

Mots clefs (Microchip, PIC24F16KA101, TVA, DA-TV, radio-amateur, F1CJN, générateur vidéo, mire)

Générateur vidéo PAL PICDREAM II

Ce document décrit le générateur vidéo PCDREAM II. Il est composé d'un micro contrôleur PIC24F16KA101 de Microchip et d'un codeur PAL AD724 de Analog Devices.

Le générateur vidéo :

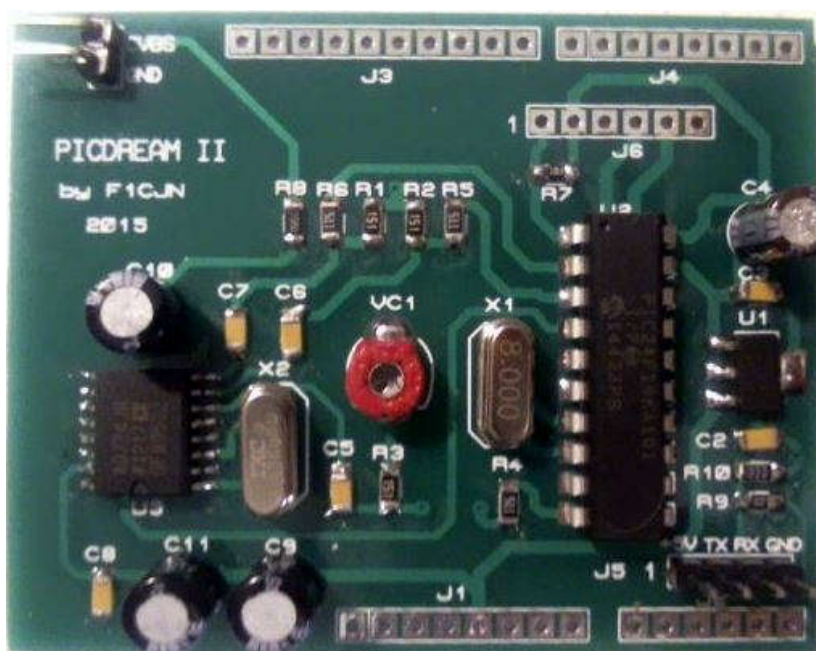
- génère un signal vidéo PAL entrelacé en 625 lignes,
- affiche un texte fixe ou déroulant en haut de l'image,
- affiche un texte fixe ou déroulant en bas de l'image,
- générer une pseudo mire de barre ou SMPE sélectionnable,
- les couleurs des fonds de texte et des textes peuvent être sélectionnées indépendamment.

Le générateur se programme à partir d'un signal série à 38400 bauds, 8N1, sans handshake soit à partir :

- d'une interface RS232 TTL (quelques euros sur EBay), avec alimentation 5V extérieure.
- d'une interface USB TTL (quelques euros sur EBay), qui alimente le montage en 5V.
- d'un Arduino car le circuit imprimé est compatible de l'Arduino Uno, ce qui permet par exemple sur un relais TVA de programmer plusieurs messages différents en temps réel. L'Arduino fournit alors la tension d'alimentation et les broches A0 et A1 doivent être utilisées pour piloter le PICDREAM II,

L'ensemble de la programmation des textes et du hardware peut être mémorisé dans l'EEPROM du PIC par une simple commande.

