

# Mémoire de projet personnel d'électronique

<https://mieboc.codeberg.page>  
<mieboc@proton.me>

mars 2016  
révision avril 2019

# Résumé

Le Minitel est un standard de terminal-modem informatique mis à disposition des particuliers en France par l'État entre 1980 et 2012. L'architecture de certains modèles constitue la base d'un micro-ordinateur et a été conçue de façon à permettre une évolution technique par l'ajout d'une carte fille. À notre connaissance, aucun produit tirant parti de cette possibilité n'a jamais été commercialisé.

Ce projet consiste en la conception et la fabrication d'une telle carte fille, avec comme principales fonctions l'extension de la mémoire et la connexion de divers périphériques dont un crayon optique (dispositif de pointage).

À cette fin, nous avons réalisé le schéma électrique d'un Minitel puis conçu une carte fille compatible et répondant à nos critères. Le produit final fonctionne sous tous ses aspects.

# Abstract

The Minitel is a computer modem and terminal set provided by the state to French individuals from 1980 to 2012. The architecture of some models serves as the base for a microcomputer and was designed with the insertion of a daughterboard in mind, in order to enhance the capabilities of the device. To our knowledge, no product benefitting from it has ever been commercialized.

This project involves the designing and manufacturing of such a daughterboard, the main functions of which are memory expansion and support for various peripherals including a light pen (a kind of pointing device).

To this end, we drew the electrical schematics of a Minitel, and then designed a compatible daughterboard answering our objectives. The final product is functional under every aspect.

# Table des matières

Résumé	i
Abstract	i
Table des figures	iii
Liste des abréviations	iv
<b>1 Introduction</b>	<b>1</b>
<b>2 Résumé du travail accompli</b>	<b>3</b>
<b>3 Architecture du Minitel</b>	<b>5</b>
<b>4 Conception</b>	<b>8</b>
4.1 Caractéristiques techniques . . . . .	8
4.2 Considérations préliminaires . . . . .	8
4.2.1 Justification de l'emploi d'un CPLD . . . . .	8
4.2.2 Modifications sur l'appareil d'origine . . . . .	9
4.2.3 Principe général de l'interfaçage avec les périphériques . . . . .	9
4.2.4 Aspects mécaniques . . . . .	9
4.3 CPLD . . . . .	10
4.4 Diviseur d'horloge . . . . .	10
4.5 <i>Power-on reset</i> . . . . .	10
4.6 Traduction des niveaux de tension . . . . .	11
4.7 SRAM . . . . .	12
4.8 EEPROM . . . . .	12
4.9 Programmation de l'EEPROM par JTAG . . . . .	13
4.10 8052 et architecture de Von Neumann . . . . .	14
4.11 Commutation des bancs mémoire . . . . .	15
4.12 Bus SPI . . . . .	16
4.13 Carte SD . . . . .	17
4.14 Audio . . . . .	17
4.15 Séparateur de synchronisation . . . . .	18
4.16 Crayon optique . . . . .	21
4.16.1 Fonctionnement d'un crayon optique . . . . .	21
4.16.2 Test préliminaire . . . . .	22
4.16.3 Observations . . . . .	23

4.16.4	Circuit du crayon optique sur la carte M5 . . . . .	24
4.16.5	Module crayon optique . . . . .	24
4.17	Ports PS/2 . . . . .	24
4.17.1	Interfaçage avec le microcontrôleur . . . . .	25
4.17.2	Double port PS/2 . . . . .	25
4.18	Contrôleur d'interruption . . . . .	26
4.19	Connecteurs d'extension . . . . .	26
4.19.1	Connecteur SPI . . . . .	26
4.19.2	Connecteur audio-vidéo . . . . .	28
4.20	Péritel . . . . .	29
<b>5</b>	<b>Écueils</b>	<b>31</b>
5.1	Méthodologie . . . . .	31
5.2	Mécanique . . . . .	31
5.3	Freins thermiques . . . . .	32
5.4	Bruit sur la sortie audio . . . . .	32
5.5	Précision et sensibilité du crayon optique . . . . .	33
5.6	Plan de masse . . . . .	33
<b>6</b>	<b>Conclusion</b>	<b>35</b>
	<b>Annexes</b>	<b>37</b>
<b>A</b>	<b>Schéma électrique de la carte UT du Minitel 1B Téléc-Alcatel</b>	<b>38</b>
<b>B</b>	<b>Photographie du dessus de la carte UT</b>	<b>47</b>
<b>C</b>	<b>Schéma électrique de la carte M5</b>	<b>48</b>
<b>D</b>	<b>Aperçu fourni par le fabricant des circuits imprimés</b>	<b>55</b>
<b>E</b>	<b>Exemple de fichier SVF programmant l'EEPROM</b>	<b>58</b>
<b>F</b>	<b>Code source du module crayon optique</b>	<b>60</b>
<b>G</b>	<b>Photographies</b>	<b>62</b>



# Table des figures

3.1	Architecture matérielle simplifiée du Minitel 1B Téléc-Alcatel . . . . .	7
4.1	Flux de travaux de la programmation <i>in situ</i> de l'EEPROM . . . . .	15
4.2	Partitionnement de l'espace d'adressage . . . . .	16
4.3	Automate du séparateur de synchronisation . . . . .	19
4.4	Chronogramme du signal de synchronisation du VGP . . . . .	20
4.5	Brochage du connecteur pour crayon optique Thomson . . . . .	21
4.6	Photographie du crayon optique utilisé . . . . .	21
4.7	Schéma simplifié du circuit de visualisation pour crayon optique . . . . .	22
4.8	Image produite par le circuit de visualisation . . . . .	23
4.9	Brochage du connecteur PS/2 vu de l'ordinateur . . . . .	25
4.10	Schéma logique du contrôleur d'interruption . . . . .	27
4.11	Brochage du connecteur d'extension SPI . . . . .	28
4.12	Routage des connecteurs d'extension SPI pour le chaînage de périphériques . . . . .	28
4.13	Brochage du connecteur d'extension audio-vidéo . . . . .	29
5.1	Exemple de pastilles avec et sans frein thermique . . . . .	32

# Liste des abréviations

<b>ACT</b>	Advanced CMOS, TTL compatible
<b>AD</b>	Address-Data
<b>ALE</b>	Address Latch Enable
<b>ALVC</b>	Advanced Low-Voltage CMOS
<b>AOP</b>	Amplificateur Opérationnel
<b>AWG</b>	American Wire Gauge
<b>BSC</b>	Boundary-Scan Testing
<b>CAO</b>	Conception Assistée par Ordinateur
<b>CMOS</b>	Complementary Metal-Oxide-Semiconductor
<b>CPLD</b>	Complex Programmable Logic Device
<b>CS</b>	Chip Select
<b>CSYNC</b>	Composite Synchronization
<b>DAC</b>	Digital-to-Analog Converter
<b>DIN</b>	Deutsches Institut für Normung
<b>DIP</b>	Dual In-line Package
<b>EEPROM</b>	Electrically Erasable Programmable Read-Only Memory
<b>ESR</b>	Equivalent Series Resistance
<b>FAT</b>	File Allocation Table
<b>GND</b>	Ground
<b>HCT</b>	High-speed CMOS, TTL compatible
<b>HSYNC</b>	Horizontal Synchronization
<b>IDC</b>	Insulation-displacement Contact
<b>INT</b>	Interrupt
<b>JTAG</b>	Joint Test Action Group
<b>LVC MOS</b>	Low-Voltage CMOS
<b>M5</b>	Module avec Mémoire pour Modifier un Minitel en Micro-ordinateur
<b>MOSFET</b>	Metal-Oxide-Semiconductor Field-Effect Transistor
<b>PC</b>	Personal Computer
<b>PS/2</b>	Personal System/2
<b>PSEN</b>	Program Store Enable

<b>PTT</b>	Postes, Télégraphes et Téléphones
<b>RAM</b>	Random-Access Memory
<b>RC</b>	Résistance-Condensateur
<b>RD</b>	Read
<b>RGB</b>	Red, Green, Blue
<b>ROM</b>	Read-Only Memory
<b>SD</b>	Secure Digital
<b>SPI</b>	Serial Peripheral Interface
<b>SRAM</b>	Static Random-Access Memory
<b>STAPL</b>	Standard Test And Programming Language
<b>SVF</b>	Serial Vector Format
<b>TQFP</b>	Thin Quad Flat Package
<b>TTL</b>	Transistor-Transistor Logic
<b>VGP</b>	Video Graphics Processor
<b>VHCT</b>	Very High-speed CMOS, TTL compatible
<b>VSYNC</b>	Vertical Synchronization
<b>WR</b>	Write
<b>XOR</b>	Exclusive OR

# Chapitre 1

## Introduction

Par ce projet personnel d'électronique, j'ai souhaité mettre en pratique mes connaissances en électronique et en informatique.

L'idée de ce projet prend sa source vers la fin de l'année 2014 lorsque, en guise de passe-temps et par curiosité, je démontais un Minitel 1B Téléc-Alcatel pour en étudier le fonctionnement. Je remarquai que les concepteurs de l'appareil avaient doté celui-ci d'un connecteur d'extension, dont la finalité était de toute évidence l'ajout d'une carte fille et l'augmentation des fonctionnalités de l'appareil. Il me vint donc naturellement l'idée de construire une telle carte fille dans le but de transformer le Minitel d'origine, dont les fonctions sont figées et qui n'offre aucune possibilité de programmation, en un véritable micro-ordinateur programmable.

Après quelques recherches sur internet, il apparut que beaucoup de personnes avaient effectué la transformation du même appareil en ordinateur, mais par une méthode toute autre : la carte vidéo du Minitel est conservée, la carte mère d'origine retirée et remplacée par un ordinateur miniature moderne (le très répandu *Raspberry Pi*<sup>1</sup> par exemple), et un circuit d'interfaçage intercalé entre les deux. Si cela est sans aucun doute la façon la plus rapide, la plus simple et de loin la moins chère d'effectuer cette transformation, cela m'a paru peu intéressant d'un point de vue électronique, d'un point de vue fonctionnel également (pourquoi brider un ordinateur moderne par un écran cathodique noir et blanc et un clavier si peu ergonomique?), et cela ne respectait pas mon souhait de conserver la totalité de l'électronique d'origine de l'appareil.

Je me suis donc orienté vers la fabrication d'une carte fille augmentant les capacités de la carte mère, sans remplacer celle-ci.

Par manque de temps et parce que mes priorités d'alors étaient autres, je ne me mis à travailler sérieusement sur ce projet qu'en juillet 2015.

Le nom du code du projet est M5 (Module avec Mémoire pour Modifier un Minitel en Micro-ordinateur), et nous désignerons par *carte M5* la carte fille que nous avons réalisée.

Ce document présente les aspects techniques nécessaires à l'élaboration de ce projet, aspects que nous avons détaillé quand cela nous a paru essentiel ou digne d'intérêt. Pour ne pas imposer au lecteur un texte trop pénible, nous n'avons pas poussé le niveau de détail à son maximum.

---

1. <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>

Nous commençons par un résumé très synthétique du travail qui a été accompli dans le cadre de ce projet. Puis, afin que le lecteur puisse appréhender le contexte électronique dans lequel celui-ci s'insère, nous présentons l'architecture générale du Minitel. Dans le chapitre qui suit, nous mentionnons les différents sous-systèmes du projet et détaillons plus ou moins leur conception. Lorsque des alternatives techniques se sont présentées, nous les énonçons, indiquons laquelle nous avons choisie et expliquons pourquoi. Enfin, le dernier chapitre fait état des obstacles et des dysfonctionnements qui sont survenus pendant ou après la réalisation du projet et, le cas échéant, des solutions qui ont été apportées ou suggérées.

## Chapitre 2

# Résumé du travail accompli

### Électronique

- Reconstitution du schéma électrique complet de la carte UT modem d'un Minitel 1B Télé-Alcatel à partir du circuit imprimé
- Compréhension du fonctionnement d'un crayon optique et réalisation d'un montage permettant la visualisation du signal émis par celui-ci sur un téléviseur
- Conception d'une carte d'extension pour l'appareil sus-cité
  - Fonctions principales de la carte d'extension : mémoire morte et mémoire vive permettant le stockage et l'exécution de programmes, lecteur de carte SD, contrôleur de bancs mémoire, contrôleur de périphériques dont crayon optique, contrôleur d'interruptions, sortie audio sur haut-parleur ou écouteurs et sur ligne téléphonique
  - Choix des composants et réalisation du schéma électrique
  - Conception des circuits imprimés sur logiciel de CAO et mise en panneau de ceux-ci
  - Fabrication du panneau confiée à une usine
  - Soudage des composants et assemblage des circuits imprimés

### Logique programmable

- Diviseur d'horloge
- Générateur de signal *power-on reset*
- Registres adressables en mémoire permettant l'accès aux périphériques et leur configuration
- Décodeur d'adresse et contrôleur de bancs mémoire
- Bus maître SPI avec tampon de taille 1 octet
- Séparateur de synchronisation par machine de Moore à quatre états
- Contrôleur de crayon optique, dont la fonction est de calculer et enregistrer les coordonnées de la position du crayon optique sur l'écran
- Double port PS/2 en *bit-banging*
- Contrôleur d'interruptions signalant, entre autres, le commencement d'une trame vidéo (impulsion de synchronisation verticale), la fin de la transmission d'un octet sur le bus SPI, l'appui sur le bouton du crayon optique, . . .
- Possibilité de faire fonctionner le Minitel tel qu'à l'origine, par pression d'un bouton au moment de la mise sous tension

**Programmes pour le Minitel modifié**

- Chargeur de programmes depuis une carte SD, avec gestion minimale du système de fichiers FAT32
- Programme de *ROM dump* : lit le contenu de la ROM interne du microcontrôleur, qui renferme le programme d'origine du Minitel, et l'envoie à un PC par le biais du port série
- Dessin sommaire sur l'écran à l'aide du crayon optique
- Synthétiseur basique permettant de jouer de la musique à l'aide du clavier et du crayon optique, ce dernier produisant une variation continue de l'intensité du son par déplacement vertical, et de sa fréquence par déplacement horizontal
- Terminal SPI permettant d'envoyer manuellement des données sur le bus SPI et de voir les données reçues

**Programme utilitaire pour PC**

- Générateur de fichier SVF pour la programmation *in situ* de l'EEPROM de la carte M5

## Chapitre 3

# Architecture du Minitel

Le Minitel est un ensemble terminal-modem. En somme, il affiche à l'écran les données envoyées par le serveur, et envoie à ce dernier les frappes clavier de l'utilisateur. Quelques fonctions typiques des terminaux informatiques sont prises en charge, telles que le soulignement des caractères ou le déplacement du curseur.

Une rapide analyse de l'architecture matérielle du Minitel montre que la totalité de la fonction terminal est réalisée par logiciel, sur un microcontrôleur. Cette même analyse révèle la présence d'une rangée de 32 broches donnant essentiellement accès aux bus d'adresse et de données du microcontrôleur. Il est donc relativement aisé de relier une mémoire sur ce connecteur et de faire exécuter par le microcontrôleur un autre programme que celui d'origine. C'est sans aucun doute la raison d'être même de cette rangée de contacts.

Nous allons brièvement décrire l'architecture générale du Minitel 1B Téléc-Alcatel et ses composants principaux.

Avant de commencer, signalons que nous avons remarqué, en faisant quelques recherches sur le web, qu'il y a eu plusieurs variantes matérielles mineures au sein du produit dit Minitel 1B Téléc-Alcatel. Par exemple, le nôtre est muni d'un microcontrôleur 8052 avec ROM interne, tandis que certains sont munis d'un 8032, compatible avec le 8052 mais sans ROM interne, et d'une carte fille enfichée sur la rangée de broches sus-citée comprenant une ROM externe. Aussi, tout ce qui suit dans ce chapitre ainsi que les schémas de l'annexe A se rapportent à l'exemplaire que nous possédons et certains éléments peuvent ne pas correspondre à d'autres.

L'appareil se décompose en six parties matérielles principales : le boîtier, le clavier, le tube cathodique (noir et blanc), une petite carte portant la mention **FILTRE SECTEUR**, une grande carte portant la mention **CARTE ALIM VIDEO**, une grande carte portant la mention **CARTE UT MODEM**. Par commodité, nous désignerons cette dernière simplement par *carte UT*.

La carte alim-vidéo génère diverses tensions d'alimentation à partir de la tension secteur et pilote le tube cathodique. La carte UT comprend toute l'électronique numérique ainsi que l'électronique analogique de la partie modem. Ces deux cartes sont électriquement et mécaniquement reliées entre elles par un connecteur à 7 broches. Celui-ci véhicule les tensions d'alimentation basse tension de la carte alim-vidéo vers la carte UT, et le signal vidéo généré par la carte UT vers la carte alim-vidéo.



Nous parlerons par la suite exclusivement de la carte UT, car c'est elle seule qui présente un intérêt pour le projet que nous voulons réaliser, et c'est avec elle uniquement que les circuits que nous allons créer doivent s'interfacer.

La figure 3.1 illustre de manière simplifiée l'architecture matérielle de l'appareil.

Les deux composants essentiels dans la mise en œuvre de la carte UT et les plus intéressants pour notre projet sont le microcontrôleur et le processeur graphique.

**8052** Célèbre microcontrôleur d'Intel de la famille MCS-51, encore utilisée de nos jours dans l'industrie. Il comporte 8 kilo-octets de ROM et 256 octets de RAM. Il a la particularité de rendre ses bus d'adresse et de données et leurs lignes de contrôle accessibles sur ses broches, comme un microprocesseur, ce qui rend possible l'exécution de programmes à partir d'une mémoire externe.

**EF9345** Processeur d'affichage de Thomson, utilisé dans certains micro-ordinateurs français comme le VG5000 $\mu$  de Philips ou l'Alice 32 de Matra. Il nécessite une RAM externe dédiée. Il génère un signal vidéo numérique TTL à 3 ou 4 bits de couleur, de standard (nombre de lignes, impulsions de synchronisation. . .) compatible télévision. Il permet l'affichage de caractères alphanumériques et semi-graphiques. Par commodité, nous l'appellerons par la suite VGP (*Video Graphics Processor*), qui est l'appellation utilisée officiellement par les PTT dans les *Spécifications Techniques d'Utilisation du Minitel 1B*.

On retrouvera en annexe A le schéma électrique de la carte UT que nous avons réalisé. Nous ne pouvons pas en toute honnêteté exclure la possibilité que celui-ci comporte quelques erreurs. Nous prions aussi le lecteur de nous excuser pour la mauvaise qualité du schéma, réalisé à la main sur papier. Le temps nous manquait et refaire le schéma au propre sur ordinateur n'était pas une priorité.

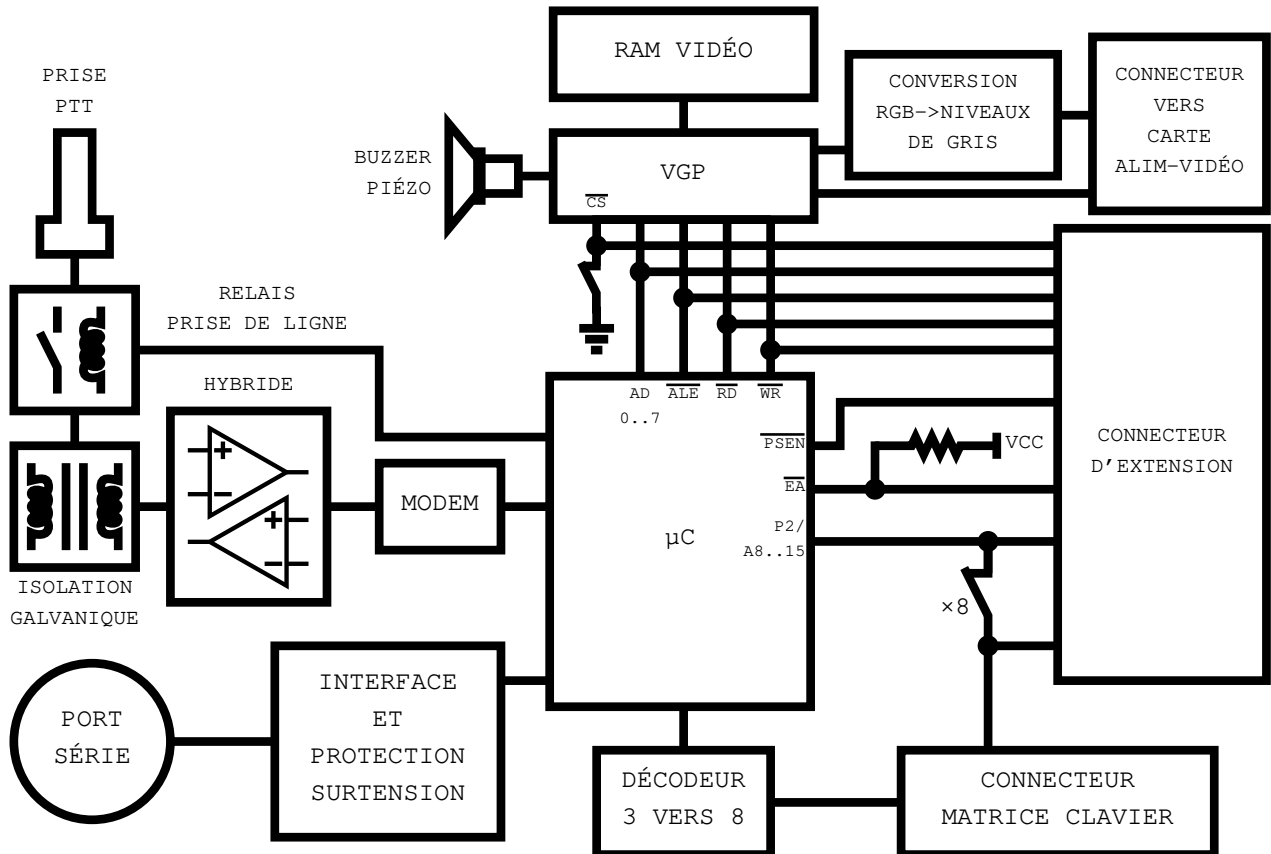


FIGURE 3.1 – Architecture matérielle simplifiée du Minitel 1B Télécom-Alcatel

# Chapitre 4

## Conception

### 4.1 Caractéristiques techniques

Nous ne donnons pas de cahier des charges *stricto sensu* car nous n'en avons pas établi rigoureusement un au début du projet (voir paragraphe 5.1) et car il serait inepte de vouloir effectuer une analyse fonctionnelle *a posteriori*. En lieu et place, nous résumons ici les principales caractéristiques techniques de l'ensemble réalisé.

