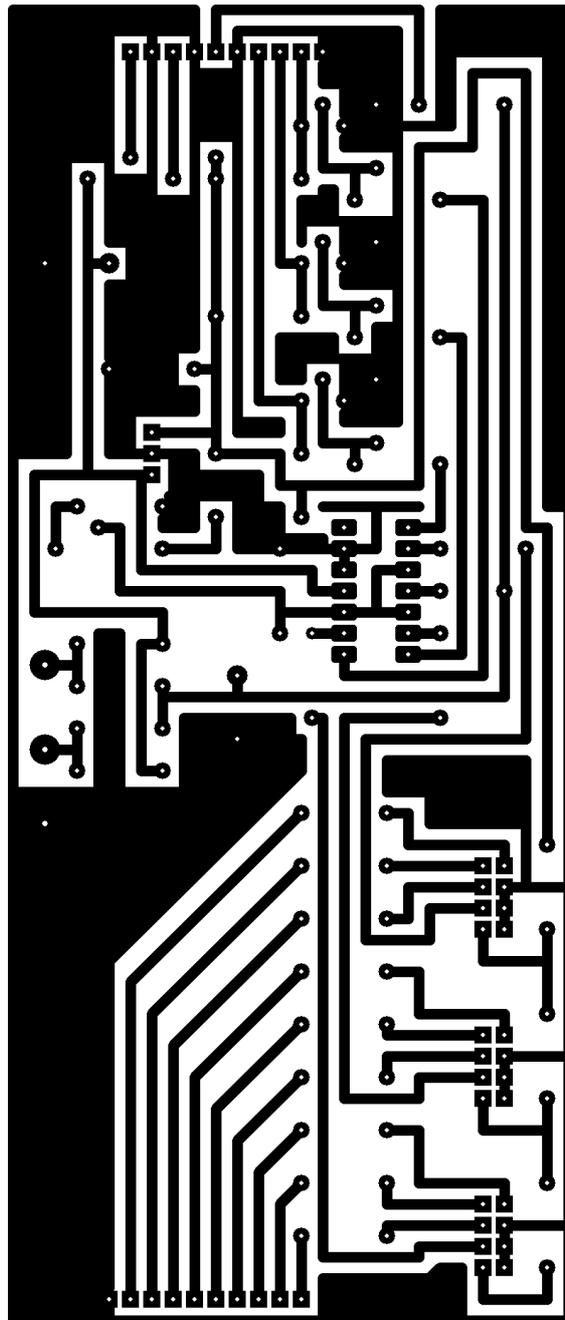
```

ANNEXES

Carte Alimentation

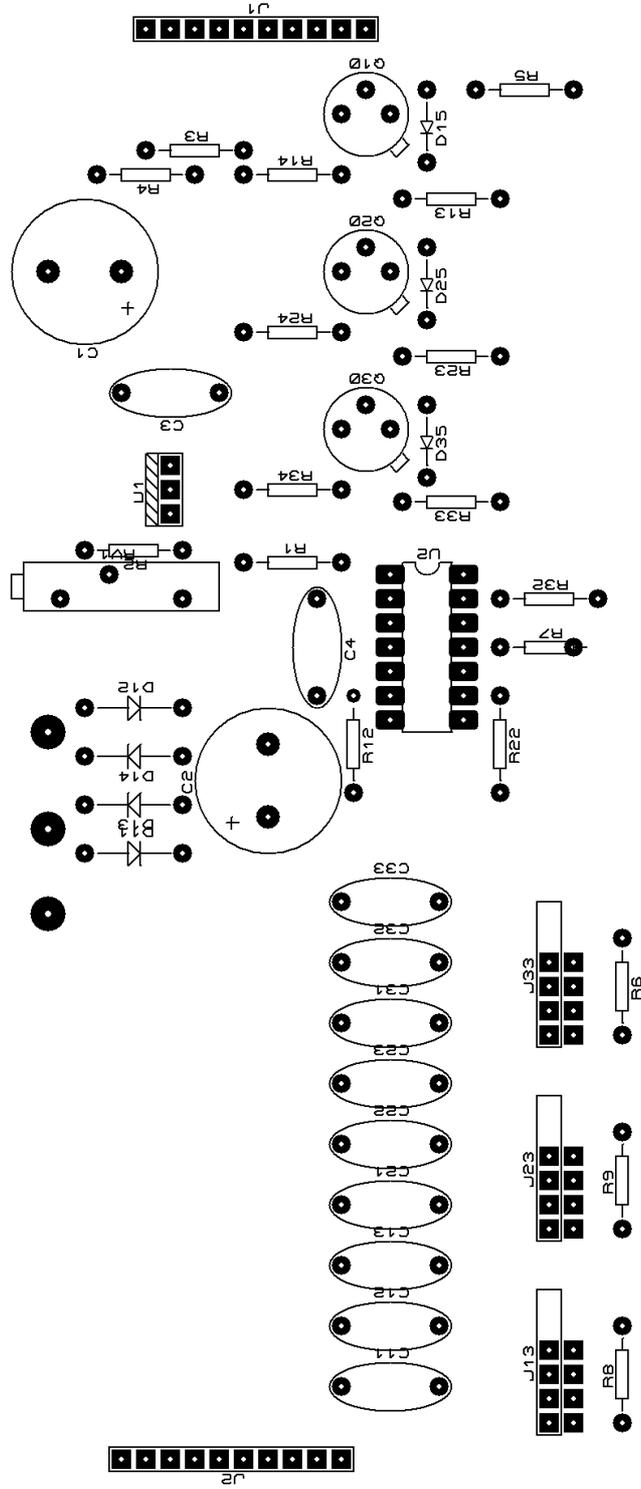
Schéma

Typon



Bloc puissance DMX 12 voies

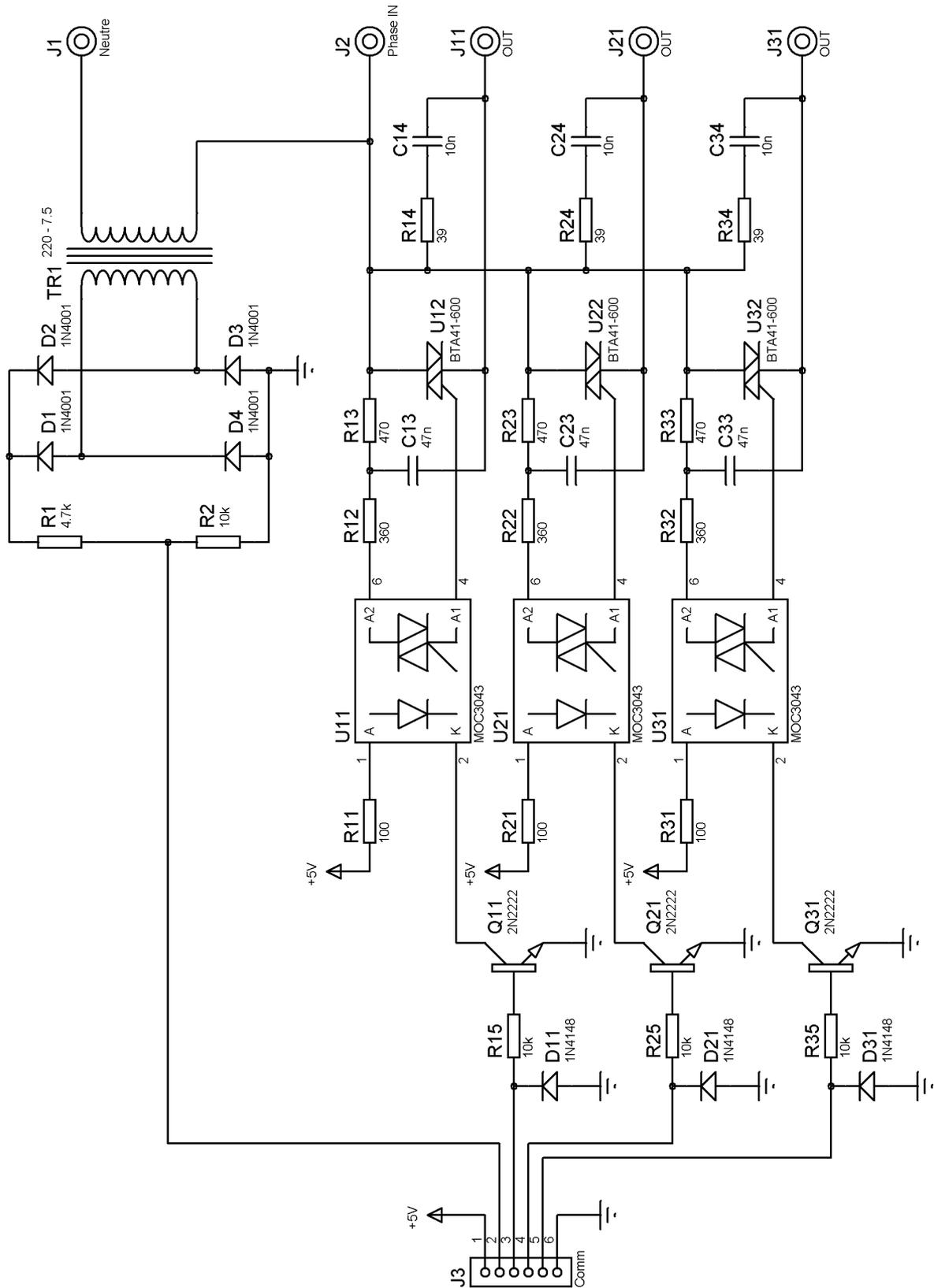
Implantation



Bloc puissance DMX 12 voies

Carte Puissance A, B, C et D

Schéma

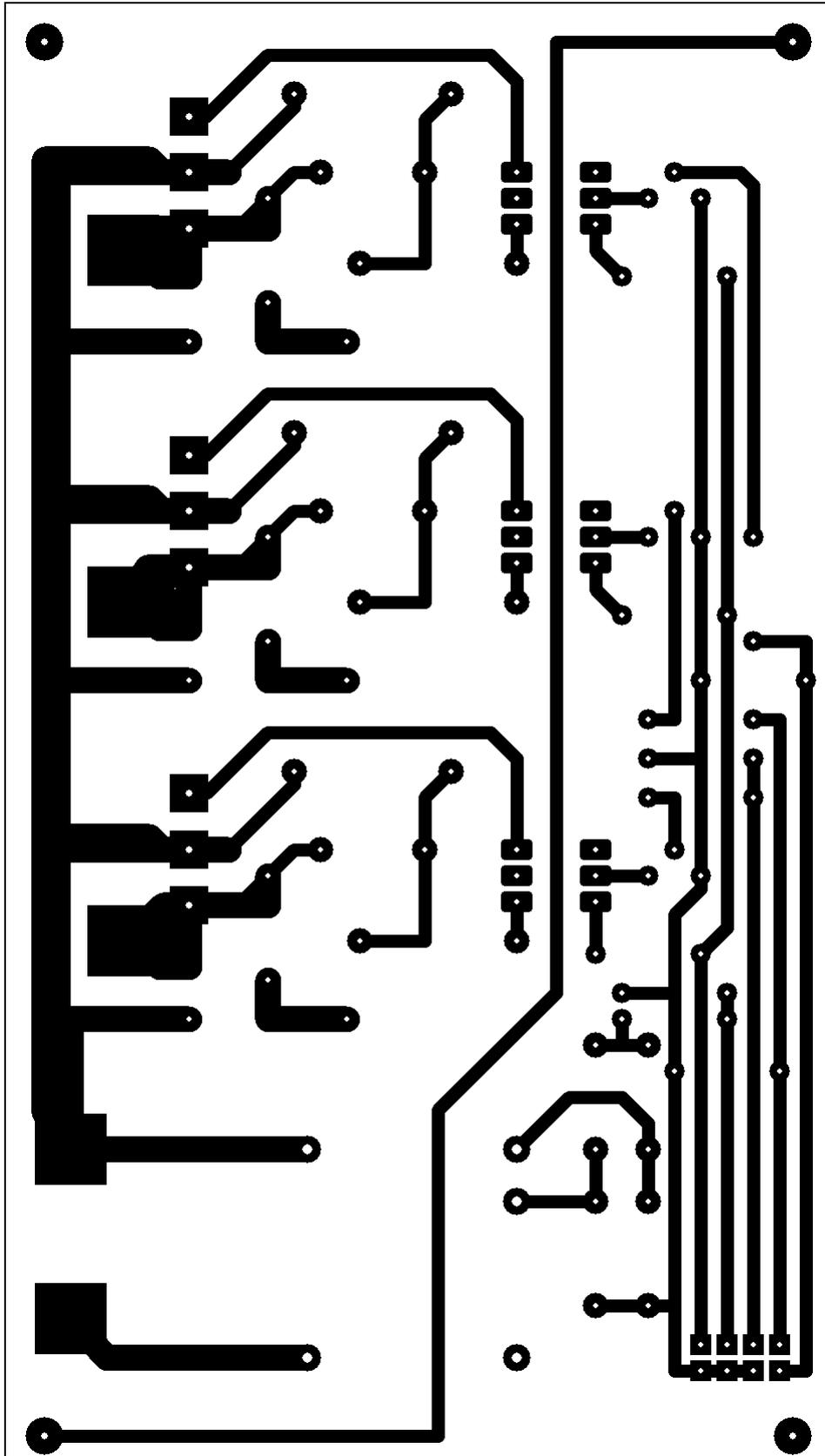


Bloc puissance DMX 12 voies

RB1204 Carte Puissance ver 1

Bloc puissance DMX 12 voies

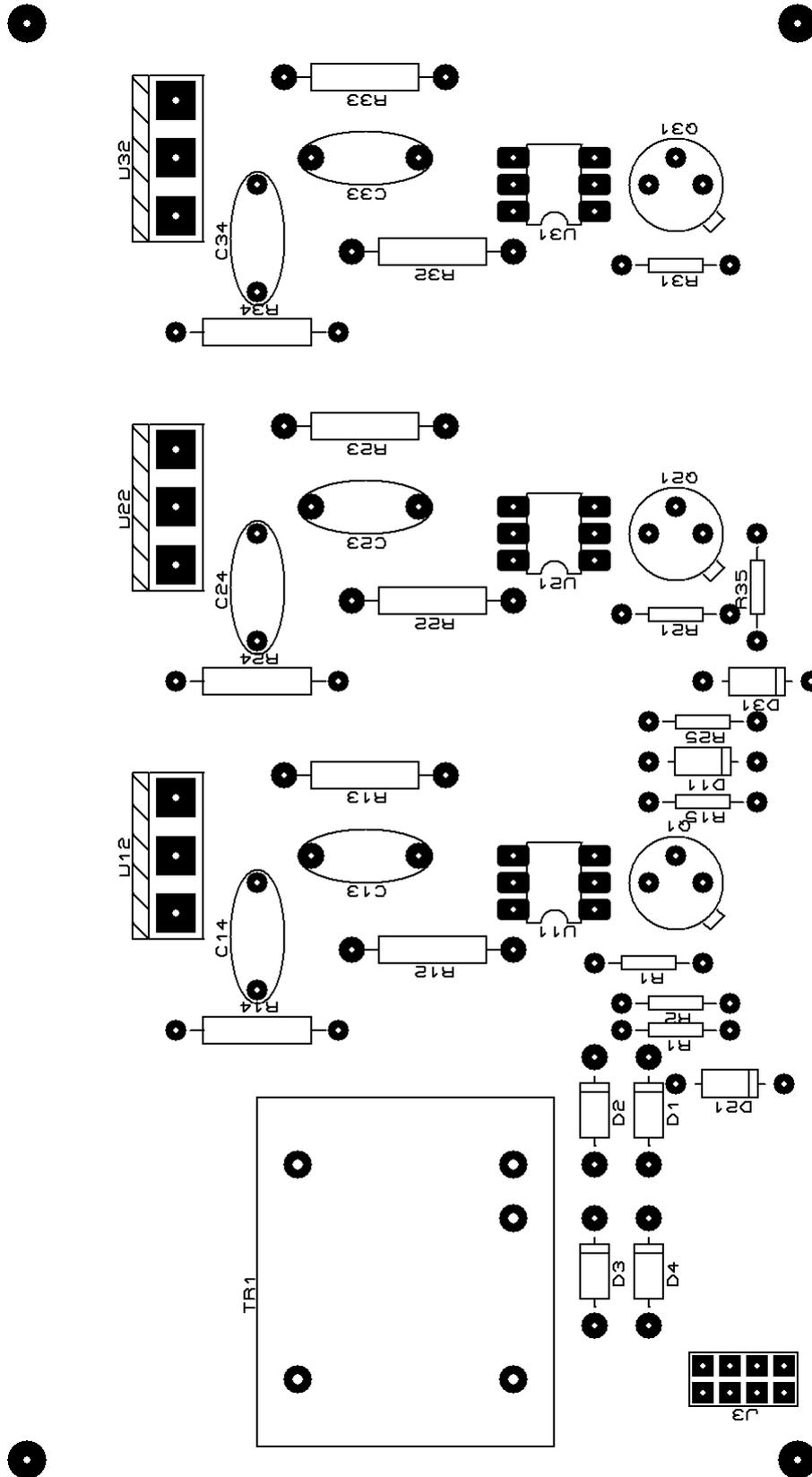
Typon



RB1204 Carte Puissance ver 1

Bloc puissance DMX 12 voies

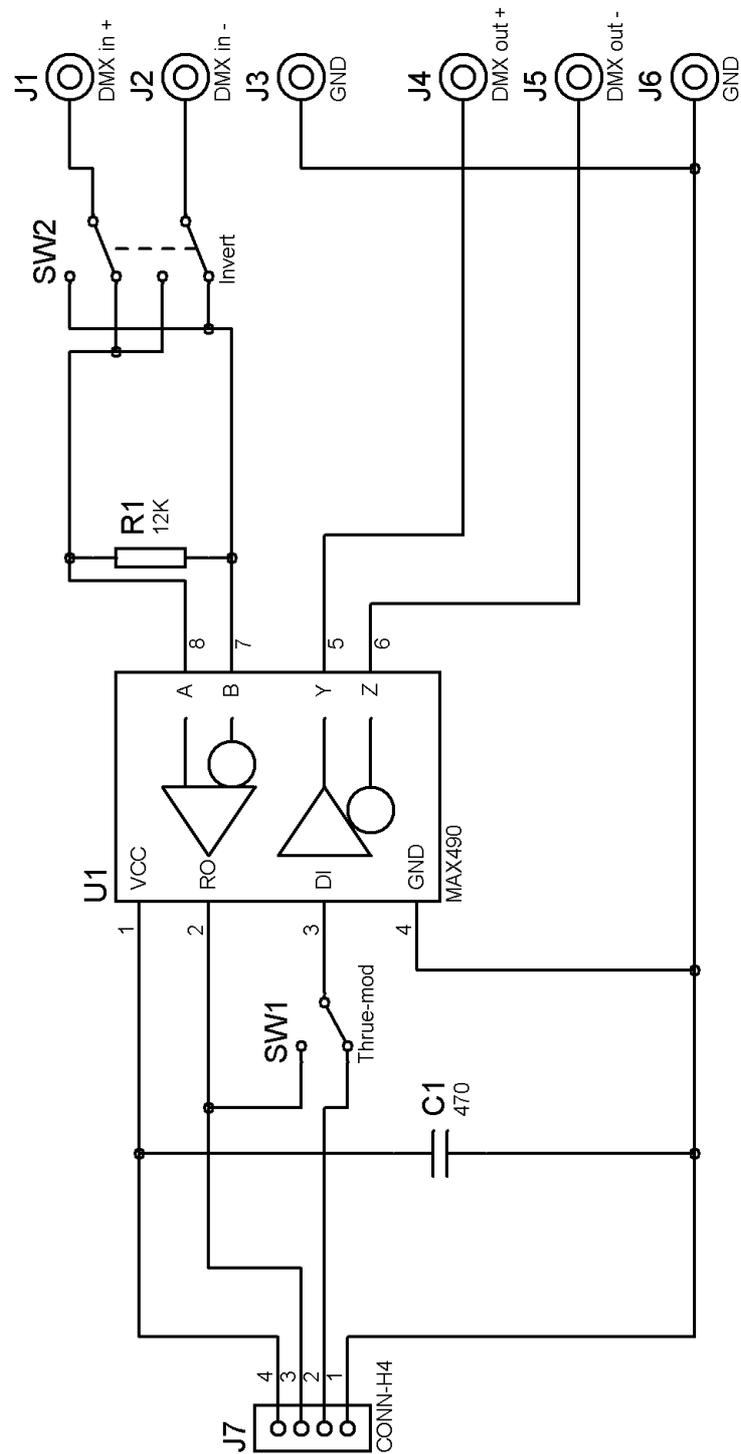
Implantation



Bloc puissance DMX 12 voies

Carte DMX

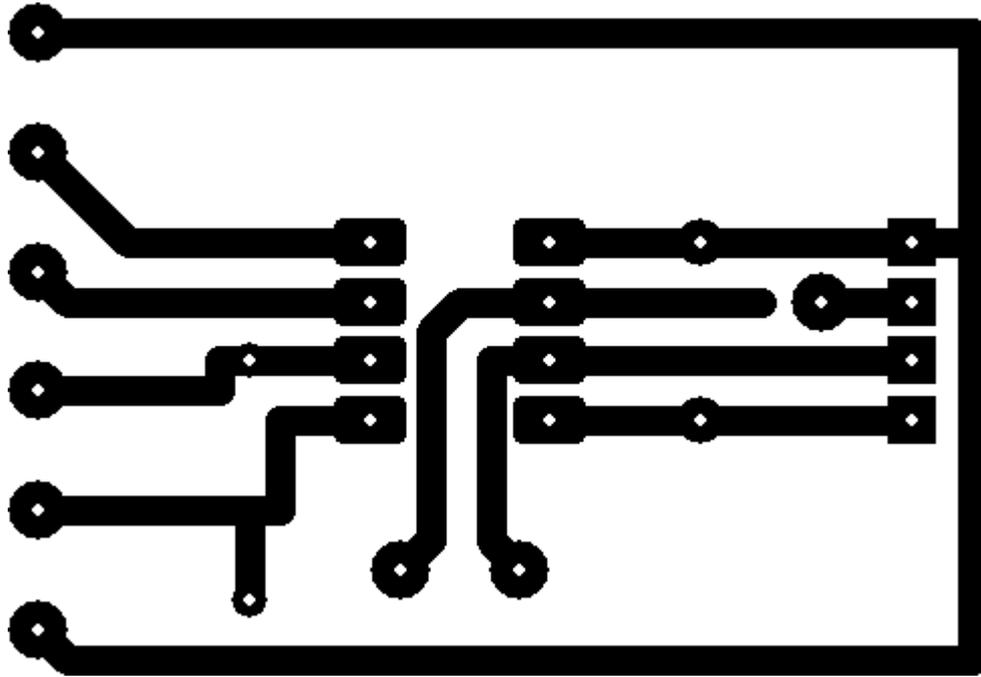
Schéma



RB1204 Carte DMX ver 1

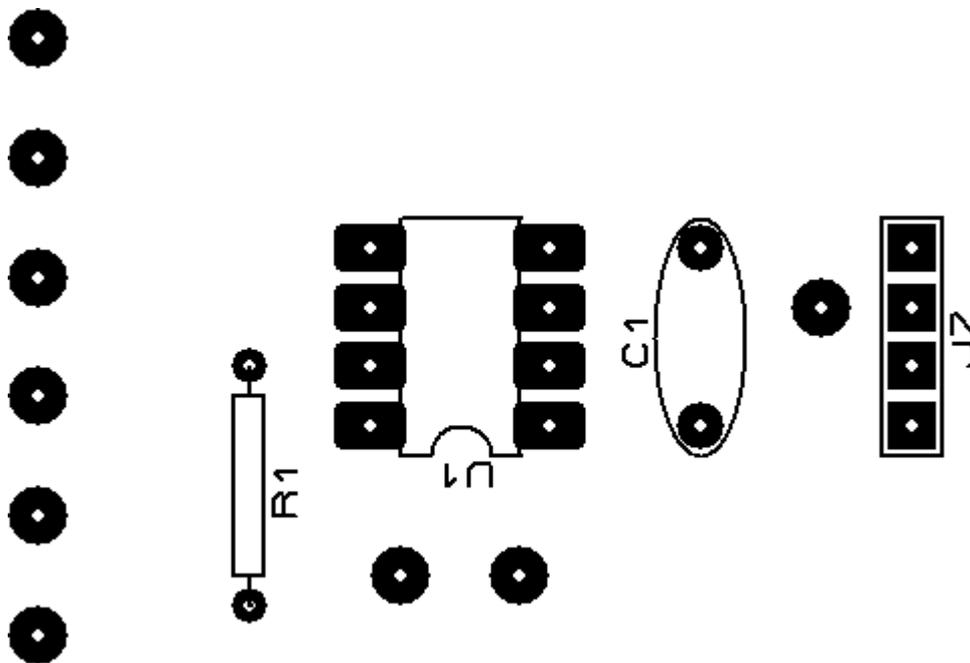
Bloc puissance DMX 12 voies

Typon



RB1204 Carte DMX ver 1

Implantation



RB1204 Carte DMX ver 1

Carte Extended

Schéma

RB1204 Carte Extended ver 1

Typon

RB1204 Carte Extended ver 1

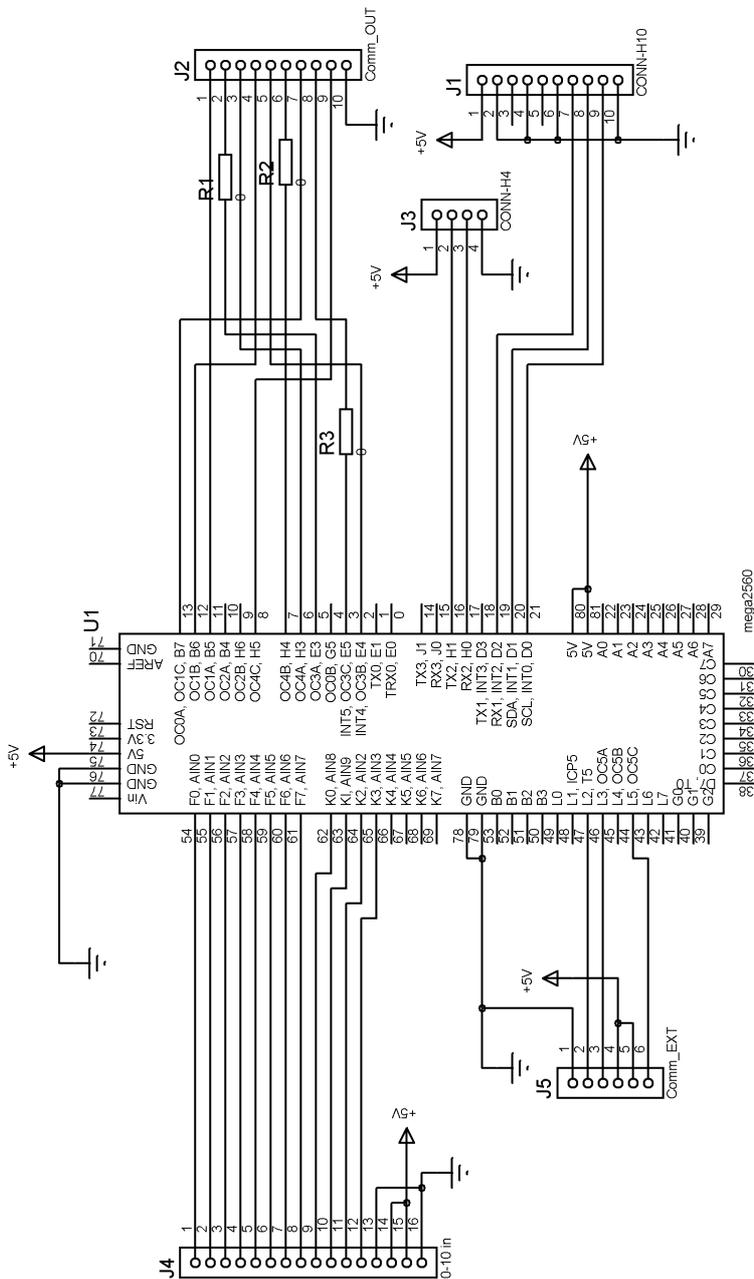
Implantation

RB1204 Carte Extended ver 1

Bloc puissance DMX 12 voies

Carte µP

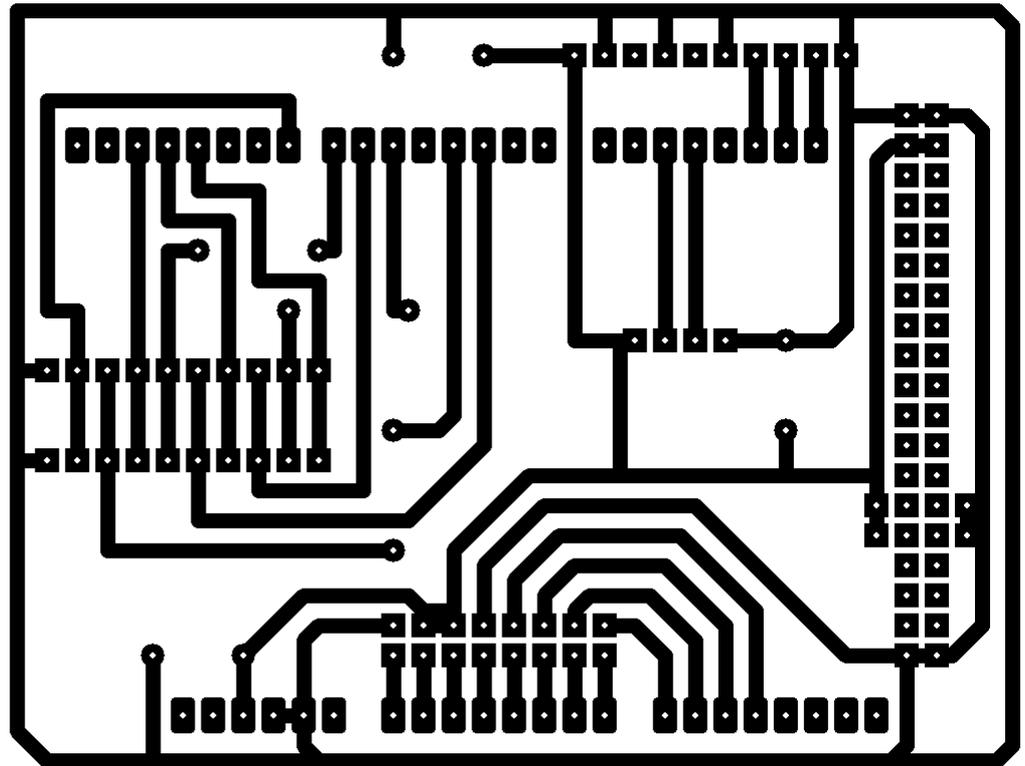
Schéma



RB1204 Carte µP ver 1

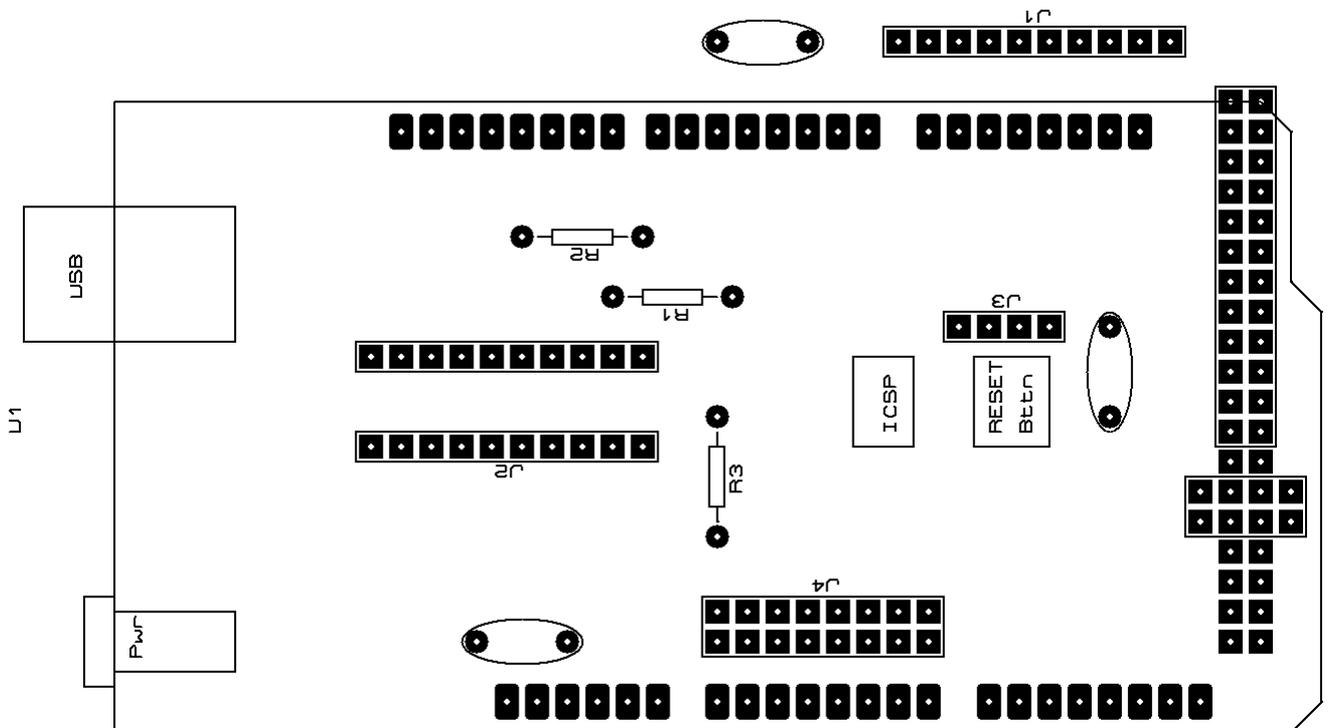
Bloc puissance DMX 12 voies

Typon



RB1204 Carte μ P ver 1

Implantation

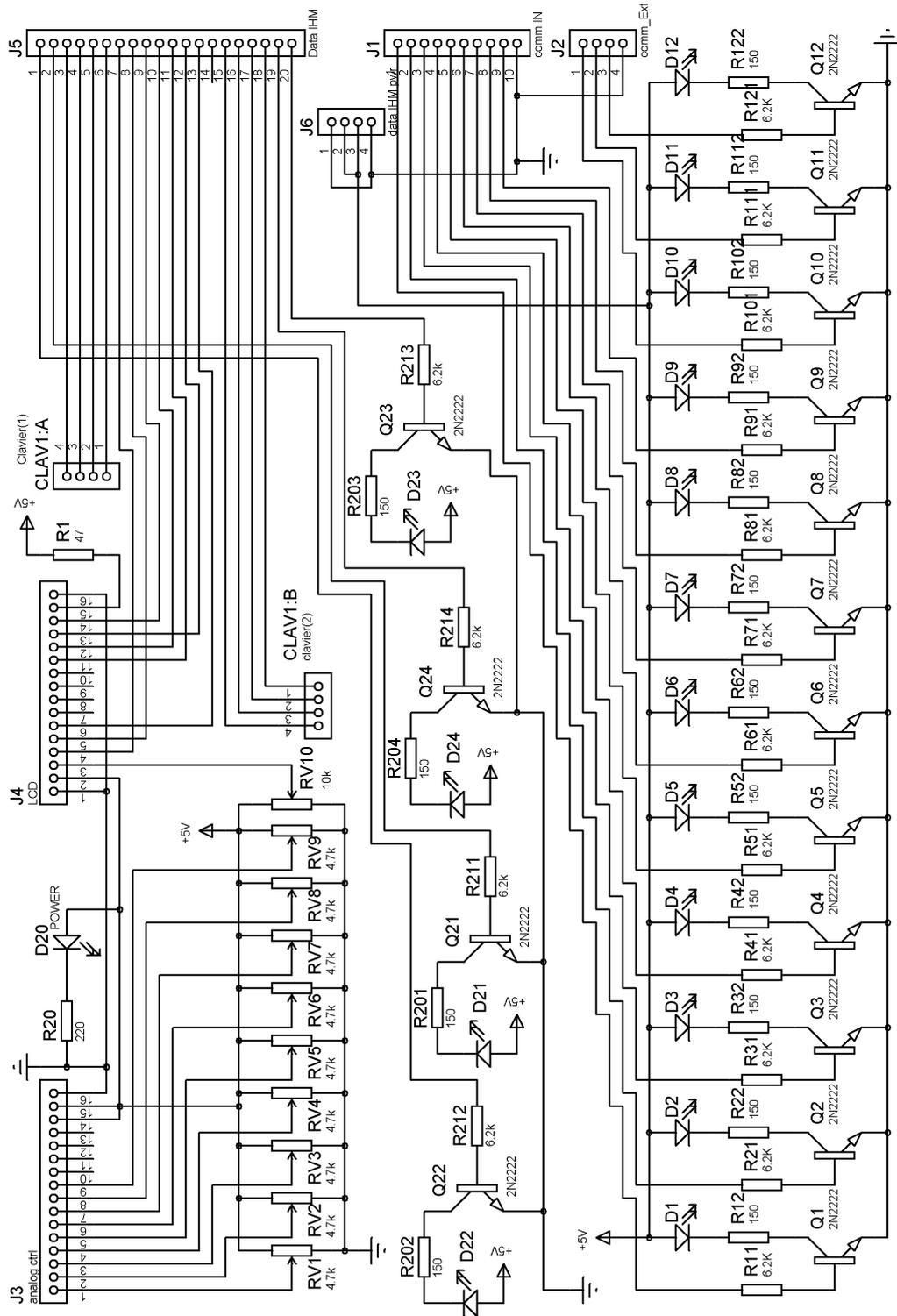


RB1204 Carte μ P ver 1

Bloc puissance DMX 12 voies

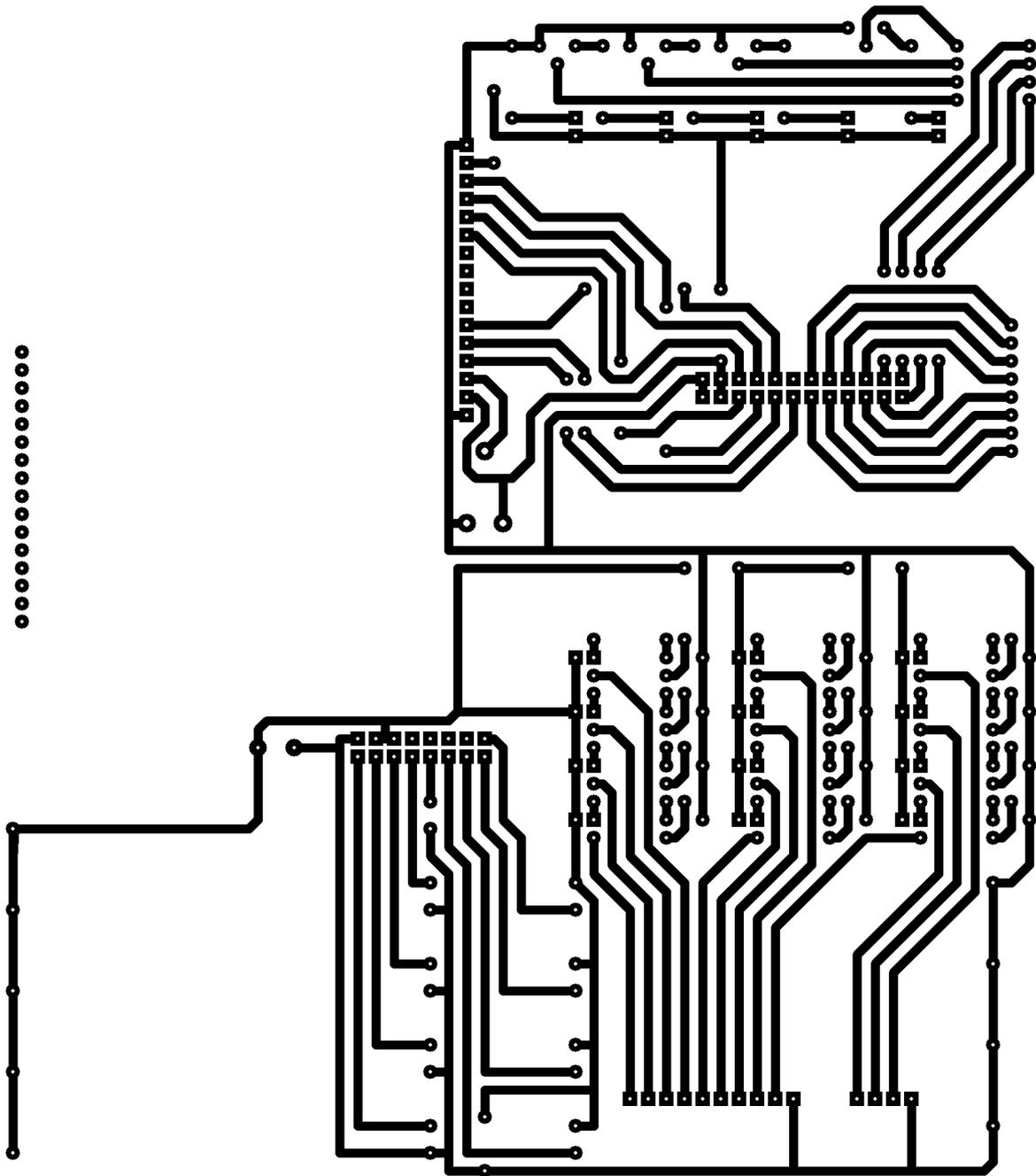
Carte Façade

Schéma



RB1204 Carte Façade ver 1

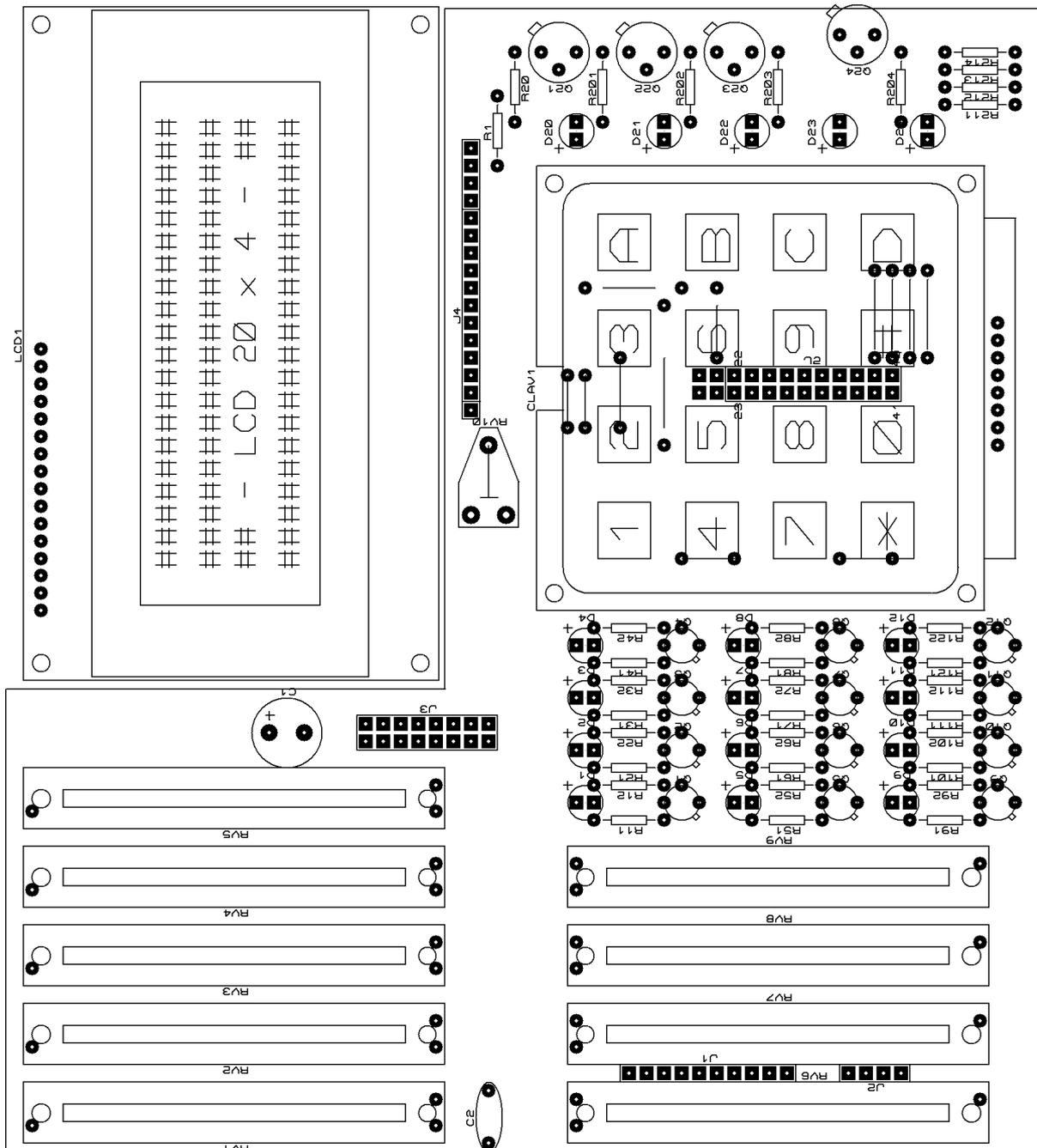
Typon



RB1204 Carte Façade ver 1

Bloc puissance DMX 12 voies

Implantation



RB1204 Carte Façade ver 1

Bloc puissance DMX 12 voies

Adresses des registres

| Address | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Page |
|---------|-----------------------|--|--------|--------|--------|-----------------------------------|---------|---------|---------|----------------------|
| 0x30 | (0x50) ACSR | ACD | ACBG | ACO | ACI | ACIE | ACIC | ACIS1 | ACIS0 | 272 |
| | (0x079) ADCH | ADC Data Register High byte | | | | | | | | 294 |
| | (0x078) ADCL | ADC Data Register Low byte | | | | | | | | 294 |
| | (0x07A) ADCSRA | ADEN | ADSC | ADATE | ADIF | ADIE | ADPS2 | ADPS1 | ADPS0 | 292 |
| | (0x07B) ADCSRB | - | ACME | - | - | MUX5 | ADTS2 | ADTS1 | ADTS0 | 272, |
| | (0x07C) ADMUX | REFS1 | REFS0 | ADLAR | MUX4 | MUX3 | MUX2 | MUX1 | MUX0 | 289 |
| | (0x0B6) ASSR | - | EXCLK | AS2 | TCN2UB | OCR2AUB | OCR2BUB | TCR2AUB | TCR2BUB | 184 |
| | (0x061) CLKPR | CLKPCE | - | - | - | CLKPS3 | CLKPS2 | CLKPS1 | CLKPS0 | 50 |