Version Prototype sans R9 et R10



1 Matériel :

1.1 liste des composants

Qté Référence Valeur

Résistances

3	R1-R3	150
3	R4-R6	510
1	R7	10k
1	R8	75
1	R9	1k
1	R10	2,2k

Condensateurs

1	C4	10uF à 22uF
6	C2,C3,C5-C8	100nF
3	C9-C11	220uF
1	VC1	condensateur ajustable 4.2-20pF

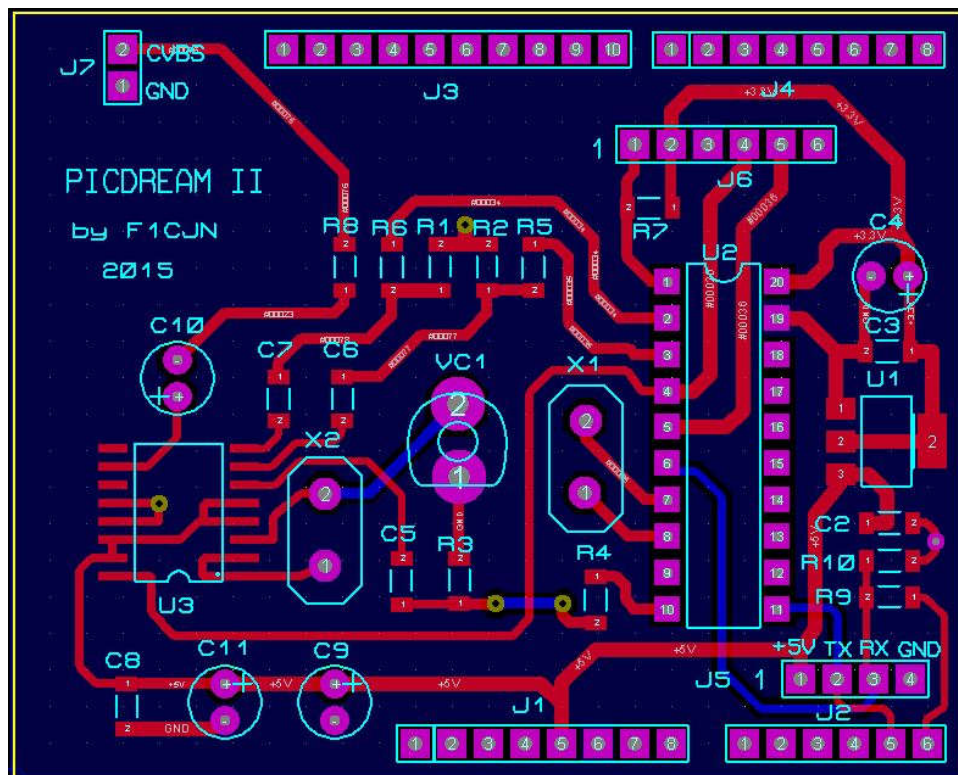
Circuits Intégrés

1	U1	LD1117-3.3
1	U2	PIC24F16KA101
1	U3	AD724 (à câbler avant C10 et X2 pour laisser de la place au fer à souder)

Divers

2	J1,J4	CONN-H8
2	J2,J6	CONN-H6
1	J3	CONN-H10
1	J5	CONN-H4
1	J7	CONN-H2
1	X1	8MHz
1	X2	4.43MHz

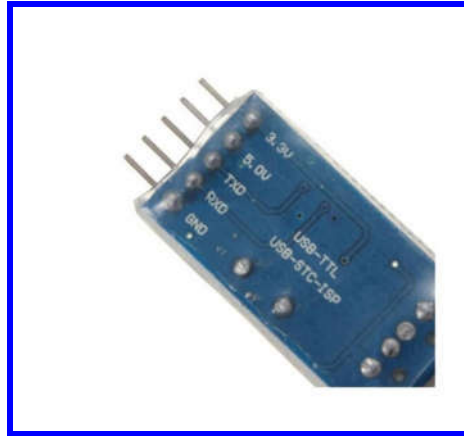
Placement des composants : vue de dessus



2 Utilisation

Exemple avec interface USB-TTL

Ces interfaces se négocient entre 1 et 2 euros sur Ebay.



Branchement du câble USB-TTL :

TX (vert) connecté à RX data de J1
RX (blanc) connecté à TX data de J1
+5V (rouge) connecté à +5V de J1
GND (noir) connecté à GND de J1

* Attention certains interfaces USB-TTL ont les noms des entrées RX et TX inversés. Dans ce cas connecter RX à RX et TX à TX.

Je conseille l'utilisation du programme de terminal "Termite", programme qui mémorise les commandes envoyées, ce qui permet de les rappeler rapidement lors des tests. Ce programme est compatible de tous les interfaces de type série.

Termite doit être programmé en **38400** bps, 8N1, no handshake

* Attention, si vous déconnectez le connecteur USB du PICDREAM II, puis rebranchez le PICDREAM II, il est nécessaire d'arrêter puis de relancer Termite, sinon il ne reconnecte pas le port USB.

3 Les commandes

Les commandes, rédigées en minuscules, doivent être encadrées par des crochets <>

Avant la première commande, envoyer un CR (bug premier caractère UART perdu »

3.1 Couleurs

<t1>montexte	= "montexte" s'affiche sur la ligne 1
<t2>montexte2	= "montexte2" s'affiche sur la ligne 2
<c1>xx	= couleur du texte de la ligne 1
<c2>xx	= couleur du texte de la ligne 2
<b1>xx	= couleur de fond de la ligne 1
<b2>xx	= couleur de fond de la ligne 2

Avec xx : ma = magenta, wh = white, ye = yellow, bk = black, cy = cyan, re = red, gr = green, et bl = blue

Exemple : <c1>cy couleur du texte de la ligne 1 = cyan

3.2 Texte en scrolling (limité à 72 caractères)

Il est nécessaire de déclarer le mode scrolling avant de rentrer le texte à scroller car le programme ajoute des espaces en tête et fin de message, afin de scroller proprement (sauf si on est déjà en mode scrolling)

<s1>	= scrolling ligne 1
<s2>	= scrolling ligne 2

3.2 Texte fixe

Il est nécessaire de déclarer le mode fixe avant de rentrer le texte (sauf si on est déjà en mode texte fixe).