**Logique programmable** CPLD/FPGA Altera MAX V EPM240ZT100

**Mémoire** EEPROM 64 Ko, RAM 128 Ko

**Stockage** carte SD

**Audio-Vidéo**

sortie 8 bits mono sur haut-parleur intégré ou écouteurs

sortie 8 bits mono sur ligne téléphonique

sortie audio et vidéo RGB sur prise Péritel

**Périphériques** 2 ports PS/2, crayon optique

**Extension**

connecteur pour bus SPI avec quatre lignes de sélection, ligne de *reset* et ligne d'interruption

connecteur d'entrée pour incrustation vidéo et audio

**Boutons**

bouton de réinitialisation

bouton d'interruption

### 4.2 Considérations préliminaires

#### 4.2.1 Justification de l'emploi d'un CPLD

En se référant à la figure 3.1 qui présente l'architecture générale du Minitel, on peut remarquer que l'ajout d'une carte d'extension contenant une mémoire morte, éventuellement de la mémoire vive, et quelques circuits intégrés de logique discrète (*glue logic* dans la terminologie anglophone), est suffisant pour transformer l'appareil en un micro-ordinateur programmable. Cependant, l'ajout de multiples lignes d'entrées-sorties pour la prise en charge de périphériques

(carte SD, ports PS/2, ...) au moyen d'une telle carte nécessiterait un très grand nombre de circuits de logique discrète. C'est pourquoi nous avons choisi d'utiliser un CPLD pour remplacer la quasi-totalité de la logique discrète par un circuit intégré unique.

D'autres facteurs nous ont incité à opter un CPLD.

- La programmabilité du CPLD nous permet, une fois le projet physiquement réalisé, de modifier à loisir une partie importante de l'architecture du système, d'ajouter ou de supprimer certaines fonctionnalités, de corriger d'éventuelles erreurs.
- L'utilisation d'un CPLD permet d'implémenter des modules qu'il aurait été fastidieux de réaliser en logique discrète. Avec le CPLD, nous pouvons par exemple implémenter un véritable contrôleur SPI tel qu'on peut en trouver dans tout microcontrôleur moderne ; la programmation logique s'en trouve simplifiée.
- La dernière raison pour laquelle nous avons voulu utiliser un CPLD n'est pas d'ordre technique ; elle relève de l'apprentissage personnel. Cela faisait longtemps que nous souhaitions expérimenter avec la logique programmable, et ce projet nous en donnait l'opportunité.

### 4.2.2 Modifications sur l'appareil d'origine

Quelques modifications sur la carte UT sont nécessaires : quelques cavalier à retirer, des broches de contact à rajouter pour avoir accès aux signaux qui ne sont pas disponibles sur le connecteur d'extension (par exemple les signaux *RESET* et  $\overline{INT0}$  du microcontrôleur), et quelques autres modifications mineures.

### 4.2.3 Principe général de l'interfaçage avec les périphériques

Le CPLD doit faire l'interface entre le microcontrôleur de la carte UT et les périphériques. À cette fin, nous utilisons la technique des entrées-sorties adressables en mémoire (*memory-mapped I/O*), qui est presque universellement utilisée et très flexible. Une petite partie de l'espace d'adressage (quelques octets) est utilisée par le CPLD pour communiquer avec le microcontrôleur. Ce dernier lit et écrit dans cette partie de l'espace d'adressage pour contrôler les périphériques ou modifier la configuration de certains modules.

### 4.2.4 Aspects mécaniques

La coque du Minitel possède des encoches permettant de maintenir une éventuelle carte fille en place. La carte M5 s'enfiche sur le connecteur d'extension du Minitel et s'insère naturellement dans ces encoches.

Cependant, un espace d'environ 5 mm existe entre le bord de la carte M5 et le bord de la fenêtre d'ouverture à l'arrière de la coque. Par conséquent, si l'on place un connecteur sur la carte M5, une partie importante de celui-ci sera caché par le boîtier, rendant son utilisation impossible.

Au moins deux solutions sont envisageables : rehausser les connecteurs de la carte M5 en les installant sur un autre circuit imprimé, lui-même empilé quelques millimètres au dessus de la carte M5 ; ou bien employer des connecteurs de type *panel mount*. Les connecteurs *panel mount* ne se soudent pas directement sur le circuit imprimé, mais se fixent au châssis de l'appareil et sont reliés au circuit imprimé par des fils.

La deuxième solution aurait été moins coûteuse et permet une certaine flexibilité dans le placement des connecteurs, mais elle aurait nécessité de percer dans le cache en plastique du Minitel des

ouvertures de formes et de dimensions assez précises. Or, ne possédant pas l'outillage adéquat et connaissant nos capacités en matière de bricolage, nous avons préféré choisir la première solution.

Le nombre de connecteurs que nous avons souhaité mettre à disposition s'est avéré assez ambitieux. Pour pouvoir tous les caser dans la surface offerte par l'ouverture de la coque, nous avons dû en placer certains sur des petits circuits imprimés séparés, empilés sur la carte M5.

Le résultat est un design composé d'un certain nombre de cartes empilées, mécaniquement fixées entre elles par des entretoises filetées (*standoffs*) et des vis (voir photographies annexe G). Ce n'est pas très élégant et, bien qu'étant une pratique standard pour la fixation de circuits imprimés, serait assez lourd dans le cadre d'une fabrication industrielle de masse. Mais pour un prototype fait main cela est tout à fait acceptable et fonctionnel.

### 4.3 CPLD

Notre choix s'est porté sur le MAX V du fabricant Altera, dans sa déclinaison EPM240ZT100 (boîtier TQFP 100 broches, 240 éléments logiques). La famille MAX V est relativement récente puisque introduite en 2010, ne coûte pas cher et convient parfaitement à l'usage que nous souhaitons en faire.

### 4.4 Diviseur d'horloge

Sur la carte UT, un oscillateur à quartz et portes inverseuses fournit au VGP un signal d'horloge de 12 MHz. La carte M5 récupère ce signal et le fait passer par le translateur de niveaux MC74VHCT50A qui le convertit aux niveaux LVCMOS 3,3 V. Enfin, ce signal d'horloge LVC-MOS est acheminé jusqu'à l'une des entrées d'horloge du CPLD.

Le CPLD dispose ainsi d'une horloge à 12 MHz. D'autres horloges, de fréquences moins élevées, sont requises pour quelques uns des modules que nous voulons implémenter dans le CPLD. Les valeurs des fréquences requises ne sont pas cruciales et un simple diviseur d'horloge par puissances de 2 à base de compteur binaire suffit pour obtenir un panel de fréquences convenant à notre application. Le compteur implémenté est de taille 6 bits, ce qui permet d'obtenir, en plus de l'horloge d'entrée à 12 MHz, des horloges à 6 MHz, 3 MHz, 1,5 MHz, 750 kHz, 375 kHz, 187,5 kHz.

### 4.5 *Power-on reset*

Il est primordial de s'assurer qu'à la mise sous tension de l'appareil, le microcontrôleur de la carte UT n'entre en opération que lorsque la tension d'alimentation du CPLD est stabilisée et que celui-ci a terminé sa phase de configuration (dont la durée maximale est de 200  $\mu s$ ).

À cette fin, nous avons fait en sorte que la broche *RESET* du microcontrôleur soit maintenue à l'état actif tant que le CPLD n'est pas prêt. Cela se fait simplement en exploitant le fait que les broches d'entrées-sorties du CPLD MAX V sont maintenues en état de haute impédance (avec résistance interne de tirage vers le haut) tant que les circuits de logique programmable n'ont pas démarré.

Nous avons relié la ligne *RESET* du microcontrôleur à une broche du CPLD et ajouté une résistance de tirage vers le haut de valeur adéquate (la *datasheet* du microcontrôleur 8052 spécifie que l'activation de *RESET* peut nécessiter un courant minimal allant jusqu'à  $I_{IH1} = 500 \mu A$ , or la résistance interne du CPLD est trop élevée pour induire un tel courant). Lorsque le CPLD a terminé sa configuration, on fait passer sa sortie à l'état bas pour désinhiber le microcontrôleur.

## 4.6 Traduction des niveaux de tension

De plus en plus de circuits intégrés modernes ne supportent pas des tensions d'alimentation ou d'entrée de 5 V. 3,3 V est sans doute la valeur maximale la plus répandue. Le CPLD MAX V ne fait pas exception. Or, nous souhaitons l'employer dans un environnement majoritairement 5 V. Il est donc nécessaire d'interfacer le CPLD avec les signaux 5 V de façon appropriée.

Dressons d'abord la liste des signaux 5 V que l'on souhaite interfacer et leur direction relativement au CPLD.

1. Le bus *AD0..7* du microcontrôleur · bidirectionnel
2. Le bus *A8..15* du microcontrôleur · bidirectionnel (à première vue, direction microcontrôleur vers CPLD uniquement, mais direction CPLD vers EEPROM en mode de programmation *in situ* de l'EEPROM – voir paragraphe 4.9)
3. Les lignes de contrôle de bus du microcontrôleur (*ALE*,  $\overline{PSEN}$ ,  $\overline{RD}$ ,  $\overline{WR}$ ) · entrée
4. L'horloge à 12 MHz extraite de la carte UT · entrée
5. Le signal de synchro émis par le VGP · entrée
6. Les lignes de communication des ports PS/2 · bidirectionnel, collecteur ouvert
7. Le signal du phototransistor du crayon optique · entrée
8. Le signal du lecteur de code barre du connecteur pour crayon optique · entrée
9. Le bouton du crayon optique · entrée

Pour ce qui suit, on pourra se référer aux schémas électriques de la carte M5, à l'annexe C.

Pour les points 1 et 2, nous avons employé le circuit intégré 74ALVC164245. Il s'agit d'un traducteur de niveaux à double alimentation (dans notre cas, 3,3 V et 5 V) composé de deux ensembles de huit tampons bidirectionnels chacun, à entrées compatibles TTL. Chaque ensemble de huit tampons comporte aussi une broche de contrôle de direction et une broche d'inhibition (haute impédance).

Pour les signaux unidirectionnels des points 3, 4 et 5, nous avons choisi le circuit intégré MC74VHCT50A de ON Semiconductor. Il comporte six tampons unidirectionnels à entrées compatibles TTL et tolérantes 5 V (même avec une tension d'alimentation inférieure). L'une des applications de cette puce est de l'alimenter en 3,3 V et de lui présenter en entrée des signaux 5 V. Les entrées ne sont pas endommagées et les signaux d'entrée sont ramenés aux niveaux CMOS 3,3 V en sortie.

Concernant le point 6, il est essentiel de noter que les lignes des ports PS/2 sont à collecteur ouvert et à fréquence de commutation relativement lente (environ 17 kHz au maximum). Sur la carte M5, une résistance mise en série et une diode Schottky de *clamp* à 3,3 V limitent la tension de chaque ligne à environ 3,6 V, ce qui est convenable puisque le CPLD MAX V tolère

jusqu'à 4 V en entrée. Nous nous sommes rendu compte – trop tard – que cette méthode n'est pas fiable car elle suppose qu'une tension de 3,3 V est suffisamment élevée pour que le contrôleur du périphérique l'interprète comme un état logique haut, ce qui n'est pas garanti. Une méthode convenable aurait été d'employer une interface à base de transistor MOSFET comme celle décrite à cette adresse.<sup>1</sup>

Nous plafonnons les signaux 7 et 8 à environ 3,6 V grâce à une résistance de valeur relativement faible mise en série et à une diode Schottky de *clamp* au rail 3,3 V.

Enfin, pour le signal 8, nous employons un pont diviseur de tension résistif pour abaisser la tension de 5 V à environ 3,3 V. Un condensateur forme avec les résistances du diviseur de tension, un filtre RC passe-bas qui fait office de dispositif anti-rebond.

## 4.7 SRAM

Le choix de la SRAM doit respecter une contrainte particulière. Les broches A8 à A15 de la SRAM sont directement reliées au microcontrôleur du Minitel, qui possède des étages d'entrée-sortie de niveaux TTL. Or une sortie de type TTL ne peut généralement pas être reliée directement à une entrée CMOS, car le standard TTL prévoit une tension de sortie haute minimale  $V_{OH,min} = 2,4 V$ , alors que le standard CMOS 5 V impose une tension d'entrée haute minimale  $V_{IH,min} = 3,5 V$  (peut varier quelque peu d'un circuit intégré à un autre). Il est donc impératif de choisir, soit une vieille SRAM TTL de récupération (il semble fort que l'on n'en fabrique plus, la technologie TTL ayant depuis longtemps cédé la place au CMOS), soit une SRAM CMOS avec entrées-sorties compatibles TTL. Cette dernière catégorie apparaît plus rarement que les SRAM purement CMOS dans les catalogues des revendeurs, mais on en trouve quand même.

Notre choix s'est porté sur la mémoire AS7C1024B fabriquée par Alliance Memory, dont la taille est de 128K × 8 bits. L'espace d'adressage du microcontrôleur 8052 ayant une taille de 16 bits, et cette mémoire étant adressable sur 17 bits, deux alternatives apparaissent :

- fixer l'état logique de l'entrée A17 de la SRAM à 0 ou à 1 et traiter la SRAM comme une mémoire de 64K × 8 bits ;
- relier l'entrée A17 de la SRAM au CPLD et programmer dans ce dernier un mécanisme de commutation de bancs mémoire (*bank switching*).

Nous avons opté pour la deuxième alternative, parce qu'elle ne demandait que peu de travail supplémentaire et que nous trouvions dommage de gaspiller 64 Ko sur les 128 offerts par la puce.

## 4.8 EEPROM

L'EEPROM est la mémoire qui contient le programme exécuté lors de la mise sous tension de l'appareil. On peut dégager deux philosophies très différentes à cet égard.

1. La philosophie que l'on pourrait qualifier de traditionnelle et que l'on retrouve dans la plupart des architectures d'ordinateur, à savoir utiliser une EEPROM parallèle accessible sur les bus d'adresse et de données du système. Le microcontrôleur accède tout de suite à l'EEPROM dès sa mise sous tension et commence l'exécution du programme immédiatement.

---

1. *Hobbytronics*. Bi-Directional MOSFET Voltage Level Converter 3.3V to 5V  
<http://www.hobbytronics.co.uk/mosfet-voltage-level-converter>

2. Un procédé plus indirect, permis par la présence du CPLD. Il s'agit d'utiliser une EEPROM série (bus SPI ou I<sup>2</sup>C notamment) connectée au CPLD. La RAM couvre presque l'entièreté de l'espace d'adressage. À la mise sous tension ou après une réinitialisation, le CPLD maintient la broche *RESET* du microcontrôleur à l'état actif. Les entrées-sorties du 8052 sont alors, soit flottantes pour certaines, soit à l'état de haute impédance avec résistance interne de tirage vers le haut pour d'autres. Le CPLD peut alors devenir maître sur les bus d'adresse et de données sans créer de contention avec le microcontrôleur. Une machine à état programmée dans le CPLD lit alors le contenu de l'EEPROM série et le copie dans la RAM, octet par octet. Une fois ce processus achevé, le CPLD désactive la ligne *RESET* du microcontrôleur et ce dernier commence l'exécution du programme copié dans la RAM. On peut aussi utiliser la mémoire flash interne du CPLD MAX V à la place d'une EEPROM série, mais l'interfaçage de cette mémoire interne requiert une très importante quantité de ressources logiques.

Le temps nécessaire à la copie du contenu de l'EEPROM série dans la RAM est négligeable. Il est de l'ordre de 10 ms pour une ROM de 8 Ko.

Un inconvénient est la nécessité d'implémenter un contrôleur SPI ou I<sup>2</sup>C dans le CPLD. Mais dans notre cas ce dernier point est caduc puisque nous avons de toute façon décidé d'implémenter un contrôleur SPI pour la communication avec la carte SD.

La méthode série offre quelques avantages intéressants – pas tellement pour un projet personnel à une unité, mais du moins le seraient-ils dans l'industrie.

- Les EEPROM série sont beaucoup plus répandues et bénéficient d'une offre plus large que les EEPROM parallèles. Ces dernières sont en voie de disparition – on peut dans certains cas les remplacer par une mémoire Flash parallèle.
- Les EEPROM série sont moins chères.
- Le routage d'une EEPROM série sur le circuit imprimé est bien plus simple, facile et rapide à faire que celui d'une EEPROM parallèle.

Tout porte donc à opter pour une EEPROM série. Pourtant, nous avons choisi la première méthode avec EEPROM parallèle. La raison est double. Premièrement, l'idée de l'EEPROM série ne nous est venue que tard dans la conception du projet, et il aurait été beaucoup trop chronophage de faire machine arrière. Deuxièmement, n'ayant jamais utilisé de logique programmable par le passé, nous n'étions pas pleinement confiant de la faisabilité du procédé de l'EEPROM série ; tandis que celui de l'EEPROM parallèle est simple et éprouvé. Nul doute que si nous devions réaliser un projet semblable à l'avenir, nous opterions pour l'EEPROM série.

## 4.9 Programmation de l'EEPROM par JTAG

Nous décrivons dans cette section le procédé que nous avons mis en place pour programmer l'EEPROM *in situ*, c'est à dire sans avoir à démonter le Minitel et retirer celle-ci de son support. Par là même, cette technique permet de s'affranchir de l'achat d'un support, et autorise même à employer une EEPROM à montage en surface plutôt qu'en boîtier DIP. Par mesure de précaution, au cas où il se serait avéré impossible de mettre en œuvre cette technique pour une raison ou une autre, nous avons tout de même utilisé une EEPROM DIP sur support.

La programmation *in situ* de l'EEPROM repose sur deux points essentiels.

- Dans notre conception, le CPLD contrôle la broche *RESET* du microcontrôleur 8052. Lorsque cette broche est maintenue à l'état actif, les entrées-sorties sont soit flottantes pour certaines,



soit à l'état de haute impédance avec résistance interne de tirage vers le haut pour d'autres. Le CPLD peut alors devenir maître sur les bus d'adresse et de données sans créer de contention avec le microcontrôleur. Il peut donc lire et écrire dans l'EEPROM.

- Tous les circuits intégrés de logique programmable modernes, ainsi que certains microprocesseur et bien d'autres puces encore, possèdent une fonctionnalité standard appelée BSC (*Boundary-Scan Testing*). En résumé, un appareil connecté à cette puce par liaison JTAG (le plus souvent un PC) peut prendre le contrôle des broches d'entrées-sorties de la puce, lire leur état et le modifier. Cette fonction a été conçue à l'origine pour tester les interconnexions entre différentes puces sur un circuit imprimé. Cependant, dans notre cas, nous pouvons en user à notre avantage pour avoir accès aux bus d'adresse et de données ainsi qu'aux lignes de contrôles afin de programmer l'EEPROM à partir d'un PC.