| 0x01 | (0x21) DDRA | DDA7 | DDA6 | DDA5 | DDA4 | DDA3 | DDA2 | DDA1 | DDA0 | 100 |
| 0x04 | (0x24) DDRB | DDB7 | DDB6 | DDB5 | DDB4 | DDB3 | DDB2 | DDB1 | DDB0 | 100 |
| 0x07 | (0x27) DDRC | DDC7 | DDC6 | DDC5 | DDC4 | DDC3 | DDC2 | DDC1 | DDC0 | 101 |
| 0x0A | (0x2A) DDRD | DDD7 | DDD6 | DDD5 | DDD4 | DDD3 | DDD2 | DDD1 | DDD0 | 101 |
| 0x0D | (0x2D) DDRE | DDE7 | DDE6 | DDE5 | DDE4 | DDE3 | DDE2 | DDE1 | DDE0 | 101 |
| 0x10 | (0x30) DDRF | DDF7 | DDF6 | DDF5 | DDF4 | DDF3 | DDF2 | DDF1 | DDF0 | 102 |
| 0x13 | (0x33) DDRG | - | - | DDG5 | DDG4 | DDG3 | DDG2 | DDG1 | DDG0 | 102 |
| | (0x101) DDRH | DDH7 | DDH6 | DDH5 | DDH4 | DDH3 | DDH2 | DDH1 | DDH0 | 103 |
| | (0x104) DDRJ | DDJ7 | DDJ6 | DDJ5 | DDJ4 | DDJ3 | DDJ2 | DDJ1 | DDJ0 | 103 |
| | (0x107) DDRK | DDK7 | DDK6 | DDK5 | DDK4 | DDK3 | DDK2 | DDK1 | DDK0 | 103 |
| | (0x10A) DDRL | DDL7 | DDL6 | DDL5 | DDL4 | DDL3 | DDL2 | DDL1 | DDL0 | 104 |
| | (0x07E) DIDR0 | ADC7D | ADC6D | ADC5D | ADC4D | ADC3D | ADC2D | ADC1D | ADC0D | 295 |
| | (0x07F) DIDR1 | - | - | - | - | - | - | AIN1D | AIN0D | 274 |
| | (0x07D) DIDR2 | ADC15D | ADC14D | ADC13D | ADC12D | ADC11D | ADC10D | ADC9D | ADC8D | 295 |
| 0x22 | (0x42) EEARH | - | - | - | - | EEPROM Address Register High Byte | | | | 35 |
| 0x21 | (0x41) EEARL | EEPROM Address Register Low Byte | | | | | | | | 35 |
| 0x1F | (0x3F) EECR | - | - | EEP0M1 | EEP0M0 | EERIE | EEMPE | EEPE | EERE | 35 |
| 0x20 | (0x40) EEDR | EEPROM Data Register | | | | | | | | 35 |
| | (0x069) EICRA | ISC31 | ISC30 | ISC21 | ISC20 | ISC11 | ISC10 | ISC01 | ISC00 | 113 |
| | (0x06A) EICRB | ISC71 | ISC70 | ISC61 | ISC60 | ISC51 | ISC50 | ISC41 | ISC40 | 114 |
| 0x1C | (0x3C) EIFR | INTF7 | INTF6 | INTF5 | INTF4 | INTF3 | INTF2 | INTF1 | INTF0 | 115 |
| 0x1D | (0x3D) EIMSK | INT7 | INT6 | INT5 | INT4 | INT3 | INT2 | INT1 | INT0 | 115 |
| 0x3C | (0x5C) EIND | - | - | - | - | - | - | - | EIND0 | 17 |
| 0x1E | (0x3E) GPIOR0 | General Purpose I/O Register 0 | | | | | | | | 37 |
| 0x2A | (0x4A) GPIOR1 | General Purpose I/O Register 1 | | | | | | | | 37 |
| 0x2B | (0x4B) GPIOR2 | General Purpose I/O Register 2 | | | | | | | | 37 |
| 0x23 | (0x43) GTCCR | TSM | - | - | - | - | - | PSRASY | PSRSYNC | 170, 194 |
| | (0x087) ICR1H | Timer/Counter1 – Input Capture Register High Byte | | | | | | | | 165 |
| | (0x086) ICR1L | Timer/Counter1 – Input Capture Register Low Byte | | | | | | | | 165 |
| | (0x097) ICR3H | Timer/Counter3 – Input Capture Register High Byte | | | | | | | | 165 |