<f1>	= ligne 1 fixe
<f2>	= ligne 2 fixe

3.3 Mires

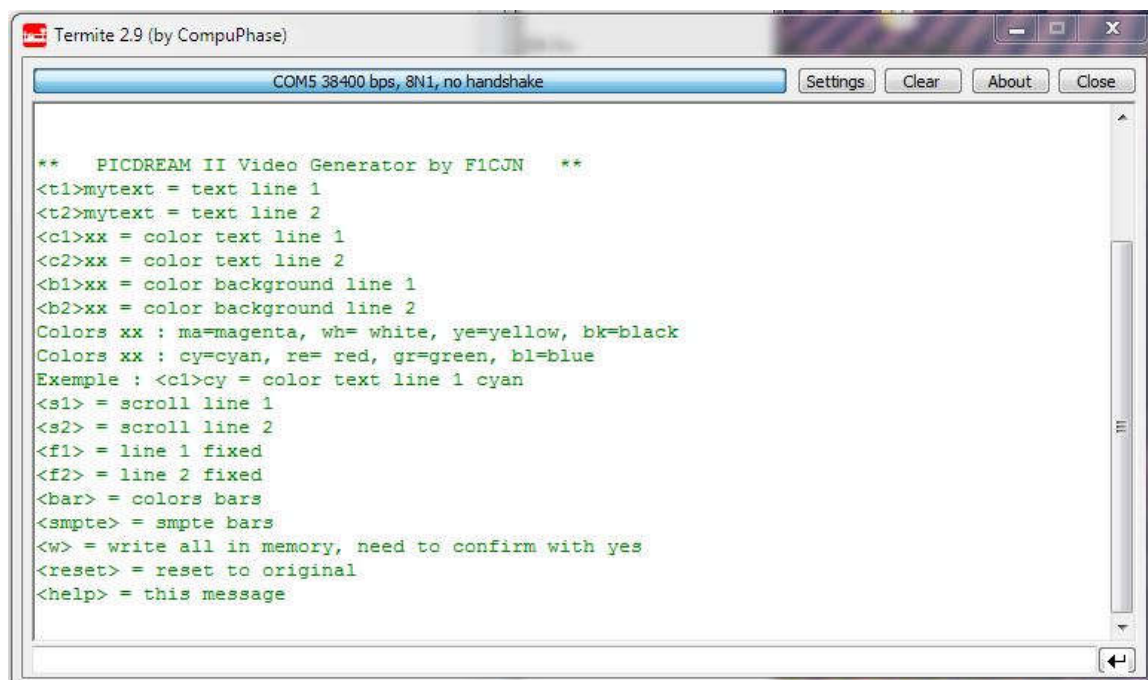
<bar>	= barres de couleurs
<smpte>	= barres SMPTE

3.4 Mémorisation en E2PROM

<w>	= mémorise la configuration, doit être confirmé avec le mot "yes". Ensuite, le PICDREAM II répond par ***WRITE OK ***.
-----	---

3.5 Divers

<reset>	= reset à l'origine, efface la personnalisation en E2PROM.
<help>	= résumé des commandes, voir l'image suivante



4 Utilisation avec Arduino

Enficher directement la carte sur un Arduino Uno. L'Arduino alimente alors le PICDREAM en 5V.

Nota : la sortie TX de l'Arduino Uno étant en 0-5Volt, un pont diviseur de tension réalisé par R9 et R10 permet de réduire la tension en 0-3,3 Volt environ sur l'entrée RXD du PIC.

Chaque commande envoyée au PICDREAM II doit être espacée de la suivante par une instruction " delay 50 " (ou délai supérieur) de façon à laisser le temps au PIC de réaliser la commande.

Pendant l'envoi de la commande, il se produit un parasite sur l'écran du à aux interruptions prioritaires sur l'entrée RXD.

Exemple de programme de test :

```
#include <SoftwareSerial.h>
SoftwareSerial mySerial(18,19); // A4,A5 analogique ~ RXpin,TXpin
// RXpin =18 et TXpin=19 sur Arduino

void setup() {
  mySerial.begin(38400); // liaison entre Arduino et PICDREAM II à 38400 bauds
  Serial.begin (115200); // liaison entre Arduino et PC
}
mySerial.write("<t2> \r"); // CR suite bug UART

void loop(){
  mySerial.write("<t2> F1CJN\r"); // Texte ligne 2
  delay(2000);
  mySerial.write("<c2>re\r"); // Texte ligne 2 en rouge
  delay(2000);
  mySerial.write("<t2>PICDREAM II \r") ;// Texte ligne 2
  delay(2000);
  mySerial.write("<c2>ye\r"); // Texte ligne 2 en jaune
  delay (50);
  mySerial.write("<bar>\r"); // mire de barres
  delay (50);
  mySerial.write("<b2>bk\r");// Fond texte ligne 2 rouge
  delay(2000);
  mySerial.write("<t2> DA-TV\r"); // Texte ligne 2
  delay(2000);
  mySerial.write("<c2>wh\r"); // Texte ligne 2 en blanc
  delay(2000);
  mySerial.write("<smpte>\r"); // Mire genre smpte
  delay(2000);
  mySerial.write("<b2>gr\r"); // Fond texte ligne 2vert
  delay (50);
}
```

5 Schéma électrique de la carte PICDREAM II :