Il faut concevoir un moyen d'envoyer de façon automatisée sur la liaison JTAG les instructions BSC permettant de basculer les entrées-sorties du CPLD pour programmer successivement chaque octet de l'EEPROM. Pour cela, il existe des langages standardisés dont le but est de contrôler automatiquement l'échange de données sur une ligne JTAG. Nous en présentons deux parmi les plus utilisés.

#### **Le format de fichier SVF (*Serial Vector Format*)**

Il contient une suite séquentielle de paquets de données à envoyer sur la ligne JTAG, et optionnellement un motif à comparer avec les données reçues. Il ne permet pas de réaliser des algorithmes ; chaque ligne est exécutée inconditionnellement et sans branchement possible.

#### **Le langage STAPL (*Standard Test and Programming Language*)**

Développé par le fabricant Altera. Il s'agit d'un véritable langage de programmation et permet à ce titre de réaliser des opérations et des algorithmes complexes.

Le standard SVF ne possède que quelques instructions et s'apprend très rapidement, tandis que le langage STAPL demande un effort d'apprentissage beaucoup plus grand. Comme les fonctionnalités du SVF, bien que très limitées, sont suffisantes pour effectuer la programmation de l'EEPROM, nous nous sommes naturellement orientés vers celui-ci.

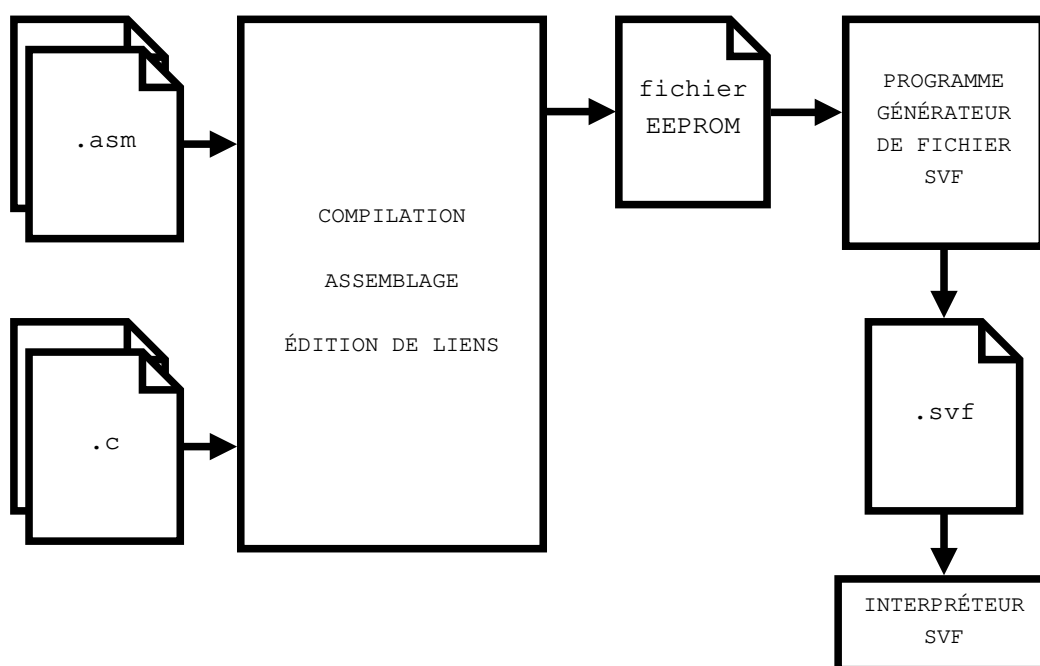
Nous avons écrit un programme pour PC qui prend en entrée le fichier à programmer, contenant une liste de couples (*adresse, donnée*), et écrit en sortie un fichier SVF qui réalise l'écriture puis la vérification de ces données dans l'EEPROM.

À titre d'exemple, nous donnons en annexe E un fichier SVF qui programme puis vérifie quatre octets de l'EEPROM.

Enfin, la figure 4.1 représente le flux de travaux (*workflow*) que nous suivons lors de la création de programmes pour le Minitel modifié.

## **4.10 8052 et architecture de Von Neumann**

Le microcontrôleur 8052 possède une architecture Harvard : deux espaces d'adressages séparés existent, l'un pour la mémoire de programme et l'autre pour la mémoire de données. Le premier est en lecture seule et est associé à une broche de contrôle de lecture  $\overline{PSEN}$ . Le second est en lecture-écriture et est associé à deux broches de contrôle  $\overline{RD}$  et  $\overline{WR}$ . Lorsque le 8052 lit une instruction à exécuter, il le fait obligatoirement depuis le premier type de mémoire. En l'état, il est impossible de le faire exécuter des instructions depuis le second type de mémoire. Cette caracté-

FIGURE 4.1 – Flux de travaux de la programmation *in situ* de l'EEPROM

téristique permet notamment d'empêcher l'exécution de code malveillant depuis la mémoire vive.

Toutefois, dans le cas d'un micro-ordinateur personnel – ce que nous souhaitons réaliser avec ce projet, on préfère généralement une architecture de Von Neumann, avec un seul espace d'adressage. Ceci est essentiel pour pouvoir charger des programmes depuis un support de stockage externe vers la RAM et l'exécuter.

C'est pourquoi nous avons modifié l'architecture du système en architecture de Von Neumann. La façon la plus simple de faire cela est de combiner les lignes  $\overline{PSEN}$  et  $\overline{RD}$  avec une porte ET (dans notre cas, c'est le CPLD qui s'en charge). C'est une technique très courante dans les projets à base de 8052. Le signal peut alors être utilisé conjointement avec  $\overline{WR}$  comme on le ferait avec un microprocesseur Von Neumann conventionnel. Il est actif lorsque le 8052 souhaite lire dans l'un ou l'autre des deux types de mémoire, de manière indifférenciée.

Il est vrai que cette technique divise la quantité de mémoire adressable par deux. La flexibilité du CPLD aidant, sans doute est-il possible d'imaginer des systèmes d'adressage plus complexes et moins gaspilleurs. Nous ne nous y sommes cependant pas attardé.

## 4.11 Commutation des bancs mémoire

Comme nous l'avons mentionné au paragraphe 4.7, nous avons réalisé un système de commutation de bancs mémoire afin de permettre au 8052, qui ne peut adresser que 64 Ko, d'accéder aux 128 Ko de la RAM (un tout petit peu moins en réalité), en plus des 32 Ko de l'EEPROM.

Nous avons implémenté dans le CPLD un registre de configuration mémoire. Le CPLD décode l'adresse émise par le microcontrôleur et, en fonction de cette adresse et du contenu de ce registre, active l'EEPROM, la RAM, le VGP, ou ses propres registres – une seule de ces alternatives à la fois. Le CPLD contrôle aussi, en fonction de ces mêmes variables, la ligne A16 de la RAM.

Une multitude de configurations et de partitionnements de l'espace d'adressage sont imaginables. Grâce à la programmabilité du CPLD, il est toujours possible de remanier partiellement ou totalement ce partitionnement (bien sûr, chaque remaniement est susceptible de rendre les programmes écrits précédemment incompatibles ; le partitionnement de l'espace d'adressage est quelque chose qui doit à un moment ou à un autre être décidé une fois pour toutes). À titre indicatif, nous donnons dans la figure 4.2 le partitionnement que nous avons mis en œuvre.

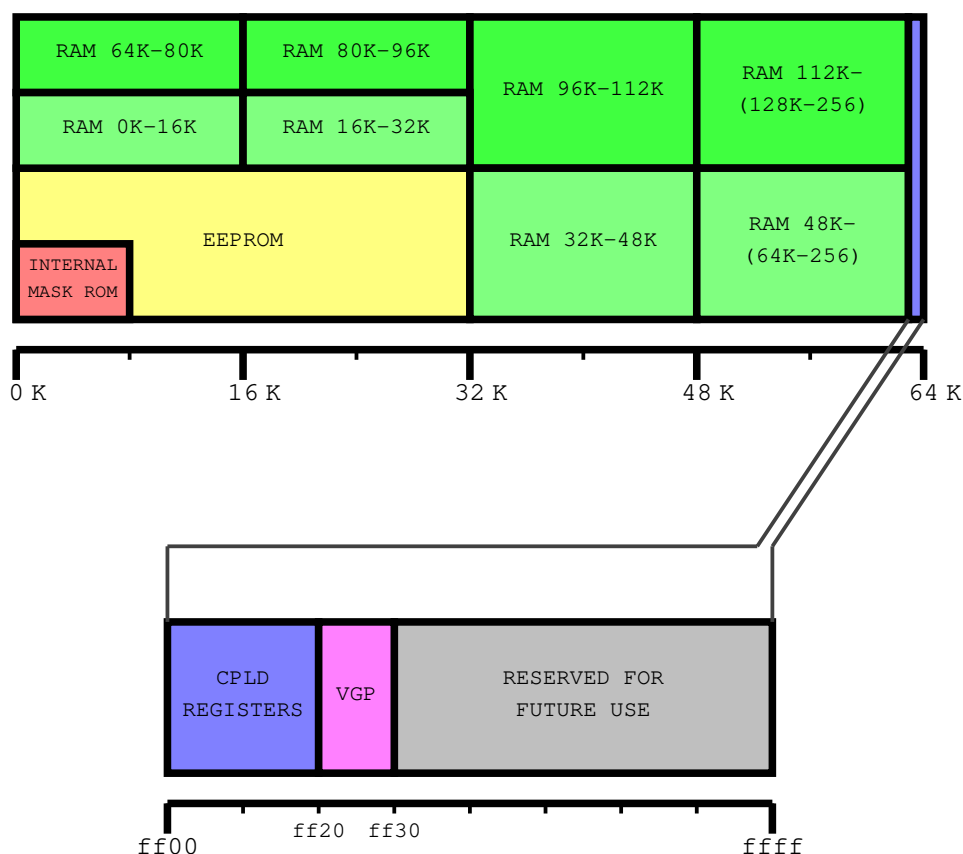


FIGURE 4.2 – Partitionnement de l'espace d'adressage

## 4.12 Bus SPI

Nous avons implémenté en logique programmable un bus SPI maître adressable en mémoire. L'écriture d'un octet à l'adresse mémoire appropriée par le microcontrôleur déclenche la transmission de cet octet sur le bus SPI. La lecture à cette même adresse donne au microcontrôleur l'octet reçu. Le CPLD peut émettre une interruption lorsque la transmission et la réception de

l'octet est achevée.

Sept sorties du CPLD sont utilisées comme lignes de sélection de périphérique : une pour la carte SD, une pour le DAC, une pour une éventuelle carte fille interne, quatre pour le connecteur d'extension décrit en 4.19.1.

La fréquence de l'horloge SPI émise par le CPLD peut être configurée par le logiciel à différentes valeurs (données en 4.4) par l'écriture dans un registre de configuration.

### 4.13 Carte SD

La carte SD est reliée au bus SPI que nous avons implémenté dans le CPLD. Les interrupteurs de détection de carte et de protection en écriture, offerts par la plupart des connecteurs pour carte SD, sont aussi reliés au CPLD avec un circuit RC anti-rebond. Le microcontrôleur peut lire l'état de ces interrupteurs *via* les registres adressables en mémoire implémentés en logique programmable. Le CPLD peut être configuré par le logiciel pour émettre une interruption lors de l'insertion ou du retrait d'une carte.

L'emploi d'un périphérique échangeable à chaud (*hot-swappable*) nécessite quelques précautions, notamment en ce qui concerne l'alimentation de celui-ci. En effet, l'insertion à chaud d'un périphérique est suivie d'un fort courant momentané (*inrush current*) sur la ligne d'alimentation du périphérique pendant que les capacitances internes à ce dernier se chargent, ce qui entraîne temporairement une chute de la tension du rail d'alimentation. Une chute trop importante peut déclencher la réinitialisation de certains circuits intégrés, ou les faire entrer dans un état indéterminé.

Pour pallier ce phénomène, nous avons suivi les recommandations données par cette page<sup>2</sup> et placé un condensateur de type *low ESR* (résistance série équivalente faible) au plus près du connecteur pour carte SD.

Nous ne disposons pas du matériel nécessaire (oscilloscope) pour réaliser des mesures et évaluer l'ampleur de ce phénomène dans notre circuit, mais jusqu'à présent toutes les insertions à chaud de carte SD que nous avons faites n'ont semblé déclencher aucun effet de bord. Pour bien faire, il faudrait cependant que nous essayions plusieurs cartes SD de modèles différents, chose que nous n'avons pas encore faite.

### 4.14 Audio

Par souci de simplicité et parce que la fonction audio ne fait pas partie des priorités de ce projet, nous n'avons pas conçu de véritable générateur sonore. Au lieu de cela, le logiciel envoie directement les échantillons sonores au DAC (convertisseur numérique-analogique). Cela implique de faire tous les calculs liés au son en logiciel, ce qui peut dans certains cas imposer une grande charge de travail pour le microcontrôleur.

La section audio est principalement composée d'un double DAC 8 bits, d'un amplificateur opérationnel utilisé en mélangeur, et d'un amplificateur de puissance qui pilote le haut-parleur et la

---

2. How to use MMC/SDC  
[http://elm-chan.org/docs/mmc/mmc\\_e.html](http://elm-chan.org/docs/mmc/mmc_e.html)

sortie pour casque.

La sortie du premier DAC est mélangée avec l'entrée audio du connecteur d'extension (voir 4.19.2) par l'amplificateur opérationnel, puis le signal résultant est acheminé au connecteur Péritel d'une part, et à un potentiomètre de réglage de volume d'autre part. La sortie du réglage de volume est amplifiée par l'amplificateur de puissance.

Le second DAC permet d'injecter un signal audio sur la prise téléphonique du Minitel. Sa sortie passe par deux composants passifs puis est acheminée vers l'entrée d'un amplificateur opérationnel sur la carte UT du Minitel. Cet AOP reçoit à l'origine sur son entrée le signal audio du modem. En y ajoutant la sortie du DAC par l'intermédiaire d'une résistance, on le fait effectivement fonctionner en mélangeur-amplificateur.

## 4.15 Séparateur de synchronisation

L'exploitation du crayon optique nécessite de disposer des deux signaux de synchronisation verticale et horizontale. Le VGP possède deux modes de configuration relatifs à ces signaux.

Mode	Broche HVS / HS	Broche PC / VS
composite	synchro composite	comparateur de phase
séparé	synchro horizontale	synchro verticale

Modes de configuration de la sortie synchro du VGP

La sortie comparateur de phase peut être utilisée pour synchroniser avec précision le VGP et un système externe. Nous ne nous en servons pas pour notre projet.

Le Minitel, d'origine, utilise la synchronisation composite. Le signal PC n'est pas utilisé. La synchronisation composite est utilisée par la carte alim-vidéo du Minitel, et par la carte d'extension Péritel de notre projet, décrite en 4.20.

Deux solutions s'offrent à nous :

- garder la synchronisation composite et en extraire la synchronisation verticale ;
- utiliser le mode H/V et recombinaison  $\overline{HSYNC}$  et  $\overline{VSYNC}$  en un signal de synchronisation composite avec une porte XOR, puis fournir cette synchronisation recombinaison à la carte alim-vidéo et à la carte Péritel.

Nous avons opté pour la première solution car elle n'entraîne pas de modification supplémentaire sur la carte UT, contrairement à la deuxième solution. L'extraction de la synchronisation verticale à partir de la synchronisation composite, appelée séparation de synchronisation, est réalisée par le CPLD au moyen d'une machine de Moore à quatre états, prenant en entrées une horloge de période bien choisie et la synchronisation composite, et donnant en sortie la synchronisation verticale. Cela ne nécessite que très peu de ressources logiques sur le CPLD.

La période  $t_{clk}$  de l'horloge d'entrée doit être telle que  $t_{hsync} < t_{clk} < \frac{1}{2}t_{vsync}$ . L'entrée d'horloge de la machine à états est alimentée par un signal de période  $t_{clk} = 5,333 \dots \mu s$  issu du diviseur d'horloge (voir 4.4). Ce séparateur de synchronisation fonctionne en mesurant (dans le domaine discret, avec une granularité  $t_{clk}$ ) la durée pendant laquelle le signal de synchro composite reste à l'état bas ou à l'état haut.

Le signal de sortie  $\overline{VSYNC}$  accuse d'un retard par rapport à l'impulsion de synchronisation

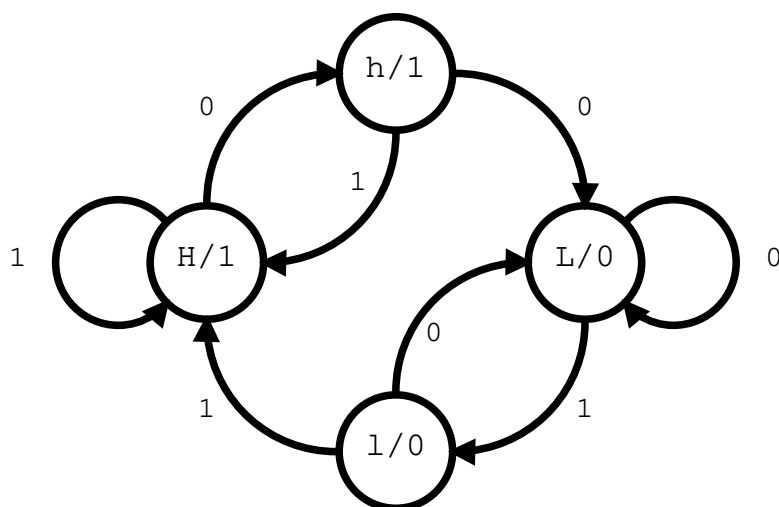


FIGURE 4.3 – Automate du séparateur de synchronisation

verticale présente dans le signal d'entrée composite, mais ceci est un défaut inhérent à la plupart des séparateurs de synchro simples. Par exemple, le très répandu LM1881, séparateur de synchro analogique à base d'intégrateur, exhibe aussi ce défaut.

Ce retard n'a aucune incidence sur le bon fonctionnement du module crayon optique, car il n'entraîne tout au plus qu'une imprécision de  $\pm 1$  ligne sur la localisation du crayon.

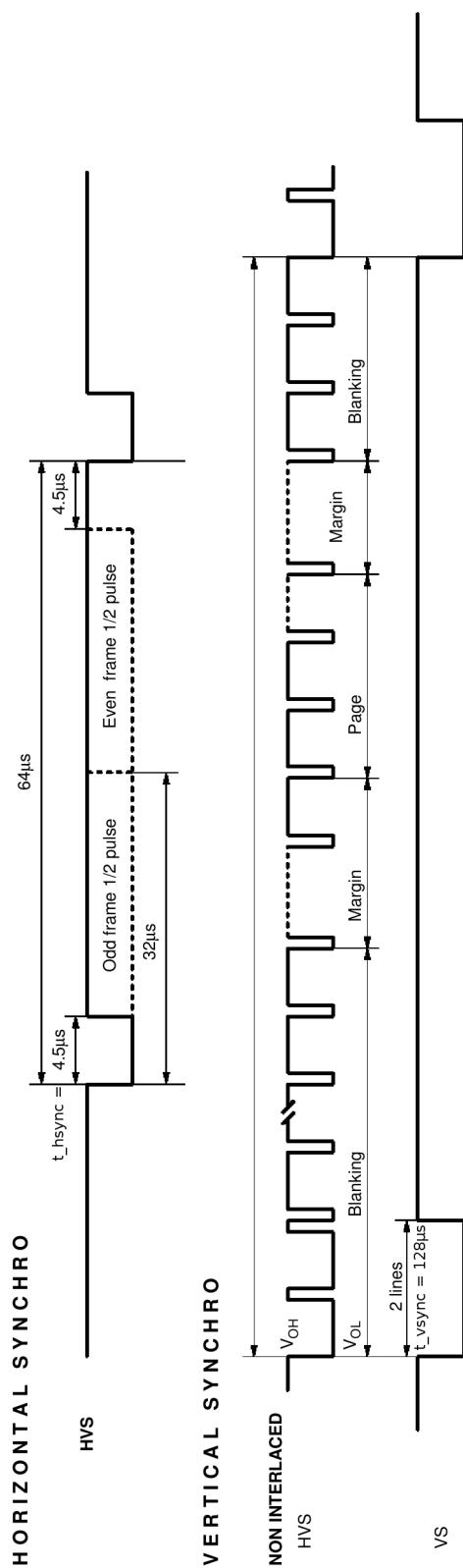


FIGURE 4.4 – Chronogramme du signal de synchronisation du VGP  
 HVS : signal sur la broche de sortie de synchro composite.  
 Extrait de la *datasheet* du EF9345.