| | (0x096) ICR3L | Timer/Counter3 – Input Capture Register Low Byte | | | | | | | | 165 |
| | (0x0A7) ICR4H | Timer/Counter4 – Input Capture Register High Byte | | | | | | | | 165 |
| | (0x0A6) ICR4L | Timer/Counter4 – Input Capture Register Low Byte | | | | | | | | 165 |
| | (0x127) ICR5H | Timer/Counter5 – Input Capture Register High Byte | | | | | | | | 165 |
| | (0x126) ICR5L | Timer/Counter5 – Input Capture Register Low Byte | | | | | | | | 165 |
| 0x35 | (0x55) MCUCR | JTD | - | - | PUD | - | - | IVSEL | IVCE | 67, 110, 100, 308 |
| 0x34 | (0x54) MCUSR | - | - | - | JTRF | WDRF | BORF | EXTRF | PORF | 308 |
| 0x31 | (0x51) OCDR | OCDR7 | OCDR6 | OCDR5 | OCDR4 | OCDR3 | OCDR2 | OCDR1 | OCDR0 | 301 |
| 0x27 | (0x47) OCR0A | Timer/Counter0 Output Compare Register A | | | | | | | | 133 |
| 0x28 | (0x48) OCR0B | Timer/Counter0 Output Compare Register B | | | | | | | | 133 |
| | (0x089) OCR1AH | Timer/Counter1 – Output Compare Register A High Byte | | | | | | | | 163 |
| | (0x088) OCR1AL | Timer/Counter1 – Output Compare Register A Low Byte | | | | | | | | 163 |
| | (0x08B) OCR1BH | Timer/Counter1 – Output Compare Register B High Byte | | | | | | | | 163 |
| | (0x08A) OCR1BL | Timer/Counter1 – Output Compare Register B Low Byte | | | | | | | | 163 |
| | (0x08D) OCR1CH | Timer/Counter1 – Output Compare Register C High Byte | | | | | | | | 163 |
| | (0x08C) OCR1CL | Timer/Counter1 – Output Compare Register C Low Byte | | | | | | | | 163 |
| | (0x0B3) OCR2A | Timer/Counter2 Output Compare Register A | | | | | | | | 191 |
| | (0x0B4) OCR2B | Timer/Counter2 Output Compare Register B | | | | | | | | 191 |
| | (0x099) OCR3AH | Timer/Counter3 – Output Compare Register A High Byte | | | | | | | | 163 |
| | (0x098) OCR3AL | Timer/Counter3 – Output Compare Register A Low Byte | | | | | | | | 163 |
| | (0x09B) OCR3BH | Timer/Counter3 – Output Compare Register B High Byte | | | | | | | | 164 |
| | (0x09A) OCR3BL | Timer/Counter3 – Output Compare Register B Low Byte | | | | | | | | 164 |
| | (0x09D) OCR3CH | Timer/Counter3 – Output Compare Register C High Byte | | | | | | | | 164 |
| | (0x09C) OCR3CL | Timer/Counter3 – Output Compare Register C Low Byte | | | | | | | | 164 |
| | (0x0A9) OCR4AH | Timer/Counter4 – Output Compare Register A High Byte | | | | | | | | 164 |
| | (0x0A8) OCR4AL | Timer/Counter4 – Output Compare Register A Low Byte | | | | | | | | 164 |
| | (0x0AB) OCR4BH | Timer/Counter4 – Output Compare Register B High Byte | | | | | | | | 164 |
| | (0x0AA) OCR4BL | Timer/Counter4 – Output Compare Register B Low Byte | | | | | | | | 164 |
| | (0x0AD) OCR4CH | Timer/Counter4 – Output Compare Register C High Byte | | | | | | | | 164 |
| | (0x0AC) OCR4CL | Timer/Counter4 – Output Compare Register C Low Byte | | | | | | | | 164 |
| | (0x129) OCR5AH | Timer/Counter5 – Output Compare Register A High Byte | | | | | | | | 164 |
| | (0x128) OCR5AL | Timer/Counter5 – Output Compare Register A Low Byte | | | | | | | | 164 |
| | (0x12B) OCR5BH | Timer/Counter5 – Output Compare Register B High Byte | | | | | | | | 165 |

Bloc puissance DMX 12 voies

| | | | | | | | | | | | | |
|------|---------|-----------------|--|---------|---------|---------|---------|----------|----------|----------|-----|-----|
| | (0x12A) | OCR5BL | Timer/Counter5 – Output Compare Register B Low Byte | | | | | | | | | 165 |
| | (0x12D) | OCR5CH | Timer/Counter5 – Output Compare Register C High Byte | | | | | | | | | 165 |
| | (0x12C) | OCR5CL | Timer/Counter5 – Output Compare Register C Low Byte | | | | | | | | | 165 |
| | (0x066) | OSCCAL | Oscillator Calibration Register | | | | | | | | | 50 |
| | (0x068) | PCICR | - | - | - | - | - | PCIE2 | PCIE1 | PCIE0 | 115 | |
| 0x1B | (0x3B) | PCIFR | - | - | - | - | - | PCIF2 | PCIF1 | PCIF0 | 116 | |
| | (0x06B) | PCMSK0 | PCINT7 | PCINT6 | PCINT5 | PCINT4 | PCINT3 | PCINT2 | PCINT1 | PCINT0 | 117 | |
| | (0x06C) | PCMSK1 | PCINT15 | PCINT14 | PCINT13 | PCINT12 | PCINT11 | PCINT10 | PCINT9 | PCINT8 | 116 | |
| | (0x06D) | PCMSK2 | PCINT23 | PCINT22 | PCINT21 | PCINT20 | PCINT19 | PCINT18 | PCINT17 | PCINT16 | 116 | |
| 0x00 | (0x20) | PINA | PINA7 | PINA6 | PINA5 | PINA4 | PINA3 | PINA2 | PINA1 | PINA0 | 100 | |
| 0x03 | (0x23) | PINB | PINB7 | PINB6 | PINB5 | PINB4 | PINB3 | PINB2 | PINB1 | PINB0 | 100 | |
| 0x06 | (0x26) | PINC | PINC7 | PINC6 | PINC5 | PINC4 | PINC3 | PINC2 | PINC1 | PINC0 | 101 | |
| 0x09 | (0x29) | PIND | PIND7 | PIND6 | PIND5 | PIND4 | PIND3 | PIND2 | PIND1 | PIND0 | 101 | |
| 0x0C | (0x2C) | PINE | PINE7 | PINE6 | PINE5 | PINE4 | PINE3 | PINE2 | PINE1 | PINE0 | 102 | |
| 0x0F | (0x2F) | PINF | PINF7 | PINF6 | PINF5 | PINF4 | PINF3 | PINF2 | PINF1 | PINF0 | 102 | |
| 0x12 | (0x32) | PING | - | - | PING5 | PING4 | PING3 | PING2 | PING1 | PING0 | 102 | |
| | (0x100) | PINH | PINH7 | PINH6 | PINH5 | PINH4 | PINH3 | PINH2 | PINH1 | PINH0 | 103 | |
| | (0x103) | PINJ | PINJ7 | PINJ6 | PINJ5 | PINJ4 | PINJ3 | PINJ2 | PINJ1 | PINJ0 | 103 | |
| | (0x106) | PINK | PINK7 | PINK6 | PINK5 | PINK4 | PINK3 | PINK2 | PINK1 | PINK0 | 103 | |