## 4.16 Crayon optique

Le crayon optique est un dispositif de pointage inventé en 1955 et popularisé par certains micro-ordinateurs des années 1980. Il prend la forme d'un stylet et permet à l'utilisateur de pointer une position sur l'écran. Il fonctionne avec n'importe quel tube cathodique standard, pourvu que l'ordinateur soit équipé d'une électronique spécialisée dont le rôle est de calculer la position du stylet sur l'écran.

Le crayon optique que nous avons utilisé est un modèle pour ordinateur Thomson MO5. Il possède un connecteur standard DIN 5 broches 240°, dont le brochage est donné par la figure 4.5.

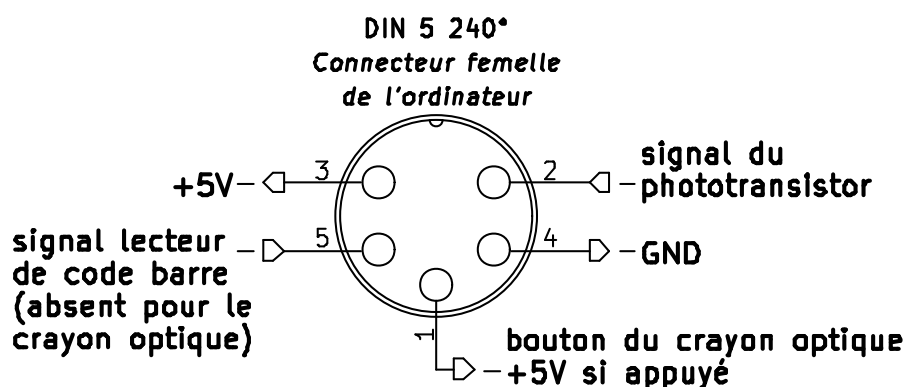


FIGURE 4.5 – Brochage du connecteur pour crayon optique Thomson

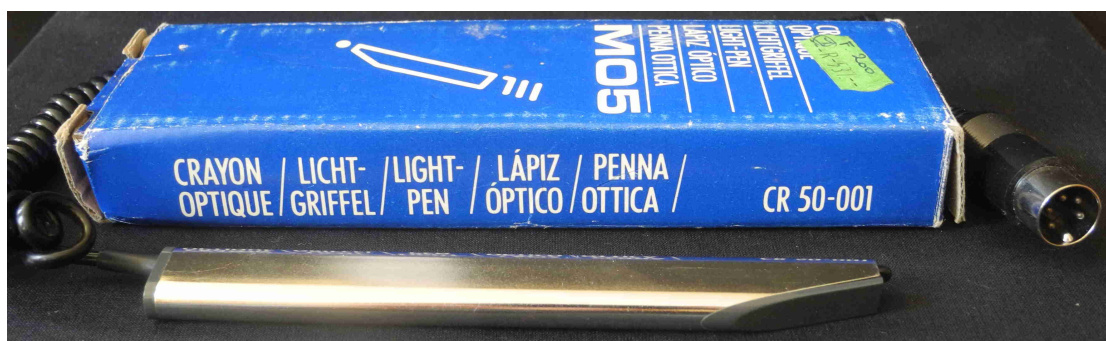


FIGURE 4.6 – Photographie du crayon optique utilisé

Nous commençons par exposer très brièvement le principe de fonctionnement du crayon optique, puis nous décrivons les essais que nous avons effectués avec ce périphérique avant de concevoir la carte M5 ainsi que les observations importantes que nous avons faites. Enfin, nous présentons la partie électronique et la partie de logique programmable relatives au positionnement du crayon optique que nous avons réalisés dans le cadre du projet.

### 4.16.1 Fonctionnement d'un crayon optique

Nous supposons les principes de l'écran à tube cathodique et de la génération d'image par balayage déjà connus du lecteur.



Le crayon optique est muni d'un phototransistor au niveau de sa pointe. Admettons que l'utilisateur maintienne le crayon contre l'écran à un endroit donné. Lorsque le spot lumineux passe près de la pointe, la conductivité du phototransistor augmente. Ce signal peut être reçu et interprété par l'électronique de l'ordinateur qui, en tenant compte des impulsions de synchro générées par le circuit vidéo, peut calculer la position de la pointe. La durée écoulée entre la dernière impulsion de synchro horizontale (qui marque le début du balayage d'une ligne) et la réception du signal lumineux par le crayon optique induit la position horizontale de la pointe. Le nombre d'impulsions de synchro horizontale qui ont eu lieu entre la dernière impulsion de synchro verticale et la réception du signal lumineux induit la position verticale de la pointe.

#### 4.16.2 Test préliminaire

Avant de commencer la conception de la carte M5, nous avons réalisé un montage d'essai dont le but était double : s'assurer que nous pouvions mettre au point un montage fonctionnel exploitant le crayon optique d'une part, et visualiser le signal généré par ce montage d'autre part.

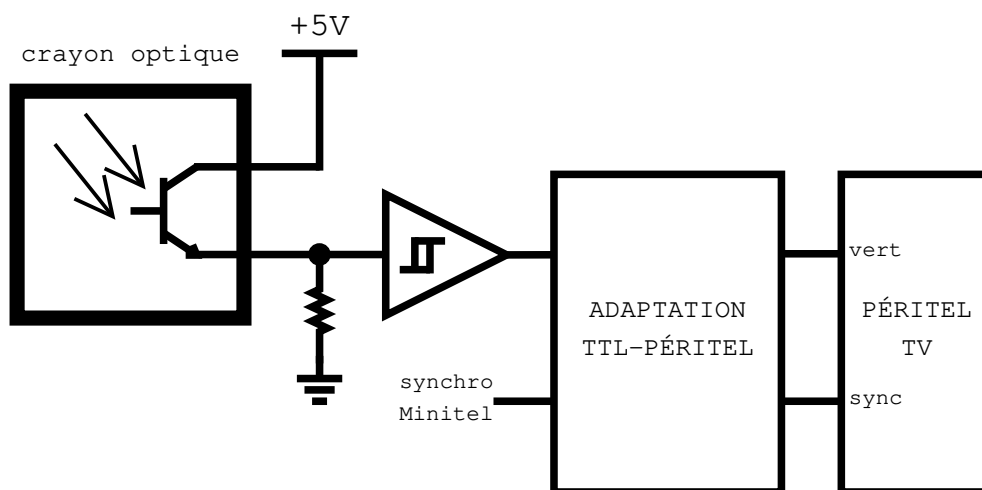


FIGURE 4.7 – Schéma simplifié du circuit de visualisation pour crayon optique

Pour réaliser les essais, nous mettons sous tension le Minitel non modifié et nous remplissons l'écran de blanc en écrivant des caractères avec le clavier. Puis nous synchronisons un téléviseur avec l'écran du Minitel, en extrayant le signal de synchro sur la carte UT du Minitel, en adaptant les niveaux de tension et l'impédance de ce signal à la norme Péritel, et en connectant le signal obtenu à un téléviseur que l'on bascule au préalable sur le canal Péritel approprié. À ce stade, le téléviseur n'affiche aucune image mais son balayage est synchronisé avec celui du tube cathodique du Minitel. Le signal émis par le crayon optique est traité de manière assez sommaire : un tampon à trigger de Schmitt convertit la tension de sortie du phototransistor en un signal binaire, et une résistance entre la ligne de sortie du crayon optique et la masse permet d'ajuster la sensibilité. La sortie du tampon est ensuite convertie aux niveaux de tension et d'impédance de la norme Péritel et reliée à l'une des broches de couleur (rouge, vert ou bleu) de la prise Péritel d'un téléviseur.

L'effet de ce montage est le suivant : lorsque l'on approche le crayon optique de l'écran du Minitel, le téléviseur affiche des lignes de couleur. La position sur l'écran du téléviseur où commencent ces lignes correspond à la position sur l'écran du Minitel où se trouve la pointe du

crayon optique. La figure 4.8 représente l'image obtenue pour un réglage de sensibilité moyen. Si la sensibilité est trop élevée, les lignes recouvrent toute la largeur de l'écran, ce qui témoigne d'un signal peu ou pas exploitable ; si la sensibilité est trop faible, on n'obtient aucune image, ce qui montre que la tension à l'entrée du tampon à trigger de Schmitt est toujours trop basse pour dépasser le seuil d'hystérésis.

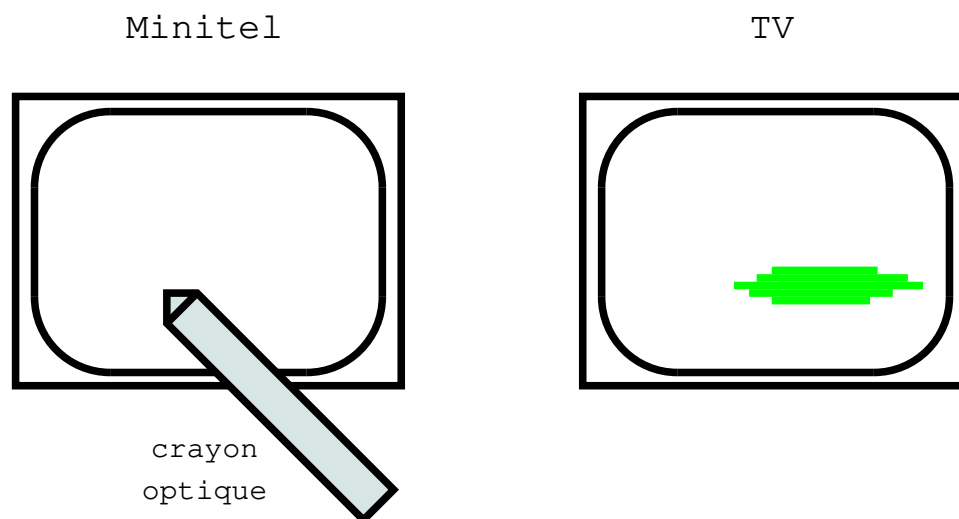


FIGURE 4.8 – Image produite par le circuit de visualisation

### 4.16.3 Observations

La forme affichée, illustrée par la figure 4.6, s'explique par le raisonnement suivant.

- Lorsque la pointe du crayon optique est plaquée contre l'écran, on peut imaginer un rayon de sensibilité autour de la pointe. Quand le spot lumineux du tube cathodique se trouve dans ce rayon, il est assez proche de la cellule du crayon optique pour que celui-ci le détecte et émette un signal.
- La ligne centrale sur l'image obtenue commence plus tôt que les autres lignes pour deux raisons. La première est liée à la forme circulaire du rayon de sensibilité. Sur la ligne de l'écran du Minitel qui passe par le centre de la cellule, le spot entre dans le cercle plus tôt que sur les autres lignes. La deuxième raison est que la tension présente sur la broche de signal du crayon optique est d'autant plus élevée et le temps de réaction du phototransistor d'autant plus faible que la quantité de lumière captée est importante. Or c'est sur la ligne centrale que le spot lumineux passe le plus longtemps dans le rayon de sensibilité et le plus près du centre de la cellule.

Une observation capitale doit être faite : en pointant le crayon optique sur l'écran du Minitel et en maintenant le bras immobile de façon à ce que la pointe du crayon optique bouge le moins possible, l'image du téléviseur n'est pas fixe, contrairement à ce que l'on pourrait attendre. La ligne de couleur centrale reste à peu près constante. Cependant, les lignes les plus extérieures voient leur position de début osciller de manière assez importante.

En conséquence, si nous utilisons un montage de ce type (résistance de réglage et trigger de

Schmitt) dans la carte M5, nous devons concevoir la logique programmable de façon à ce que seule la ligne centrale soit utilisée pour calculer la position du crayon optique sur l'écran.

#### 4.16.4 Circuit du crayon optique sur la carte M5

On se reportera au quatrième schéma de l'annexe C pour les parties du circuit de la carte M5 relatives au crayon optique.

Concernant le signal du phototransistor, nous avons opté pour un étage d'entrée semblable à celui réalisé dans le montage de test (paragraphe 4.16.2).

Un potentiomètre permet à l'utilisateur d'ajuster la sensibilité. N'étant pas certains de la valeur du potentiomètre que nous devons choisir, nous avons pris un potentiomètre de faible valeur (1 k $\Omega$ ) en série duquel nous avons ajouté un connecteur femelle à deux broches destiné à recevoir soit un cavalier si la gamme du potentiomètre est convenable, soit une résistance si celle-ci n'est pas assez élevée (après fabrication du projet, il s'est avéré que c'était le cas).

Une résistance et une diode Schottky limitent la tension émise par le crayon optique à environ 3,6 V (rappelons que le crayon optique est alimenté en 5 V tandis que les banques d'entrées-sorties du CPLD le sont en 3,3 V et tolèrent une tension de 4 V maximum).

Enfin, ce signal est relié à une entrée du CPLD dont on active le trigger de Schmitt interne.

#### 4.16.5 Module crayon optique

Nous donnons en annexe F le code source Verilog relatif au positionnement du crayon optique. Celui-ci repose essentiellement sur un compteur de lignes et un compteur de colonnes (pixel au sein d'une ligne) qui suivent la position du spot lumineux sur l'écran. Le compteur de lignes est réinitialisé (respectivement incrémenté) à chaque impulsion de synchro verticale (respectivement horizontale). Le compteur de colonnes est réinitialisé à chaque impulsion de synchro horizontale. Il est incrémenté par une horloge à 3 MHz. Comme chaque ligne vidéo dure 52  $\mu s$  (sans l'impulsion de synchro), cette fréquence d'horloge induit une division de chaque ligne en 156 colonnes.

Conformément à ce qui a été dit au paragraphe 4.16.3, le module a été conçu pour ne retenir, parmi toutes les impulsions du signal optique d'une même trame vidéo, que celle dont l'intervalle de temps entre l'impulsion elle-même et le début de la ligne correspondante est minimal. Cela est fait au moyen d'une comparaison numérique. Pendant une trame vidéo donnée, chaque impulsion venant du crayon optique est susceptible de déclencher l'enregistrement par une bascule à verrou des coordonnées formées par les compteurs de lignes et de colonnes. Cet enregistrement a effectivement lieu seulement si la coordonnée horizontale (colonne) actuelle est plus petite que celle qui a été enregistrée précédemment. Ainsi, à la fin de la trame, ce sont les coordonnées du début de ce que nous appelions au paragraphe 4.16.3 la ligne centrale qui sont enregistrées.

### 4.17 Ports PS/2

Concernant la liaison électrique des ports PS/2 au CPLD, on se reportera au paragraphe 4.6. Nous expliquons ici l'accès par le logiciel aux ports PS/2 dans un premier temps, puis le brochage du double port PS/2 dans un deuxième temps.

### 4.17.1 Interfaçage avec le microcontrôleur

Afin d'économiser un peu des ressources programmables du CPLD, nous avons choisi de ne pas implémenter de contrôleur de communication, mais de donner au microcontrôleur 8052 l'accès direct à l'état électrique des lignes de communication des ports PS/2, *via* le CPLD. Un registre adressable en mémoire implémenté dans le CPLD permet au microcontrôleur de lire et de modifier l'état des lignes.

Le logiciel exécuté par le 8052 doit donc gérer le protocole, mais aussi la commutation des lignes à l'échelle du bit. Cette technique est communément appelée *bit-banging*. Son avantage est la simplicité du matériel, son inconvénient la charge de travail accrue pour le microcontrôleur.

Notons que nous pouvons nous permettre cette technique grâce à la fréquence de commutation relativement faible des ports PS/2, et grâce à la possibilité que laisse la norme PS/2 au microcontrôleur d'inhiber la communication quand il le souhaite.

La fréquence maximale s'élève à environ 17 kHz. Dans le pire des cas, nous avons donc une période d'horloge avoisinant les  $60 \mu s$ , soit une commutation toute les  $30 \mu s$ . Sachant que le 8052 du Minitel exécute la plupart des instructions en un peu plus d'une microseconde, il dispose donc d'une vingtaine d'instructions en moyenne pour traiter chaque commutation et enregistrer ou émettre chaque bit. Certes, le nombre d'instructions permises pour le traitement de chaque bit semble un peu juste, mais c'est tout de même suffisant. Dans le cas d'une réception, une fois tous les bits du paquet de données enregistrés, le logiciel peut si nécessaire inhiber la communication PS/2 et dispose alors de tout le temps qu'il faut pour traiter ces données.

### 4.17.2 Double port PS/2

Nous décrivons ici une astuce qui fut utilisée par les fabricants de PC portable pour offrir deux ports PS/2 en un seul connecteur, réalisant par là même un gain de place.

Cette astuce est extrêmement simple. Le côté gauche de la figure 4.9 montre le brochage d'un port PS/2 standard : deux broches pour l'alimentation électrique, deux broches pour la communication, et deux broches inutilisées. Le côté droit de cette même figure montre le brochage d'un port PS/2 double : les deux broches de communication du deuxième port sont simplement relocalisées sur les deux broches auparavant inutilisées du premier port. Pour pouvoir utiliser les deux ports, il suffit de se procurer un simple adaptateur passif en Y, facilement trouvable sur des commerces en ligne.

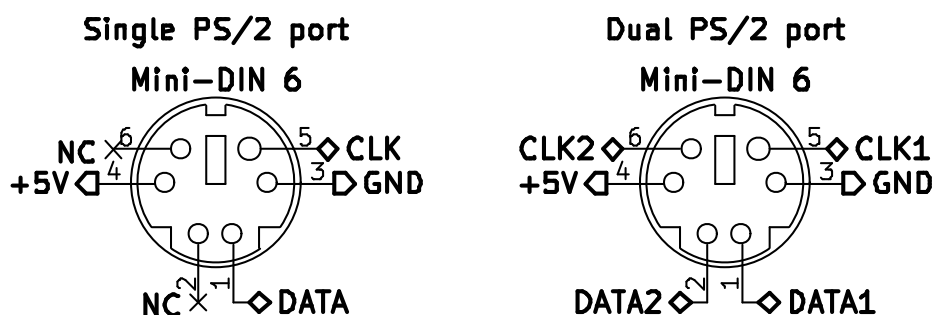


FIGURE 4.9 – Brochage du connecteur PS/2 vu de l'ordinateur

## 4.18 Contrôleur d'interruption

Afin d'éviter au microcontrôleur certaines opérations de *polling* (interrogation répétée), nous avons profité de la présence d'un CPLD pour implémenter un contrôleur d'interruptions. Ce dernier émet un signal d'interruption lorsque l'une de ses entrées au moins a changé d'état. Le microcontrôleur capte l'interruption et doit par la suite remettre à zéro le contrôleur d'interruption en écrivant à une adresse donnée. Un masque permet d'ignorer certaines entrées. Ce masque est accessible par un registre adressable en mémoire et peut donc être modifié par le logiciel.

La figure 4.10 montre le schéma logique du contrôleur d'interruption.

## 4.19 Connecteurs d'extension

Pour accroître la flexibilité du système, nous avons inclus deux connecteurs d'extension personnalisés (donc non standards) permettant le branchement de circuits ou de périphériques spécialement conçus.

### 4.19.1 Connecteur SPI

Un connecteur 16 broches donne accès au bus SPI, à quatre lignes de sélection d'esclave, à la ligne de réinitialisation du système, à une ligne d'interruption dédiée, à la masse et aux tensions d'alimentation 3,3 V et 5 V (brochage figure 4.11). C'est un connecteur IDC, destiné à recevoir un câble en nappe.

On remarquera que, du fait de leur finesse, les fils des câbles en nappe peuvent entraîner des chutes de tension non négligeables, augmentant avec le courant débité et la longueur du fil. Par exemple, si un périphérique est relié à l'appareil par un câble d'un mètre de long et de section AWG 26, et débite 500 mA sur le rail d'alimentation 3,3 V, la tension reçue par le périphérique est 3,17 V. Aussi, on veillera à ce que les périphériques tirant leur alimentation de ce connecteur ne consomment pas trop de courant et on utilisera des câbles courts.

Nous avons conçu les signaux et le brochage de ce connecteur de manière à permettre le chaînage de plusieurs périphériques (*daisy-chaining*).

Cela peut être fait en dotant chaque périphérique de deux connecteurs et en s'assurant que tous les périphériques respectent une certaine norme. Le premier connecteur sert à la liaison au Minitel si le périphérique considéré est le premier de la chaîne, ou à la liaison au périphérique précédent sinon. Le deuxième connecteur sert à la liaison au périphérique suivant, ou reste non connecté si le périphérique considéré est le dernier de la chaîne. Tous les périphériques doivent respecter les conditions suivantes.

- Chaque périphérique doit utiliser les lignes de sélection de plus bas indice. Par exemple, si un périphérique nécessite deux lignes de sélection, il doit utiliser  $\overline{CS0}$  et  $\overline{CS1}$ .
- Tous les signaux du premier connecteur sauf les lignes de sélections sont retransmis au deuxième connecteur en respectant le même brochage. Les lignes de sélection utilisées par le périphérique ne doivent pas être retransmises. Celles qui ne sont pas utilisées doivent être retransmises, mais en décalant leurs indices en commençant à  $\overline{CS0}$ .