| | (0x109) | PINL | PINL7 | PINL6 | PINL5 | PINL4 | PINL3 | PINL2 | PINL1 | PINL0 | 104 | |
| 0x02 | (0x22) | PORTA | PORTA7 | PORTA6 | PORTA5 | PORTA4 | PORTA3 | PORTA2 | PORTA1 | PORTA0 | 100 | |
| 0x05 | (0x25) | PORTB | PORTB7 | PORTB6 | PORTB5 | PORTB4 | PORTB3 | PORTB2 | PORTB1 | PORTB0 | 100 | |
| 0x08 | (0x28) | PORTC | PORTC7 | PORTC6 | PORTC5 | PORTC4 | PORTC3 | PORTC2 | PORTC1 | PORTC0 | 101 | |
| 0x0B | (0x2B) | PORTD | PORTD7 | PORTD6 | PORTD5 | PORTD4 | PORTD3 | PORTD2 | PORTD1 | PORTD0 | 101 | |
| 0x0E | (0x2E) | ORTE | ORTE7 | ORTE6 | ORTE5 | ORTE4 | ORTE3 | ORTE2 | ORTE1 | ORTE0 | 101 | |
| 0x11 | (0x31) | PORTF | PORTF7 | PORTF6 | PORTF5 | PORTF4 | PORTF3 | PORTF2 | PORTF1 | PORTF0 | 101 | |
| 0x14 | (0x34) | PORTG | - | - | PORTG5 | PORTG4 | PORTG3 | PORTG2 | PORTG1 | PORTG0 | 102 | |
| | (0x102) | PORTH | PORTH7 | PORTH6 | PORTH5 | PORTH4 | PORTH3 | PORTH2 | PORTH1 | PORTH0 | 102 | |
| | (0x105) | PORTJ | PORTJ7 | PORTJ6 | PORTJ5 | PORTJ4 | PORTJ3 | PORTJ2 | PORTJ1 | PORTJ0 | 103 | |
| | (0x108) | PORTK | PORTK7 | PORTK6 | PORTK5 | PORTK4 | PORTK3 | PORTK2 | PORTK1 | PORTK0 | 103 | |
| | (0x10B) | PORTL | PORTL7 | PORTL6 | PORTL5 | PORTL4 | PORTL3 | PORTL2 | PORTL1 | PORTL0 | 104 | |
| | (0x064) | PRR0 | PRTWI | PRTIM2 | PRTIM0 | - | PRTIM1 | PRSPI | PRUSART0 | PRADC | 56 | |
| | (0x065) | PRR1 | - | - | PRTIM5 | PRTIM4 | PRTIM3 | PRUSART3 | PRUSART2 | PRUSART1 | 57 | |
| 0x3B | (0x5B) | RAMPZ | - | - | - | - | - | - | RAMPZ1 | RAMPZ0 | 17 | |
| 0x29 | (0x49) | Reserved | - | - | - | - | - | - | - | - | - | |
| 0x2F | (0x4F) | Reserved | - | - | - | - | - | - | - | - | - | |
| 0x32 | (0x52) | Reserved | - | - | - | - | - | - | - | - | - | |
| 0x36 | (0x56) | Reserved | - | - | - | - | - | - | - | - | - | |
| 0x38 | (0x58) | Reserved | - | - | - | - | - | - | - | - | - | |
| 0x39 | (0x59) | Reserved | - | - | - | - | - | - | - | - | - | |
| 0x3A | (0x5A) | Reserved | - | - | - | - | - | - | - | - | - | |
| 0x33 | (0x53) | SMCR | - | - | - | - | SM2 | SM1 | SM0 | SE | 52 | |
| 0x2C | (0x4C) | SPCR | SPIE | SPE | DORD | MSTR | CPOL | CPHA | SPR1 | SPR0 | 202 | |
| 0x2E | (0x4E) | SPDR | SPI Data Register | | | | | | | | | 204 |
| 0x3E | (0x5E) | SPH | SP15 | SP14 | SP13 | SP12 | SP11 | SP10 | SP9 | SP8 | 16 | |
| 0x3D | (0x5D) | SPL | SP7 | SP6 | SP5 | SP4 | SP3 | SP2 | SP1 | SP0 | 16 | |
| 0x37 | (0x57) | SPMCSR | SPMIE | RWWSB | SIGRD | RWWSRE | BLBSET | PGWRT | PGERS | SPMEN | 332 | |
| 0x2D | (0x4D) | SPSR | SPIF | WCOL | - | - | - | - | - | SPI2X | 203 | |
| 0x3F | (0x5F) | SREG | I | T | H | S | V | N | Z | C | 14 | |
| 0x24 | (0x44) | TCCR0A | COM0A1 | COM0A0 | COM0B1 | COM0B0 | - | - | WGM01 | WGM00 | 129 | |
| 0x25 | (0x45) | TCCR0B | FOC0A | FOC0B | - | - | WGM02 | CS02 | CS01 | CS00 | 132 | |
| | (0x080) | TCCR1A | COM1A1 | COM1A0 | COM1B1 | COM1B0 | COM1C1 | COM1C0 | WGM11 | WGM10 | 158 | |
| | (0x081) | TCCR1B | ICNC1 | ICES1 | - | WGM13 | WGM12 | CS12 | CS11 | CS10 | 160 | |
| | (0x082) | TCCR1C | FOC1A | FOC1B | FOC1C | - | - | - | - | - | 161 | |
| | (0x0B0) | TCCR2A | COM2A1 | COM2A0 | COM2B1 | COM2B0 | - | - | WGM21 | WGM20 | 191 | |
| | (0x0B1) | TCCR2B | FOC2A | FOC2B | - | - | WGM22 | CS22 | CS21 | CS20 | 190 | |
| | (0x090) | TCCR3A | COM3A1 | COM3A0 | COM3B1 | COM3B0 | COM3C1 | COM3C0 | WGM31 | WGM30 | 158 | |
| | (0x091) | TCCR3B | ICNC3 | ICES3 | - | WGM33 | WGM32 | CS32 | CS31 | CS30 | 160 | |
| | (0x092) | TCCR3C | FOC3A | FOC3B | FOC3C | - | - | - | - | - | 162 | |
| | (0x0A0) | TCCR4A | COM4A1 | COM4A0 | COM4B1 | COM4B0 | COM4C1 | COM4C0 | WGM41 | WGM40 | 158 | |
| | (0x0A1) | TCCR4B | ICNC4 | ICES4 | - | WGM43 | WGM42 | CS42 | CS41 | CS40 | 160 | |
| | (0x0A2) | TCCR4C | FOC4A | FOC4B | FOC4C | - | - | - | - | - | 162 | |
| | (0x120) | TCCR5A | COM5A1 | COM5A0 | COM5B1 | COM5B0 | COM5C1 | COM5C0 | WGM51 | WGM50 | 158 | |
| | (0x121) | TCCR5B | ICNC5 | ICES5 | - | WGM53 | WGM52 | CS52 | CS51 | CS50 | 160 | |
| | (0x122) | TCCR5C | FOC5A | FOC5B | FOC5C | - | - | - | - | - | 162 | |
| 0x26 | (0x46) | TCNT0 | Timer/Counter0 (8 Bit) | | | | | | | | | 133 |
| | (0x085) | TCNT1H | Timer/Counter1 – Counter Register High Byte | | | | | | | | | 162 |
| | (0x084) | TCNT1L | Timer/Counter1 – Counter Register Low Byte | | | | | | | | | 162 |
| | (0x0B2) | TCNT2 | Timer/Counter2 (8 Bit) | | | | | | | | | 191 |
| | (0x095) | TCNT3H | Timer/Counter3 – Counter Register High Byte | | | | | | | | | 162 |
| | (0x094) | TCNT3L | Timer/Counter3 – Counter Register Low Byte | | | | | | | | | 162 |
| | (0x0A5) | TCNT4H | Timer/Counter4 – Counter Register High Byte | | | | | | | | | 163 |
| | (0x0A4) | TCNT4L | Timer/Counter4 – Counter Register Low Byte | | | | | | | | | 163 |
| | (0x125) | TCNT5H | Timer/Counter5 – Counter Register High Byte | | | | | | | | | 163 |
| | (0x124) | TCNT5L | Timer/Counter5 – Counter Register Low Byte | | | | | | | | | 163 |
| 0x15 | (0x35) | TIFR0 | - | - | - | - | - | OCF0B | OCF0A | TOV0 | 134 | |

Bloc puissance DMX 12 voies