Dans le cas de périphériques n'utilisant tous qu'une seule ligne de sélection, ce procédé permet le chaînage de quatre périphériques au plus.

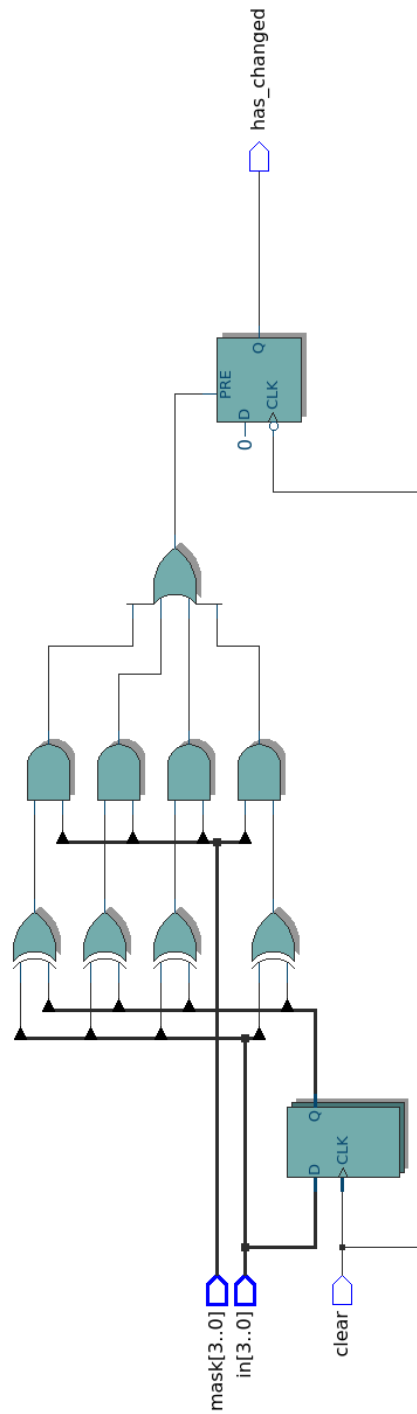


FIGURE 4.10 – Schéma logique du contrôleur d'interruption

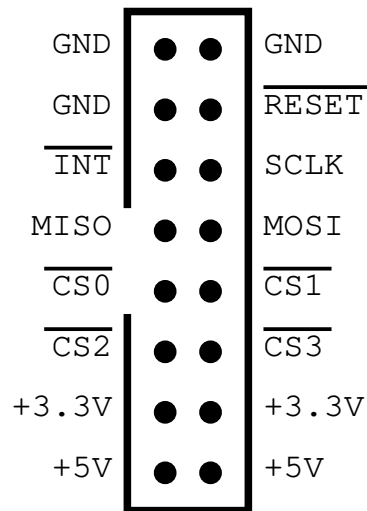


FIGURE 4.11 – Brochage du connecteur d’extension SPI

La figure 4.12 montre le routage des deux connecteurs dans le cas d’un périphérique n’utilisant qu’une seule ligne de sélection.

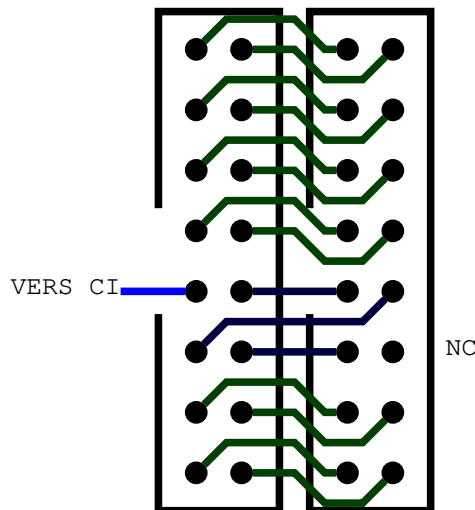


FIGURE 4.12 – Routage des connecteurs d’extension SPI pour le chaînage de périphériques

### 4.19.2 Connecteur audio-vidéo

Certaines anciennes consoles de jeu vidéo et micro-ordinateurs possèdent des entrées permettant à un matériel extérieur (cartouche de jeu ou périphérique) d’insérer un signal audio ou vidéo en addition de celui généré par la console. Puisqu’une telle fonction était faisable pour notre projet sans demander beaucoup d’effort, nous l’avons réalisée.

La carte M5 est dotée d'un connecteur standard Mini-DIN 9 broches dont le brochage est donné par la figure 4.13.

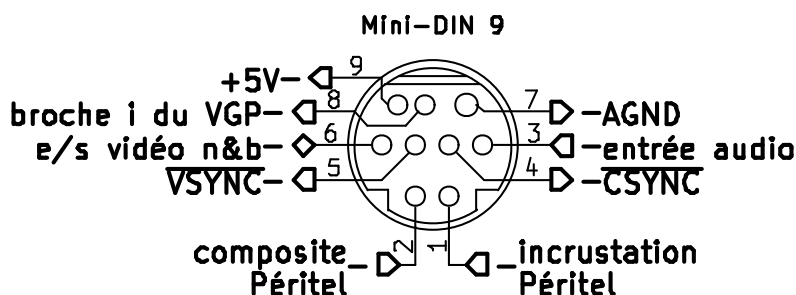


FIGURE 4.13 – Brochage du connecteur d'extension audio-vidéo

Un périphérique peut notamment, par le biais de ce connecteur :

- injecter un signal audio qui sera mélangé dans la sortie son,
- injecter un signal vidéo en niveaux de gris qui sera incrusté sur l'écran du Minitel (il peut le faire, par exemple, pendant toute la durée de la trame vidéo, ou seulement sur quelques lignes, ou encore seulement certains pixels),
- injecter un signal vidéo composite couleur sur la Périitel et l'incruster sur l'écran du téléviseur (même remarque que pour le point précédent).

## 4.20 Périitel

Sur la carte UT se trouve une rangée de huit trous offrant entre autres les signaux  $R$ ,  $G$ ,  $B$  et  $\overline{CSYNC}$  du VGP. On pourra remarquer que le cache en plastique à l'arrière de l'appareil présente un profil de découpe en forme de connecteur Périitel, témoignant de la possibilité de pourvoir l'appareil d'une sortie vidéo Périitel.

Nous avons soudé une barrette de broches sur ces huit trous et nous avons réalisé une petite carte qui vient s'enficher sur ce connecteur et qui offre un connecteur Périitel. Cette carte prend les sorties du VGP, qui sont aux niveaux TTL, et les convertit aux niveaux de la norme Périitel tout en étant capable de fournir assez de courant pour piloter une charge de  $75\ \Omega$ .

Entrée	Tension crête-à-crête	Impédance
R, G, B	0,7 V	$75\ \Omega$
composite	synchro 0,3 V, vidéo 1 V	$75\ \Omega$
commutation rapide	composite 0-0,4 V, RGB 1-3 V	$75\ \Omega$

Tensions et impédances des entrées vidéo Périitel

Pour réaliser le convertisseur TTL-Périitel, nous avons choisi le circuit intégré 74ACT241 qui contient essentiellement 8 tampons, que nous faisons suivre d'un pont diviseur de tension résistif qui sert à la fois d'adaptateur de tension et d'adaptateur d'impédance. La famille 74ACT est caractérisée par ses courants de sortie élevés.<sup>3</sup> La *datasheet* du SN74ACT241 de Texas Instruments

3. Texas Instruments. *Logic Guide 2014 (Rev. AA)*, page 10.  
<http://www.ti.com/lit/sg/sdyu001aa/sdyu001aa.pdf>



donne pour courant continu maximal par sortie recommandé  $I_O = \pm 24 \text{ mA}$ , et pour courant continu maximal absolu dans  $V_{CC}$  ou  $GND$ ,  $\pm 200 \text{ mA}$ . Ce sont des valeurs sensiblement plus élevées que pour des familles plus conventionnelles telles que les 74HCT. En extrapolant à partir des informations données par la *datasheet*, on peut considérer que la tension de sortie minimale à l'état haut, avec  $V_{CC} = 5 \text{ V}$  et  $I_{OH} = -24 \text{ mA}$ , est  $V_{OH,min} = 4,26 \text{ V}$ . Cette valeur  $V_{OH,min}$  est aussi bien plus élevée que celle de la famille 74HCT qui se situe généralement autour de  $2,4 \text{ V}$ .

# Chapitre 5

## Écueils

À différents stades de la réalisation du projet, nous avons rencontré quelques problèmes. Nous les énumérons ici et donnons, lorsque cela a été fait, la contre-mesure qui a été prise.

### 5.1 Méthodologie

Dans le monde de l'industrie, la réalisation d'un projet commence généralement par la définition claire et précise des besoins auxquels doit répondre le produit et des contraintes qu'il doit respecter. Puis l'on décide des techniques et des moyens qui seront employés pour réaliser un produit répondant au cahier des charges. C'est seulement ensuite que peut commencer la conception proprement dite. Si l'analyse des besoins et des fonctions a été correctement effectuée, la conception doit normalement pouvoir se faire sans accroc et il ne sera jamais nécessaire de revenir en arrière sur les fonctions ou les contraintes.

Dans notre cas, cette méthodologie n'a pas été suivie, ce qui s'est traduit par un retard très important dans la réalisation du projet.

Nous avons bien commencé à travailler en respectant cette méthodologie, mais tout au long du travail sur le projet nous avons eu d'autres idées techniques et d'autres fonctionnalités que nous avons absolument voulu implémenter. Plutôt que de s'en tenir au cahier des charges élaboré au début du projet, nous avons sans cesse ajouté, supprimé ou modifié certains aspects du produit. Au final le projet a tout de même été mené à bien, mais non sans une considérable perte de temps.

### 5.2 Mécanique

Puisque nous créons un produit destiné à être assemblé à l'intérieur d'un appareil déjà existant, il fallait bien entendu que les dimensions et la forme du produit soient adaptées au boîtier de l'appareil.

Malgré tous les soins que nous avons accordé aux mesures de la coque de l'appareil, nous avons commis quelques erreurs. Concrètement, cela se traduit par un connecteur Péritel peu accessible, et par le fait que nous n'avons pas réussi à placer le petit circuit imprimé qui contient les boutons poussoirs, le connecteur pour casque et les potentiomètres de réglages, là où nous le souhaitions.

Pour l’instant nous nous sommes contentés de caler ce circuit imprimé tant bien que mal derrière le connecteur PériTel. Nous tâcherons à l’avenir de trouver un moyen de fixation plus honorable. De toute façon, cela affecte seulement l’aspect esthétique et non le bon fonctionnement de l’appareil.

### 5.3 Freins thermiques

En conception de circuit imprimé, lorsqu’une pastille est physiquement connectée à une grande zone de cuivre (par exemple un plan de masse, ou une piste très large), il est d’usage de relier la pastille à la zone de cuivre non pas directement, mais par l’intermédiaire d’un frein thermique (*thermal relief* dans la terminologie anglophone), c’est-à-dire par une ou plusieurs pistes fines, tel qu’illustré par la figure 5.1. La raison en est de faciliter le soudage des composants sur une telle pastille. En effet, en l’absence de frein thermique, la chaleur transférée par le fer à souder sur la pastille est en grande partie évacuée vers la zone de cuivre. La pastille est alors difficile à chauffer et le soudage demande beaucoup d’insistance, quand il n’est pas même impossible.

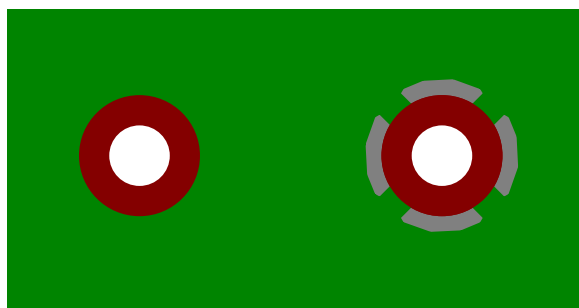
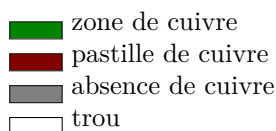


FIGURE 5.1 – Exemple de pastilles avec (droite) et sans (gauche) frein thermique



De manière tout à fait arbitraire et sans le moindre appui scientifique (!), la largeur de piste de frein thermique proposée par défaut par notre logiciel de CAO nous a semblé quelque peu petite et il nous a semblé bon de l’augmenter (nous sommes passé de 0,25 mm à 0,5 mm). Mal nous en a pris ! Nous avons dû nous acharner sur ces pastilles avec le fer à souder de longs instants pour parvenir à réaliser la moindre soudure. La leçon est retenue !

### 5.4 Bruit sur la sortie audio

En approchant l’oreille du haut-parleur, ou en augmentant le volume, on peut entendre un bourdonnement dans les graves. Nous ignorons si ce bruit a pour cause la conception maladroite de la section audio du circuit, ou s’il est dû aux bruits électromagnétiques environnant (fréquence secteur 50 Hz peut-être ?).

Peut-être qu’un condensateur électrolytique de valeur élevée inséré juste derrière le haut-parleur,

formant avec la résistance de celui-ci un filtre passe-haut, suffirait à atténuer le phénomène. Nous n'avons pas encore essayé cette idée.

## 5.5 Précision et sensibilité du crayon optique

L'utilisation du crayon optique sur notre système est sujette aux mêmes défauts que les micro-ordinateurs d'autrefois, d'après ce que nous avons pu lire sur internet : le positionnement du crayon optique était plutôt imprécis, et il fallait souvent pousser la luminosité du moniteur à son maximum.

En fait, sur notre système, c'est une question de compromis. Lorsque l'on règle la sensibilité au minimum fonctionnel, la précision est excellente (en maintenant le crayon optique immobile sur une position donnée de l'écran, les coordonnées verticales et horizontales lues par le module crayon optique du CPLD présentent une oscillation de  $\pm 1$  seulement) mais il faut régler la luminosité de l'écran au maximum et la détection du signal du phototransistor n'a lieu que si le crayon optique est posé sur une zone blanche ou gris très clair de l'écran ; on ne peut pas l'utiliser sur des teintes plus foncées car elles n'émettent pas assez de lumière. Lorsque l'on augmente la sensibilité de façon à ce que le crayon optique fonctionne sur les teintes de gris très foncé, alors la précision horizontale est exécrable (oscillations de  $\pm 10$  pour un certain réglage, donc un intervalle de taille 20, ce qui représente un sixième de la largeur d'écran sur laquelle le VGP peut afficher du texte).

N'ayant pas d'oscilloscope à disposition, nous ne sommes pas en mesure de visualiser le signal du phototransistor dans son exactitude et, par conséquent, nous ne pouvons que difficilement suggérer des solutions. Quoi qu'il en soit, celles-ci peuvent être de trois types :

- matérielle : concevoir un circuit électronique différent pour le traitement du signal ;
- logique programmable : modifier le module du CPLD relatif au crayon optique pour, par exemple, ajouter un filtre numérique ou un algorithme suffisamment simple pour être réalisé avec des circuits logiques ;
- logiciel : au lieu que le programme ne fasse que lire les coordonnées du crayon données par le CPLD, il pourrait implémenter un algorithme traitant ces coordonnées sur le temps afin d'en calculer de plus précises.

En supposant que les coordonnées calculées par le CPLD oscillent de manière équiprobable dans le temps autour de la vraie position du crayon et avec des amplitudes à gauche et à droite identiques ou presque, on pourrait alors programmer le logiciel pour que celui-ci calcule une moyenne des positions lues sur un certain intervalle de temps. Un inconvénient est le temps de réaction non négligeable que cette méthode induit. Par exemple, admettons que la moyenne doit être calculée sur 10 échantillons pour être représentative de la vraie position. Chaque échantillon ayant lieu pendant une trame vidéo qui lui est propre, et chaque trame ayant une durée de 20 *ms*, il faudrait alors 0,2 *s* après un grand déplacement du crayon pour que le logiciel détermine sa nouvelle position. C'est un délai suffisamment élevé pour être remarqué par l'utilisateur et ainsi gêner la souplesse d'utilisation du système.

## 5.6 Plan de masse

Le circuit imprimé que nous avons réalisé est à deux couches. Conformément aux pratiques standards, nous avons réservé l'une des couches comme plan de masse.

Toutefois, le routage de tous les signaux sur une seule couche s'avérant impossible, nous avons utilisé des *vias* et fait passer quelques pistes de signaux sur la couche du plan de masse. Bien que nous nous sommes efforcé d'avoir recours à ce procédé aussi peu que possible, à la fin de la conception du circuit imprimé le plan de masse était zébré de pistes (voir annexe D).

Pour un circuit comme celui-ci, fonctionnant à fréquence relativement basse (12 MHz), ce n'est pas critique. Nous ne nous attarderons pas ici sur les détails de la théorie mais, dans l'absolu, il est fortement déconseillé de couper le plan de masse par des pistes car cela perturbe les chemins empruntés par les courants de retour au sein du plan de masse et a pour conséquences une augmentation du bruit superposé sur les lignes de signaux et d'alimentation, et une augmentation des bruits électromagnétiques émis par le système.

Encore une fois, ce sont des considérations qui s'appliquent surtout aux systèmes modernes à fréquences élevées, ou aux appareils de mesure de précision.

Dans notre cas, une meilleure pratique aurait été de router les signaux difficiles par des fils cavaliers (comme c'est d'ailleurs le cas sur la carte UT du Minitel), laissant ainsi le plan de masse presque intact.

## Chapitre 6

# Conclusion

Le produit final est entièrement fonctionnel. Si les sous-systèmes audio et crayon optique présentent chacun des imperfections (voir 5.4 et 5.5), celles-ci sont purement d'ordre qualitatif (qualité du son, précision du crayon optique). Elles ne remettent nullement en cause le bon fonctionnement de l'ensemble ou les fondements logiques du sous-système correspondant, et peuvent faire l'objet d'améliorations dans le cadre d'une hypothétique conception future.

Sur un plan personnel, ce projet fut extrêmement enrichissant.

Avant tout car son aboutissement représente déjà une petite réussite personnelle. Même si les problèmes de méthodologie évoqués plus tôt (voir 5.1) associés à une motivation en dents de scie ont été un frein et ne m'ont pas permis d'achever le projet aussi vite que je l'aurai souhaité, je suis satisfait d'avoir tout de même pu le mener à bien.

Concernant les connaissances et le savoir-faire technique, j'en ressors beaucoup plus expérimenté.

Ce projet fut premièrement un prétexte pour m'essayer à la logique programmable, un aspect majeur et omniprésent de l'électronique moderne. J'ai pu expérimenter avec un CPLD, m'initier au fonctionnement général de ce genre de composant, appréhender ses atouts comme ses limites. Les notes d'application du fabricant se sont une fois de plus avérés être une mine d'or d'informations. L'apprentissage du langage Verilog, de ses pièges aussi, fut très formateur. Aussi, c'est en expérimentant avec le CPLD et les outils de simulation associés que j'ai pu vraiment comprendre les dangers de la logique asynchrone et pourquoi tous les experts et professeurs du domaine insistent sur l'usage quasi-systématique de la logique synchrone.

J'ai également pu m'initier à la communication JTAG et au système de test d'interconnexions BSC, encore deux éléments omniprésents dans l'électronique et dans l'industrie.

Dans le cadre de ce projet j'ai aussi dû m'intéresser de près aux différentes techniques de traduction de niveaux de tension, les avantages et les inconvénients de chacune d'entre elles et les cas dans lesquels elles sont adaptées ou non.

Enfin, ce fut la première fois que je concevais un circuit imprimé et ce fut une expérience particulièrement enrichissante. Même si celui-ci est sans doute perfectible, j'ai pu acquérir les techniques élémentaires de conception, connaître quelques pièges à éviter et apprendre les contraintes im-

posées par les fabricants.

Ce travail aura été l'occasion de mettre mes connaissances en pratique, mais aussi d'en acquérir de nouvelles. Il m'aura également permis d'appréhender le processus de réalisation d'un système du début à la fin.