| | | | | | | | | | | | | |
|------|---------|---------------|---|---------|--------|-------|--------|--------|--------|--------|-----|-----|
| 0x16 | (0x36) | TIFR1 | - | - | ICF1 | - | OCF1C | OCF1B | OCF1A | TOV1 | 167 | |
| 0x17 | (0x37) | TIFR2 | - | - | - | - | - | OCF2B | OCF2A | TOV2 | 193 | |
| 0x18 | (0x38) | TIFR3 | - | - | ICF3 | - | OCF3C | OCF3B | OCF3A | TOV3 | 167 | |
| 0x19 | (0x39) | TIFR4 | - | - | ICF4 | - | OCF4C | OCF4B | OCF4A | TOV4 | 167 | |
| 0x1A | (0x3A) | TIFR5 | - | - | ICF5 | - | OCF5C | OCF5B | OCF5A | TOV5 | 166 | |
| | (0x06E) | TIMSK0 | - | - | - | - | - | OCIE0B | OCIE0A | TOIE0 | 134 | |
| | (0x06F) | TIMSK1 | - | - | ICIE1 | - | OCIE1C | OCIE1B | OCIE1A | TOIE1 | 166 | |
| | (0x070) | TIMSK2 | - | - | - | - | - | OCIE2B | OCIE2A | TOIE2 | 193 | |
| | (0x071) | TIMSK3 | - | - | ICIE3 | - | OCIE3C | OCIE3B | OCIE3A | TOIE3 | 166 | |
| | (0x072) | TIMSK4 | - | - | ICIE4 | - | OCIE4C | OCIE4B | OCIE4A | TOIE4 | 166 | |
| | (0x073) | TIMSK5 | - | - | ICIE5 | - | OCIE5C | OCIE5B | OCIE5A | TOIE5 | 166 | |
| | (0x0BD) | TWAMR | TWAM6 | TWAM5 | TWAM4 | TWAM3 | TWAM2 | TWAM1 | TWAM0 | - | 269 | |
| | (0x0BA) | TWAR | TWA6 | TWA5 | TWA4 | TWA3 | TWA2 | TWA1 | TWA0 | TWGCE | 269 | |
| | (0x0B8) | TWBR | 2-wire Serial Interface Bit Rate Register | | | | | | | | | 266 |
| | (0x0BC) | TWCR | TWINT | TWEA | TWSTA | TWSTO | TWWC | TWEN | - | TWIE | 266 | |
| | (0x0BB) | TWDR | 2-wire Serial Interface Data Register | | | | | | | | | 268 |
| | (0x0B9) | TWSR | TWS7 | TWS6 | TWS5 | TWS4 | TWS3 | - | TWPS1 | TWPS0 | 268 | |
| | (0x0C5) | UBRR0H | USART0 Baud Rate Register High Byte | | | | | | | | | 227 |
| | (0x0C4) | UBRR0L | USART0 Baud Rate Register Low Byte | | | | | | | | | 227 |
| | (0x0CD) | UBRR1H | USART1 Baud Rate Register High Byte | | | | | | | | | 227 |
| | (0x0CC) | UBRR1L | USART1 Baud Rate Register Low Byte | | | | | | | | | 227 |
| | (0x0D5) | UBRR2H | USART2 Baud Rate Register High Byte | | | | | | | | | 227 |
| | (0x0D4) | UBRR2L | USART2 Baud Rate Register Low Byte | | | | | | | | | 227 |
| | (0x135) | UBRR3H | USART3 Baud Rate Register High Byte | | | | | | | | | 227 |
| | (0x134) | UBRR3L | USART3 Baud Rate Register Low Byte | | | | | | | | | 227 |
| | (0x0C0) | UCSR0A | RXC0 | TXC0 | UDRE0 | FE0 | DOR0 | UPE0 | U2X0 | MPCM0 | 238 | |
| | (0x0C1) | UCSR0B | RXCIE0 | TXCIE0 | UDRIE0 | RXEN0 | TXEN0 | UCSZ02 | RXB80 | TXB80 | 238 | |
| | (0x0C2) | UCSR0C | UMSEL01 | UMSEL00 | UPM01 | UPM00 | USBS0 | UCSZ01 | UCSZ00 | UCPOL0 | 239 | |