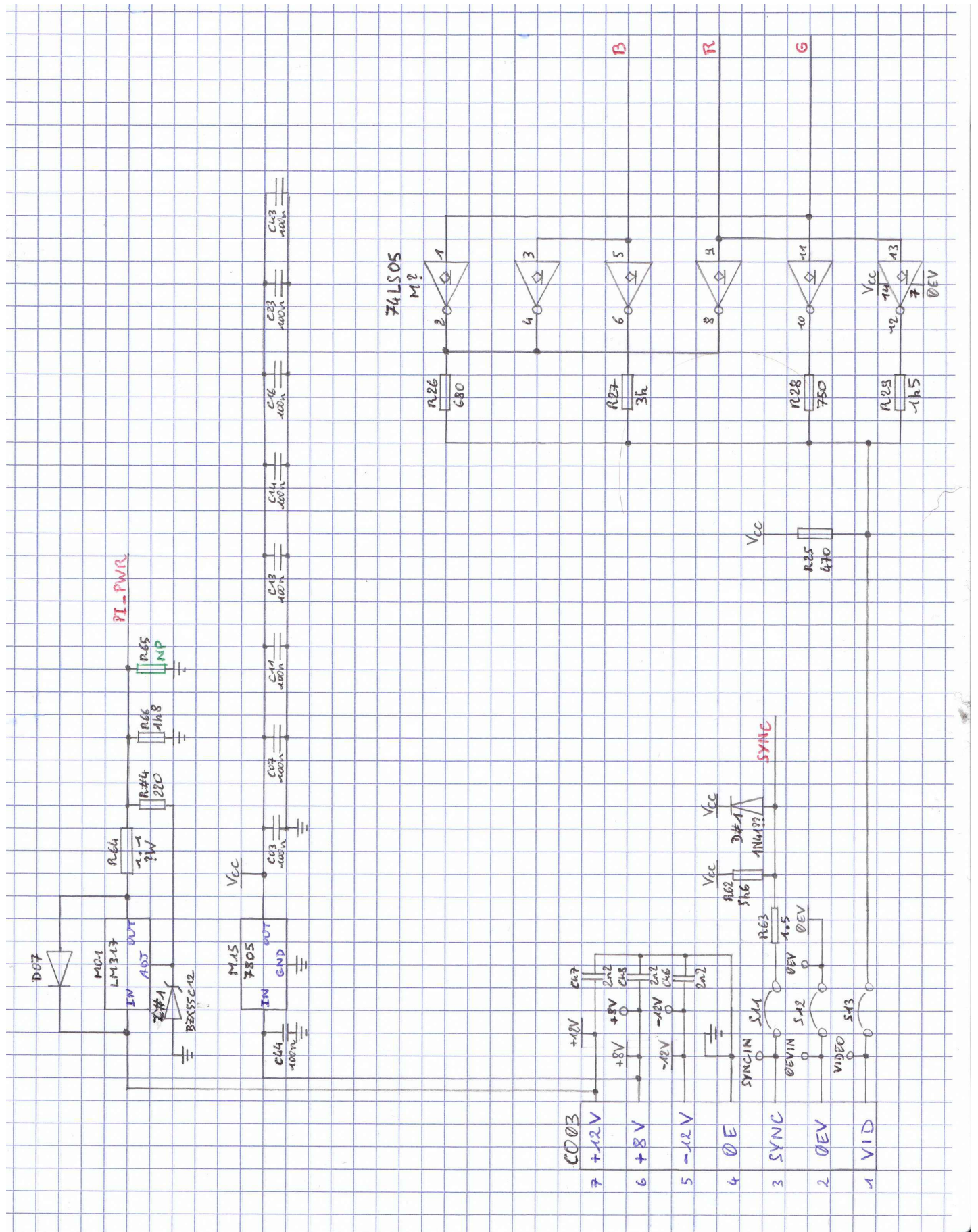
# Annexes



## Annexe A

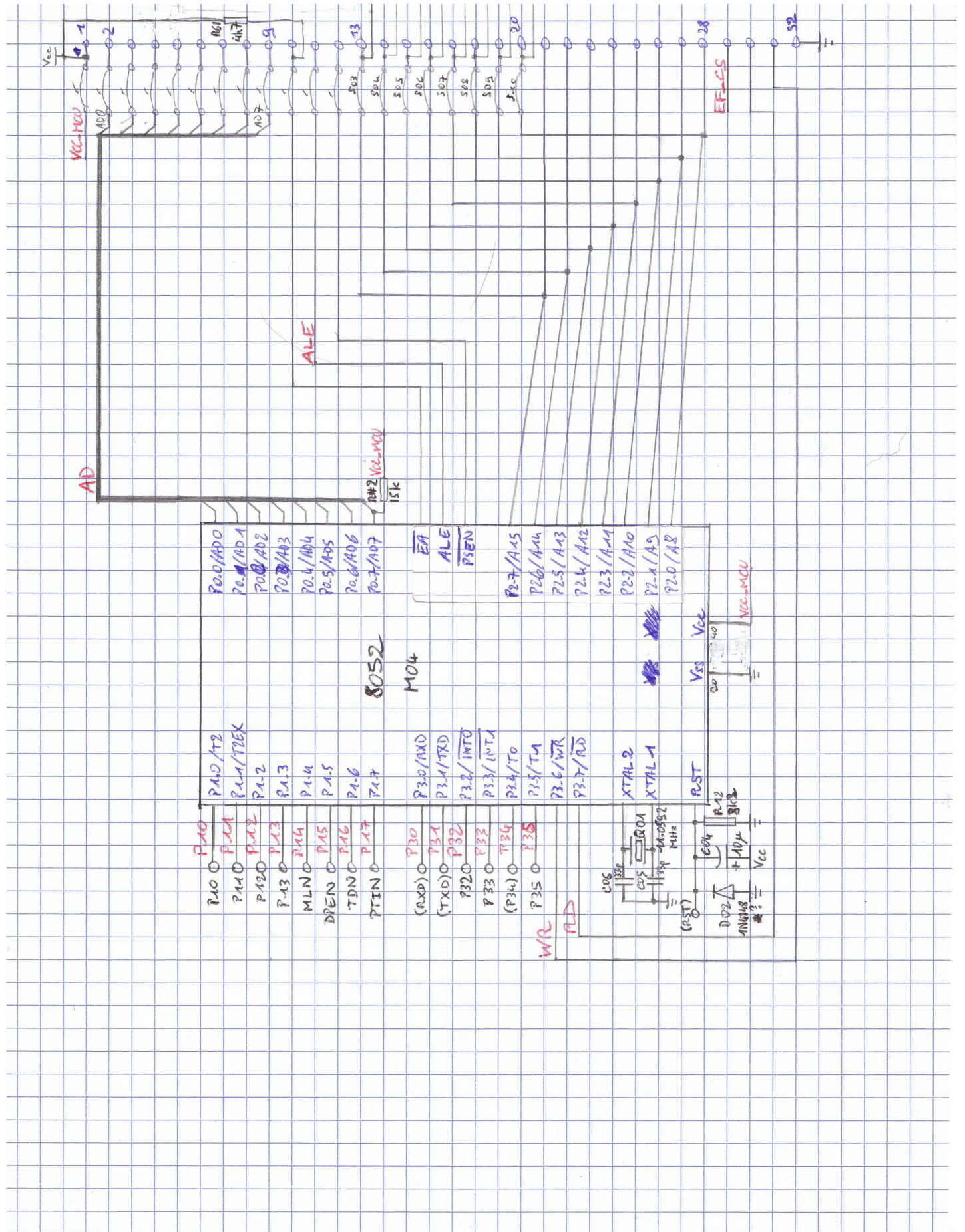
# Schéma électrique de la carte UT du Minitel 1B Téléc-Alcatel