| | (0x0C8) | UCSR1A | RXC1 | TXC1 | UDRE1 | FE1 | DOR1 | UPE1 | U2X1 | MPCM1 | 238 | |
| | (0x0C9) | UCSR1B | RXCIE1 | TXCIE1 | UDRIE1 | RXEN1 | TXEN1 | UCSZ12 | RXB81 | TXB81 | 238 | |
| | (0x0CA) | UCSR1C | UMSEL11 | UMSEL10 | UPM11 | UPM10 | USBS1 | UCSZ11 | UCSZ10 | UCPOL1 | 239 | |
| | (0x0D0) | UCSR2A | RXC2 | TXC2 | UDRE2 | FE2 | DOR2 | UPE2 | U2X2 | MPCM2 | 238 | |
| | (0x0D1) | UCSR2B | RXCIE2 | TXCIE2 | UDRIE2 | RXEN2 | TXEN2 | UCSZ22 | RXB82 | TXB82 | 238 | |
| | (0x0D2) | UCSR2C | UMSEL21 | UMSEL20 | UPM21 | UPM20 | USBS2 | UCSZ21 | UCSZ20 | UCPOL2 | 239 | |
| | (0x130) | UCSR3A | RXC3 | TXC3 | UDRE3 | FE3 | DOR3 | UPE3 | U2X3 | MPCM3 | 238 | |
| | (0x131) | UCSR3B | RXCIE3 | TXCIE3 | UDRIE3 | RXEN3 | TXEN3 | UCSZ32 | RXB83 | TXB83 | 238 | |
| | (0x132) | UCSR3C | UMSEL31 | UMSEL30 | UPM31 | UPM30 | USBS3 | UCSZ31 | UCSZ30 | UCPOL3 | 239 | |
| | (0x0C6) | UDR0 | USART0 I/O Data Register | | | | | | | | | 222 |
| | (0x0CE) | UDR1 | USART1 I/O Data Register | | | | | | | | | 222 |
| | (0x0D6) | UDR2 | USART2 I/O Data Register | | | | | | | | | 222 |
| | (0x136) | UDR3 | USART3 I/O Data Register | | | | | | | | | 222 |
| | (0x060) | WDTCR | WDIF | WDIE | WDP3 | WDCE | WDE | WDP2 | WDP1 | WDP0 | 67 | |
| | (0x074) | XMCRA | SRE | SRL2 | SRL1 | SRL0 | SRW11 | SRW10 | SRW01 | SRW00 | 37 | |
| | (0x075) | XMCRB | XMBK | - | - | - | - | XMM2 | XMM1 | XMM0 | 38 | |

- Notes:** 1. For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.
2. I/O registers within the address range \$00 - \$1F are directly bit-accessible using the SBI and CBI instructions. In these registers, the value of single bits can be checked by using the SBIS and SBIC instructions.
3. Some of the status flags are cleared by writing a logical one to them. Note that the CBI and SBI instructions will operate on all bits in the I/O register, writing a one back into any flag read as set, thus clearing the flag. The CBI and SBI instructions work with registers 0x00 to 0x1F only.
4. When using the I/O specific commands IN and OUT, the I/O addresses \$00 - \$3F must be used. When addressing I/O registers as data space using LD and ST instructions, \$20 must be added to these addresses. The ATmega640/1280/1281/2560/2561 is a complex microcontroller with more peripheral units than can be supported within the 64 location reserved in Opcode for the IN and OUT instructions. For the Extended I/O space from \$60 - \$1FF in SRAM, only the ST/STS/STD and LD/LDS/LDD instructions can be used.