Alimentation et conversion RGB-niveaux de gris



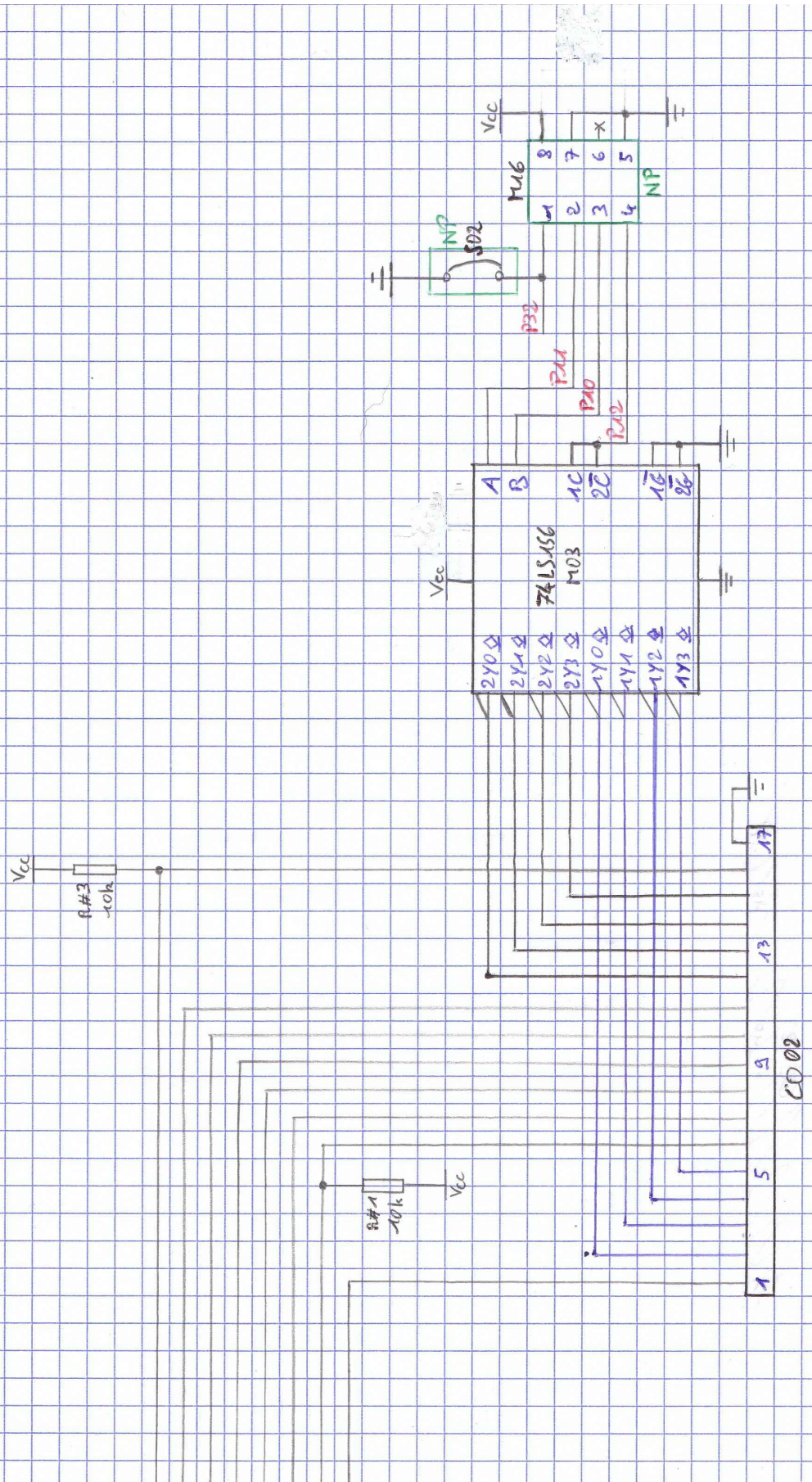


Microcontrôleur, clavier et connecteur d'extension – Feuille gauche



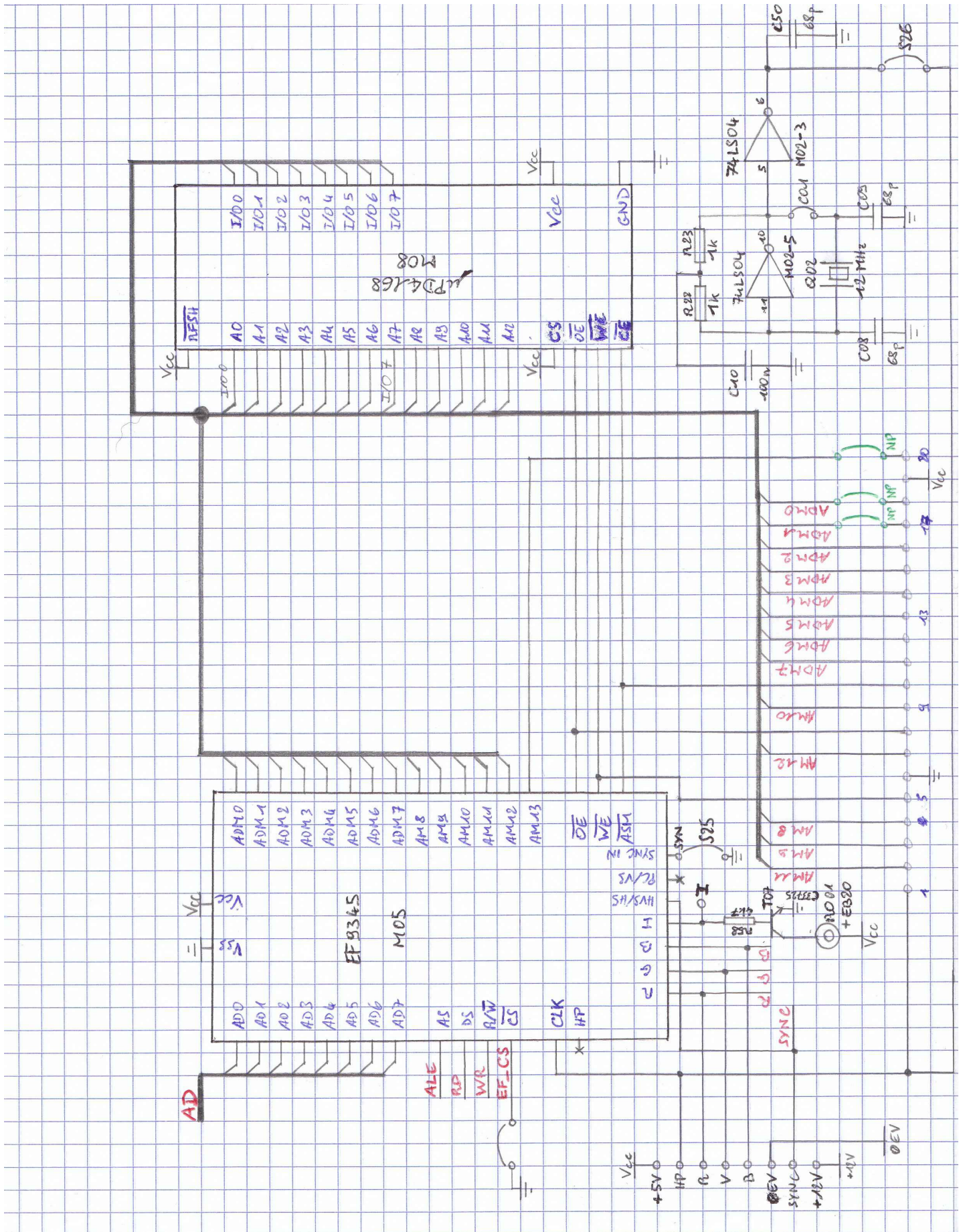


Microcontrôleur, clavier et connecteur d'extension – Feuille droite



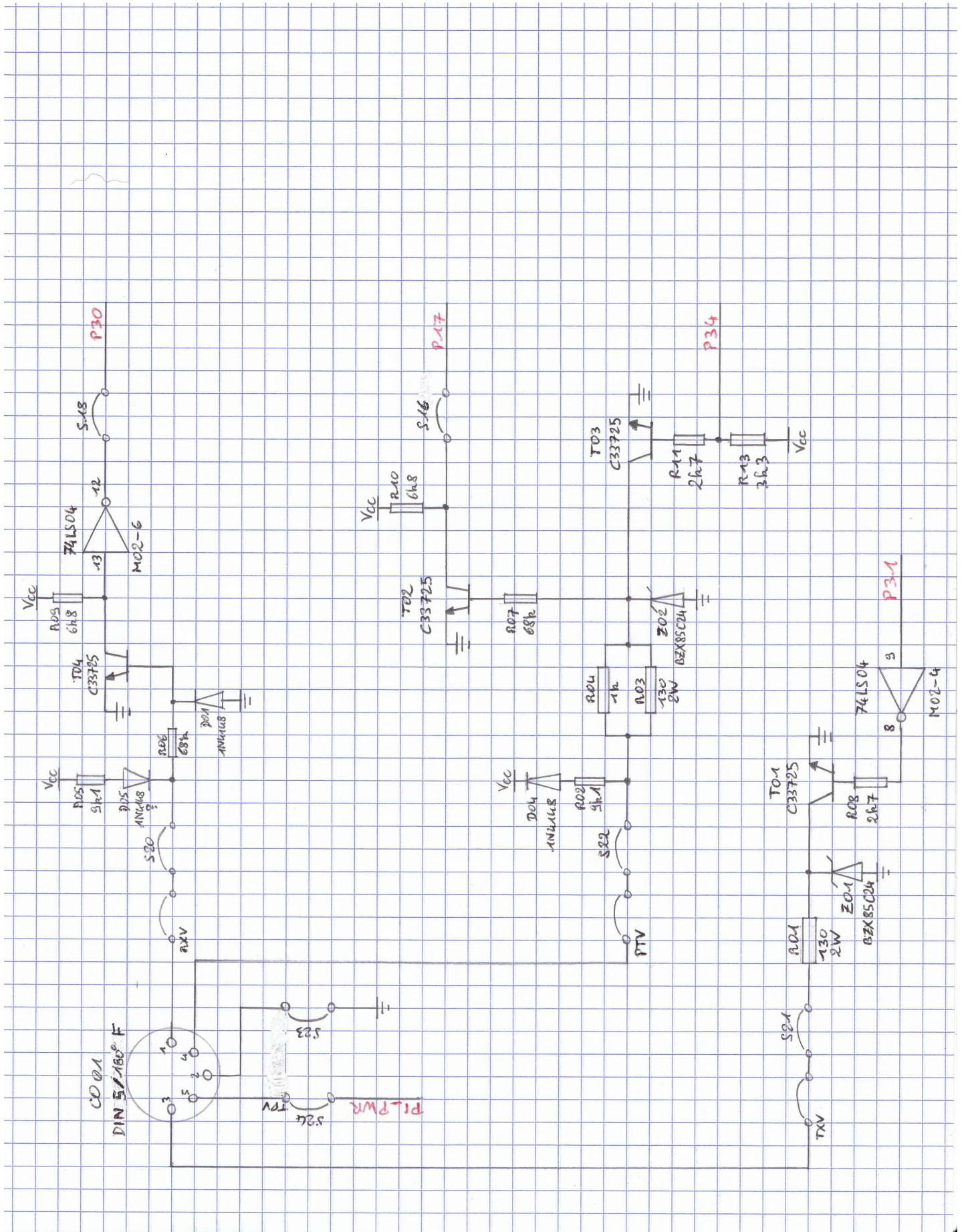


Graphismes



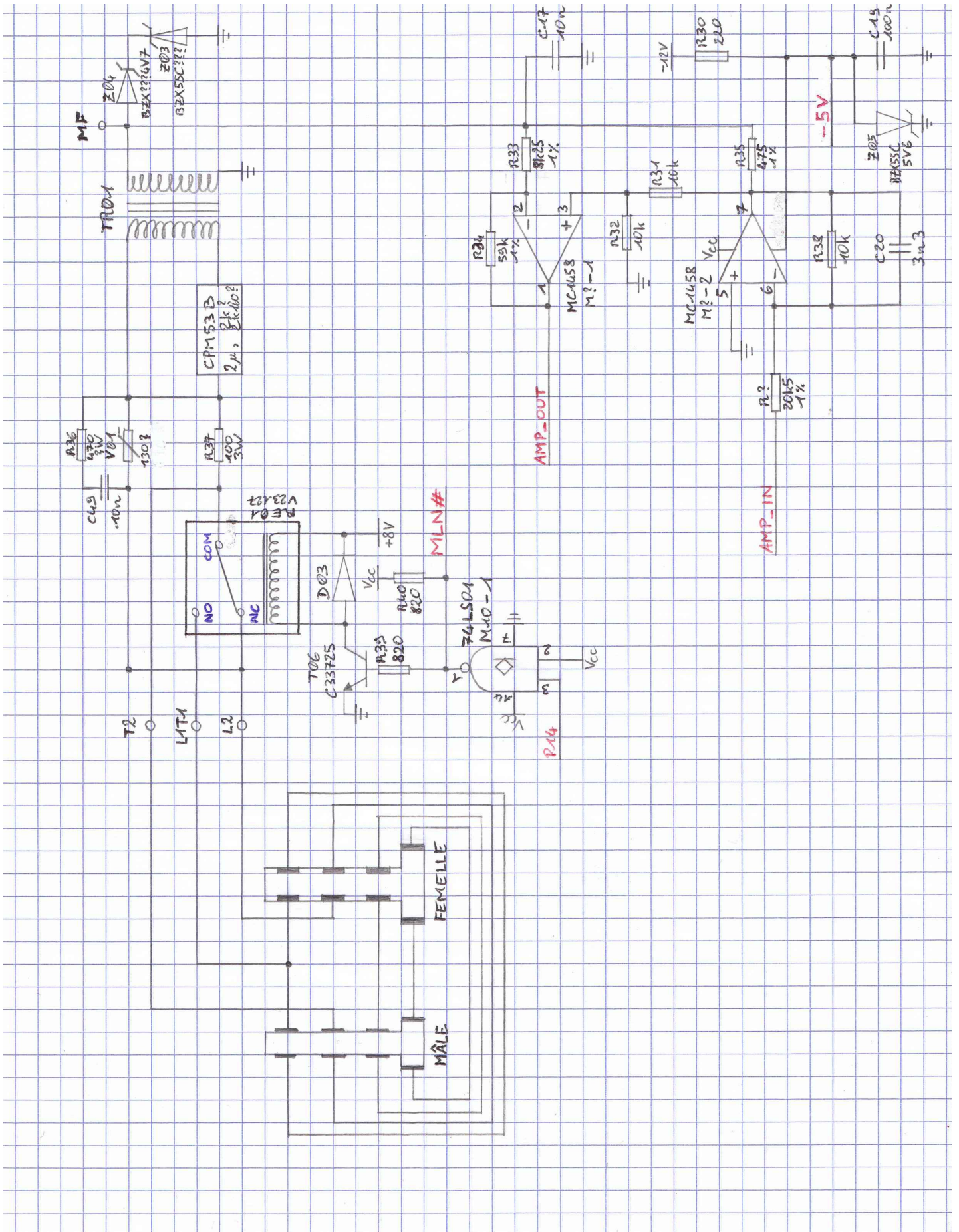


Interface prise péri-informatique



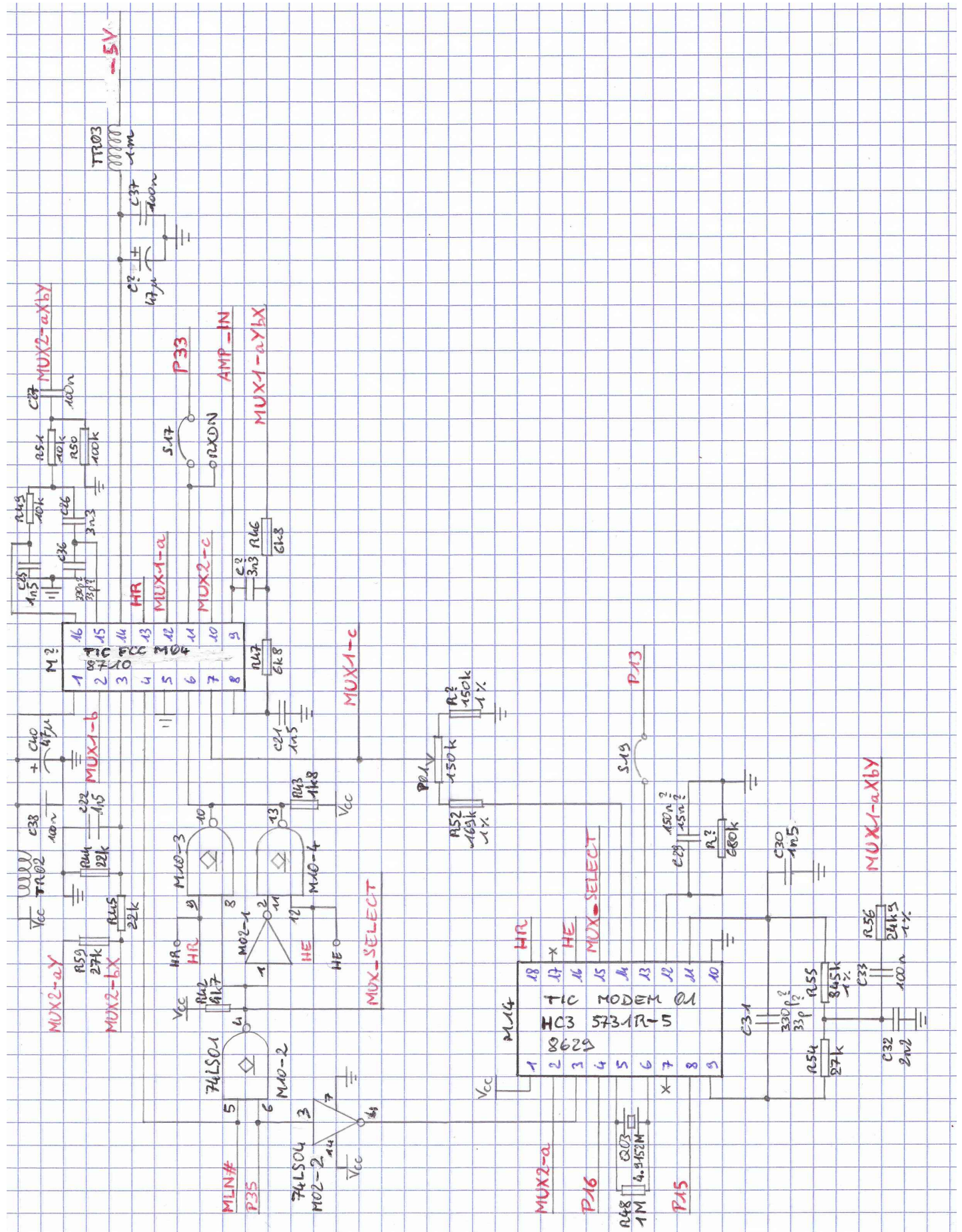


Interface réseau téléphonique



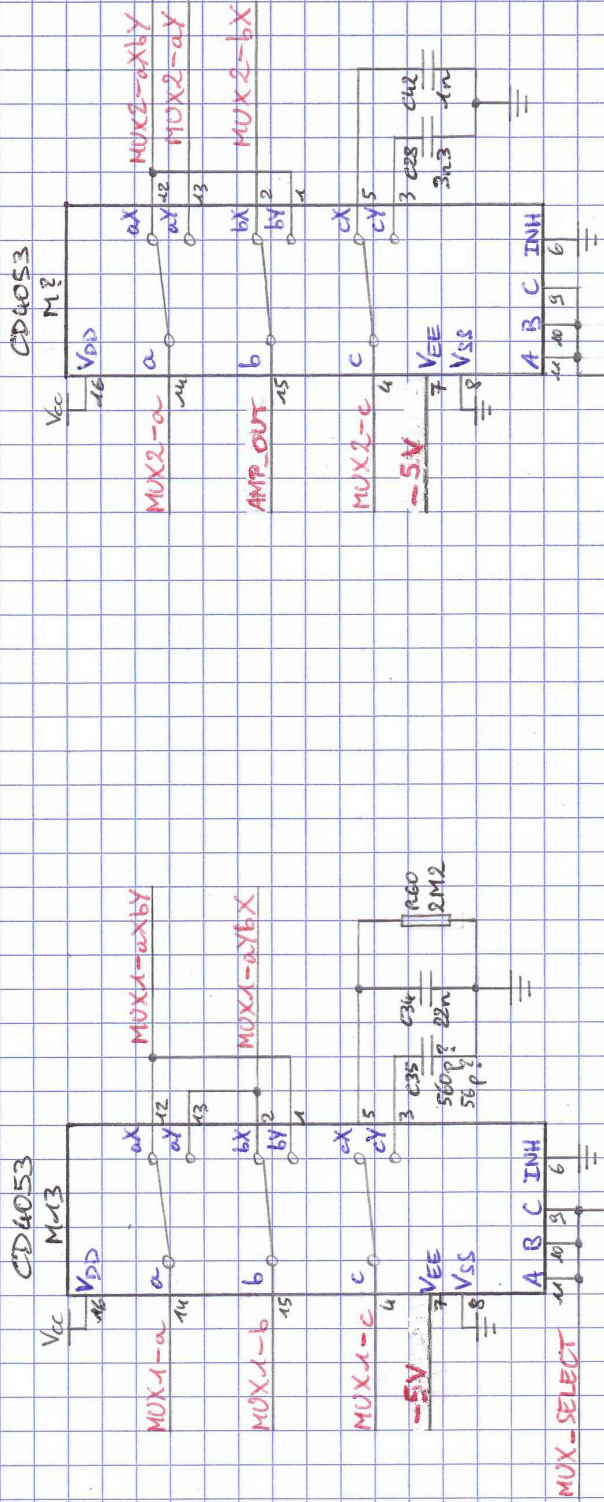


Modem



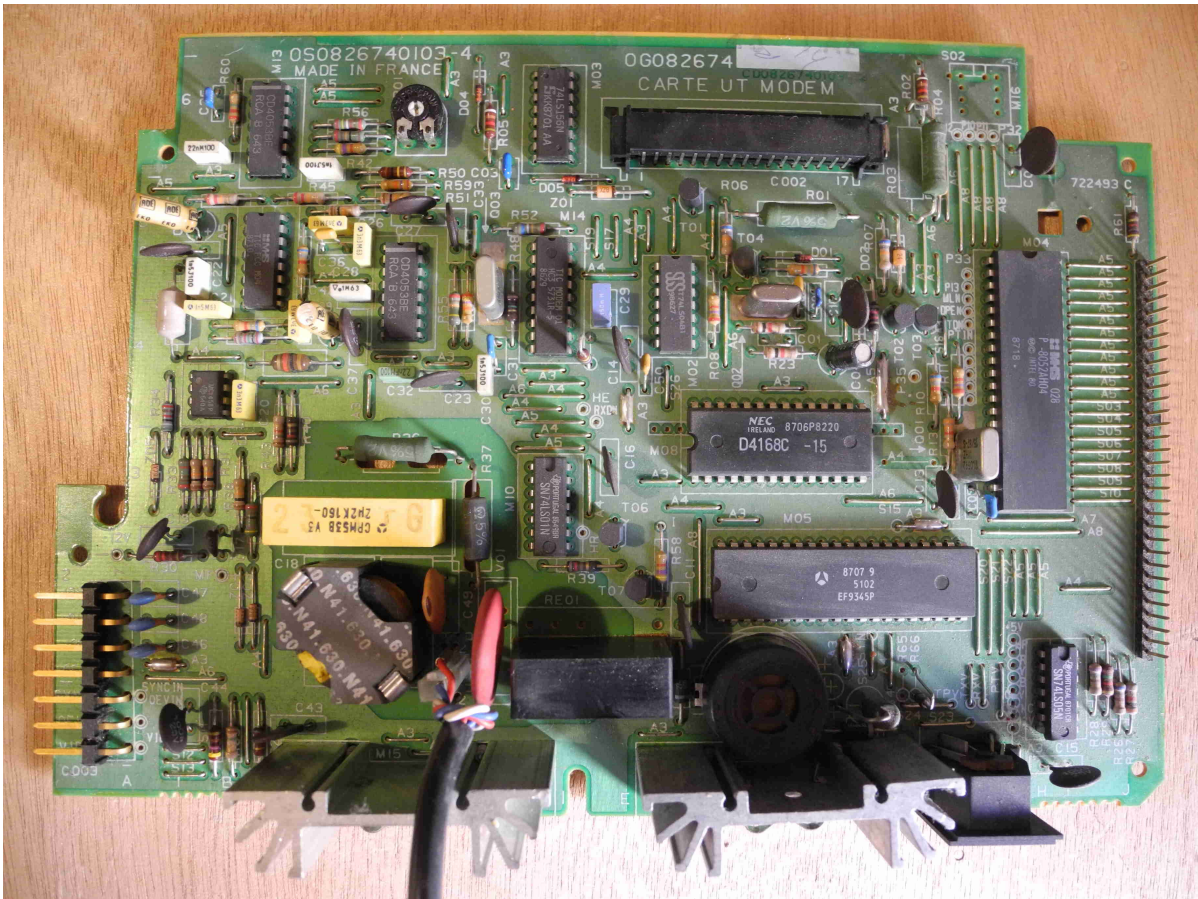


# Multiplexage modem



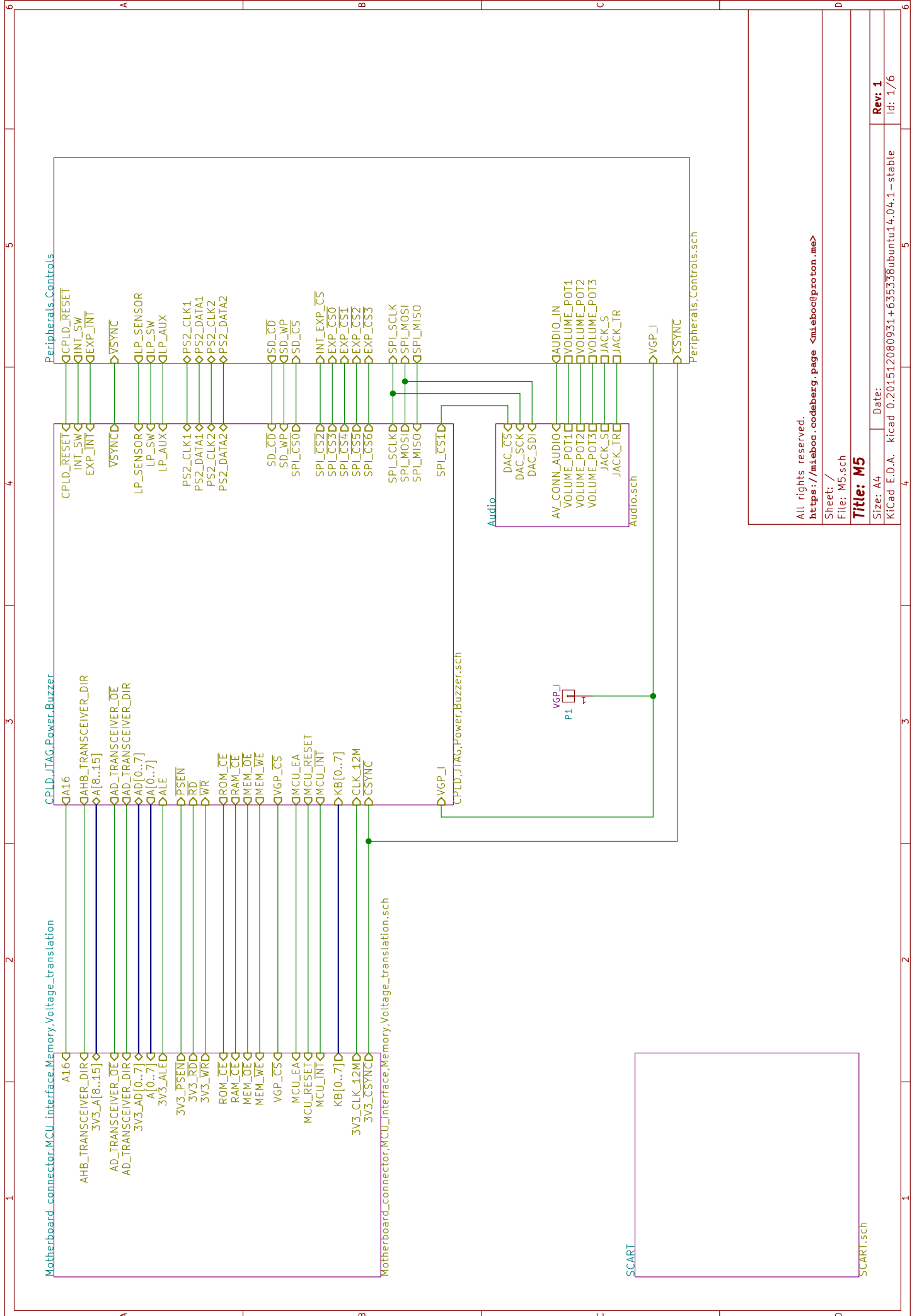
## Annexe B

# Photographie du dessus de la carte UT



## Annexe C

# Schéma électrique de la carte M5



All rights reserved.  
<https://mieboc.codeberg.page> <mieboc@proton.me>

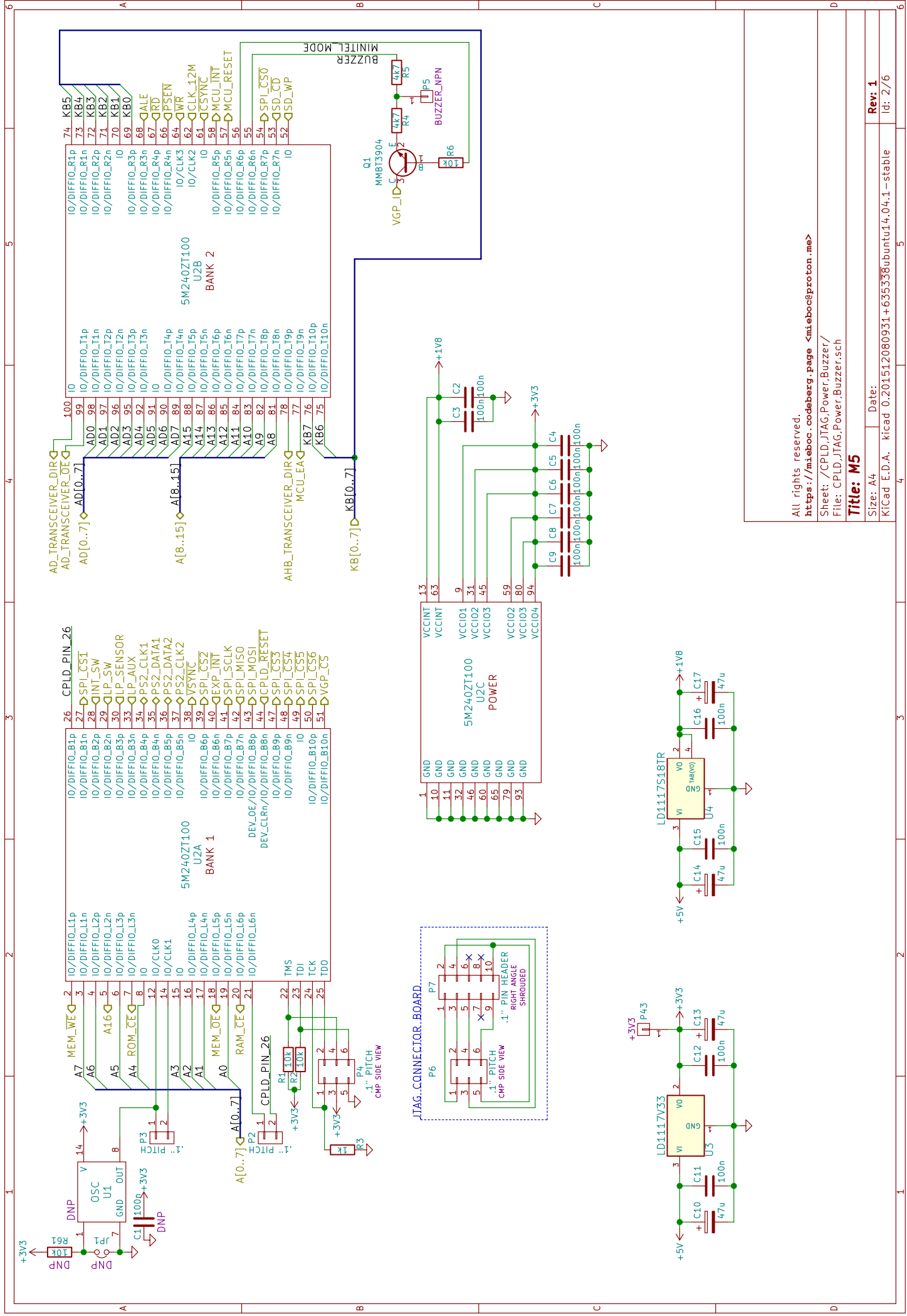
Sheet: /  
 File: M5.sch

**Title: M5**

Size: A4 Date:  
 KiCad E.D.A. kicad 0.201512080931+635338ubuntu14.04.1 - stable

Rev: 1  
 Id: 1/6



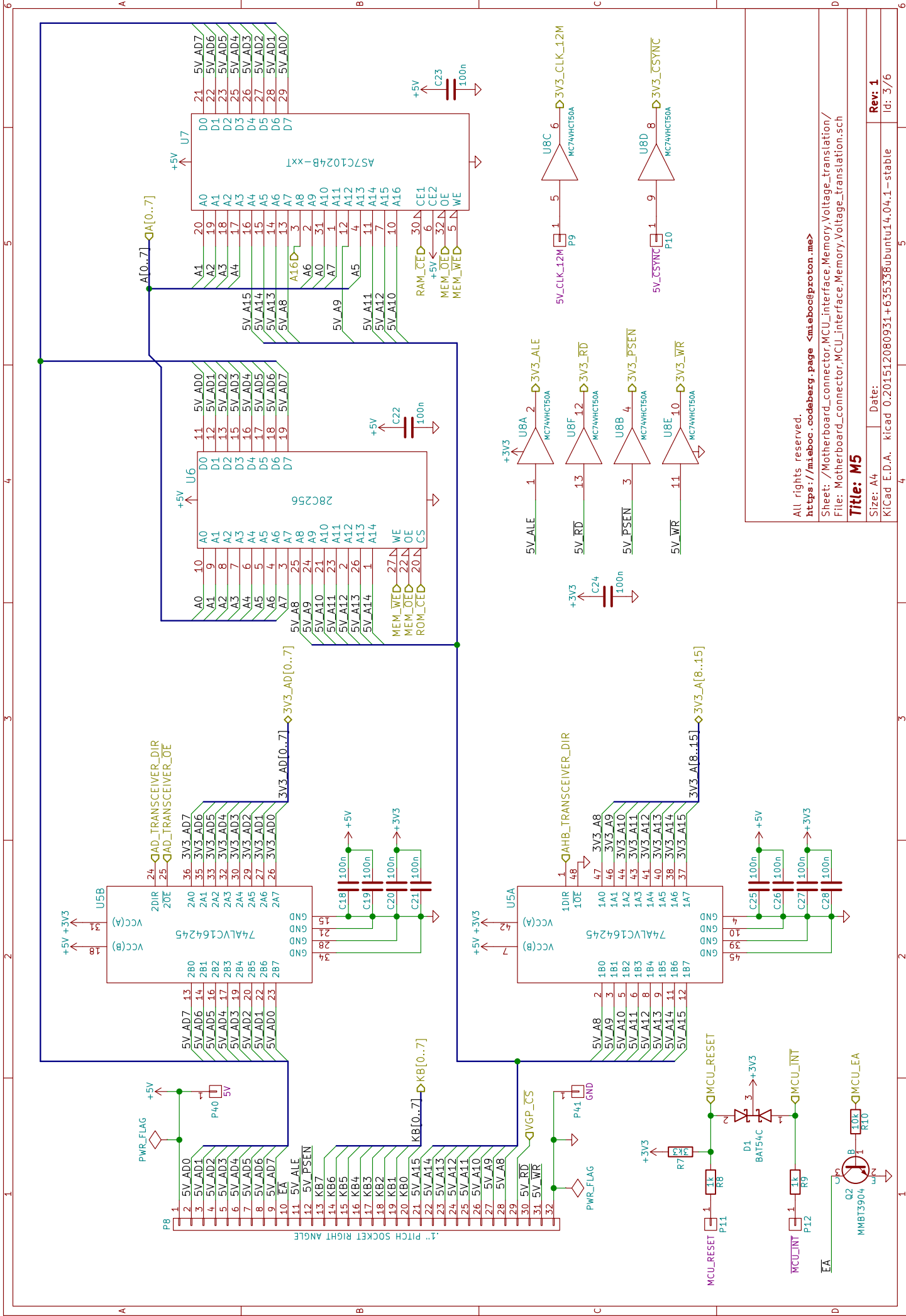


All rights reserved.  
<https://mieboc.codeberg.page> <mieboc@proton.me>

Sheet: /CPLD\_JTAG.Power.Buzzer/  
 File: CPLD\_JTAG.Power.Buzzer.sch

**Title: M5**

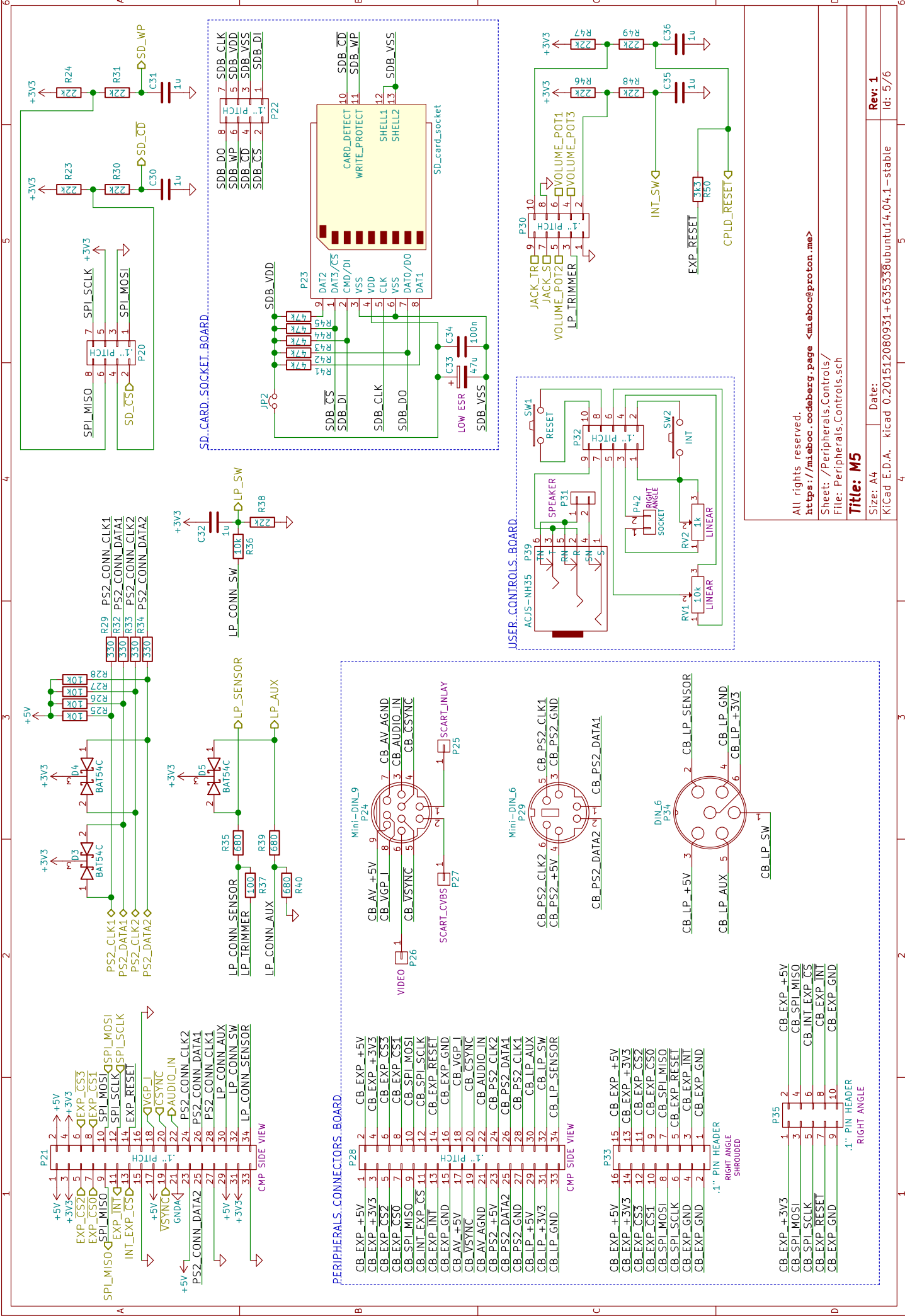
Size: A4	Date:
KiCad E.D.A. kicad 0.201512080931+635338ubuntu14.04.1 - stable	
Rev: 1	
Id: 2/6	



All rights reserved.  
<https://mieboc.codeberg.page> <mieboc@proton.me>  
 Sheet: /Motherboard\_connector,MCU\_interface,Memory,Voltage\_translation/  
 File: Motherboard\_connector,MCU\_interface,Memory,Voltage\_translation.sch

**Title: M5**

Size: A4	Date:
KiCad E.D.A.	kiCad 0.201512080931+635338ubuntu14.04.1-stable
Rev: 1	Id: 3/6



All rights reserved.  
<https://miebob.codeberg.page> <miebob@proton.me>  
 Sheet: /Peripherals,Controls/  
 File: Peripherals,Controls.sch

**Title: M5**

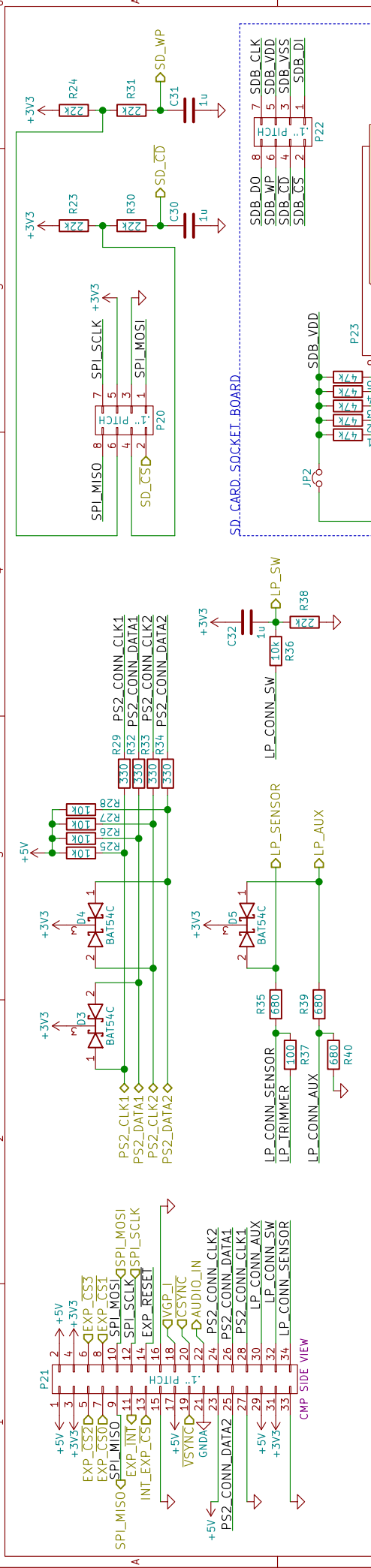
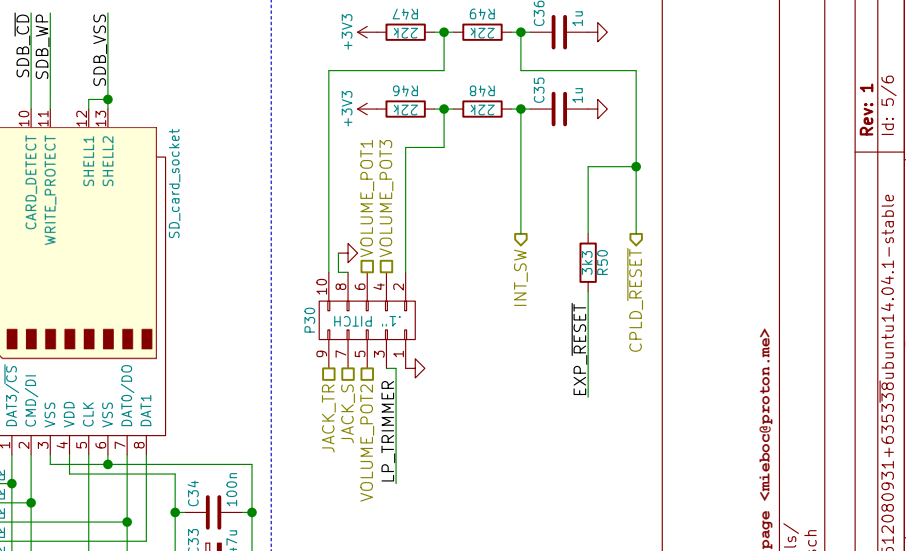
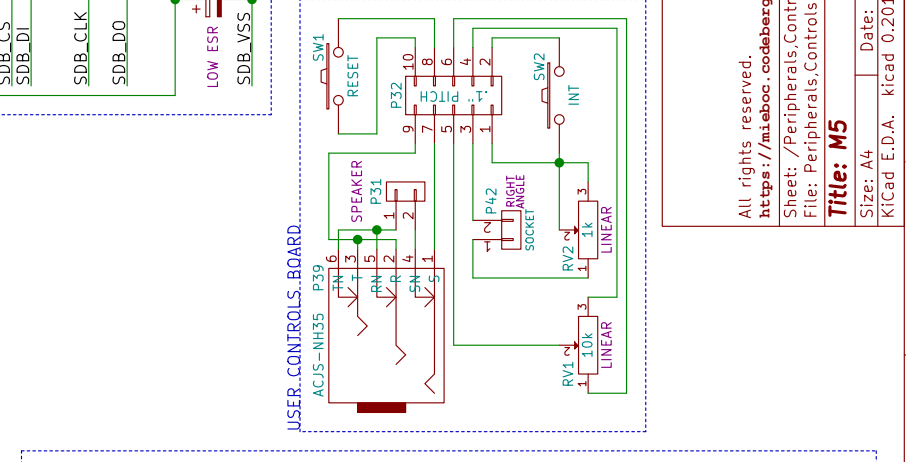
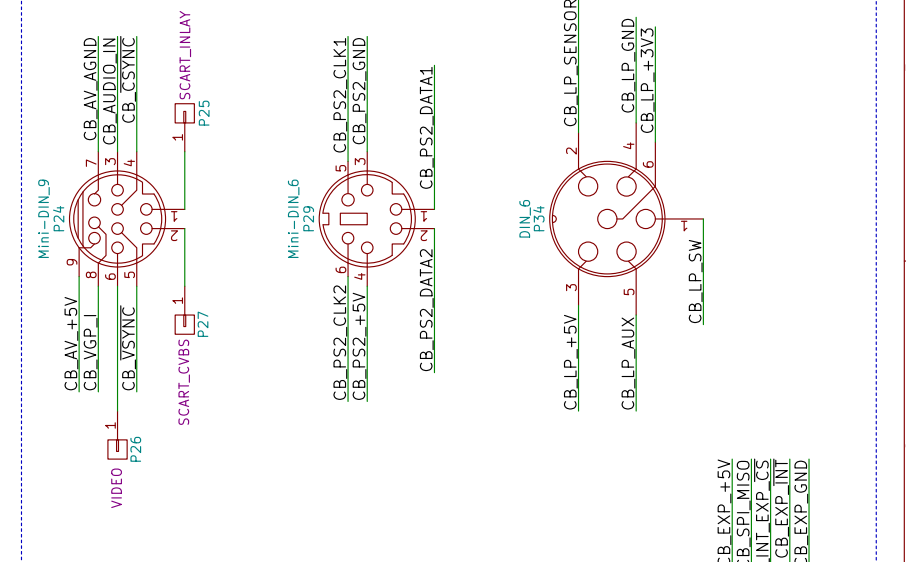
Size: A4 Date: \_\_\_\_\_  
 KiCad E.D.A. kicad 0.201512080931+635338ubuntu14.04.1-stable Id: 5/6

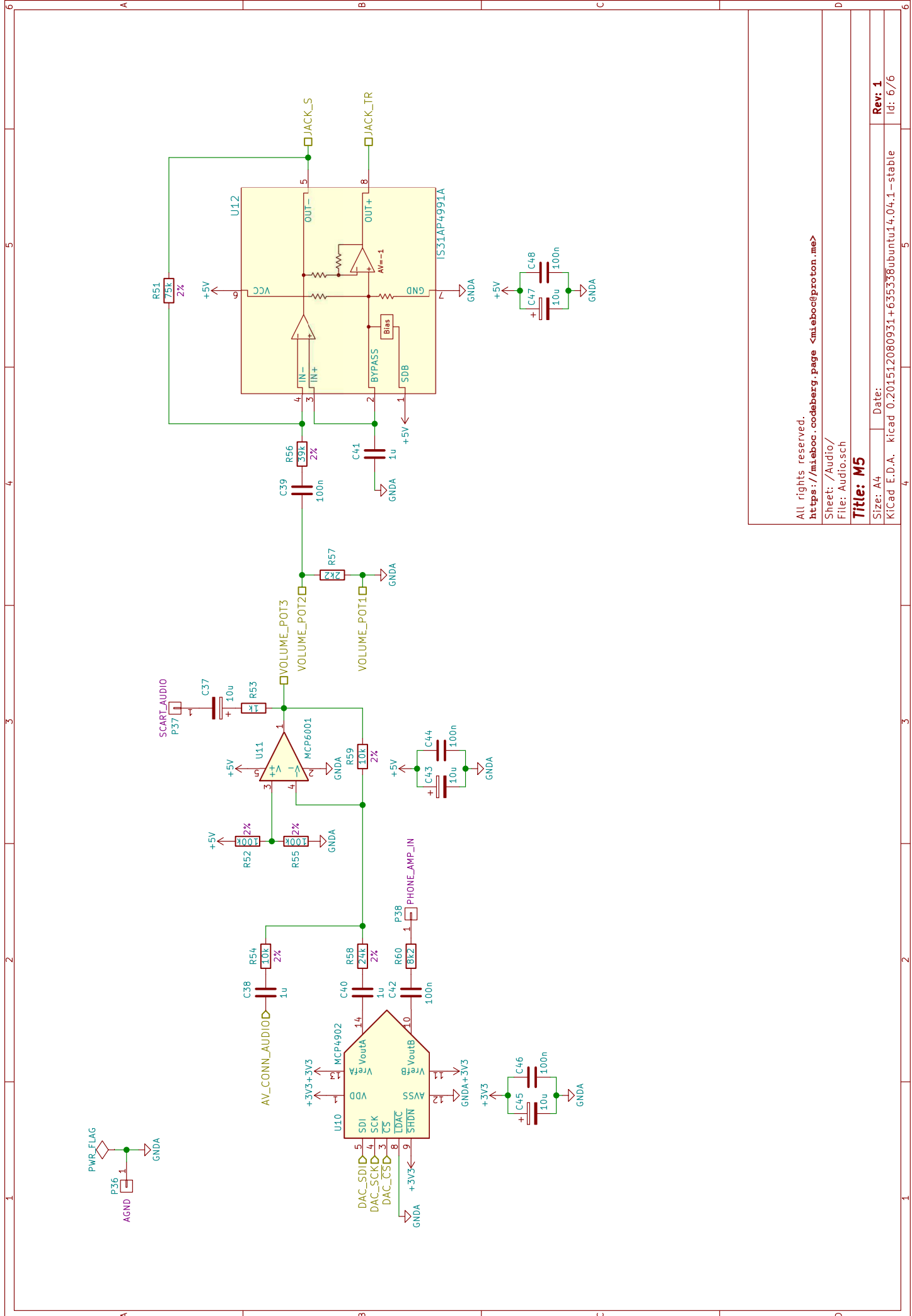
**PERIPHERALS\_CONNECTORS\_BOARD.**

1	CB_EXP +5V	2	CB_EXP +5V
2	CB_EXP +3V3	3	CB_EXP +3V3
3	CB_EXP_CS2	4	CB_EXP_CS3
4	CB_EXP_CS0	5	CB_EXP_CS1
5	CB_SPL_MISO	6	CB_SPL_MISO
6	CB_EXP_CS1	7	CB_SPL_MISO
7	CB_EXP_CS0	8	CB_SPL_MISO
8	CB_SPL_MISO	9	CB_SPL_SCLK
9	CB_EXP_CS1	10	CB_EXP_RESET
10	CB_EXP_CS0	11	CB_EXP_RESET
11	CB_SPL_SCLK	12	CB_EXP_GND
12	CB_EXP_RESET	13	CB_EXP_GND
13	CB_EXP_GND	14	CB_EXP_GND
14	CB_EXP_GND	15	CB_EXP_GND
15	CB_EXP_GND	16	CB_EXP_GND
16	CB_EXP_GND	17	CB_EXP_GND
17	CB_EXP_GND	18	CB_EXP_GND
18	CB_EXP_GND	19	CB_EXP_GND
19	CB_EXP_GND	20	CB_EXP_GND
20	CB_EXP_GND	21	CB_EXP_GND
21	CB_EXP_GND	22	CB_EXP_GND
22	CB_EXP_GND	23	CB_EXP_GND
23	CB_EXP_GND	24	CB_EXP_GND
24	CB_EXP_GND	25	CB_EXP_GND
25	CB_EXP_GND	26	CB_EXP_GND
26	CB_EXP_GND	27	CB_EXP_GND
27	CB_EXP_GND	28	CB_EXP_GND
28	CB_EXP_GND	29	CB_EXP_GND
29	CB_EXP_GND	30	CB_EXP_GND
30	CB_EXP_GND	31	CB_EXP_GND
31	CB_EXP_GND	32	CB_EXP_GND
32	CB_EXP_GND	33	CB_EXP_GND
33	CB_EXP_GND	34	CB_EXP_GND

1	CB_EXP +5V	2	CB_EXP +5V
2	CB_EXP +3V3	3	CB_EXP +3V3
3	CB_EXP_CS2	4	CB_EXP_CS3
4	CB_EXP_CS0	5	CB_EXP_CS1
5	CB_SPL_MISO	6	CB_SPL_MISO
6	CB_EXP_CS1	7	CB_SPL_MISO
7	CB_EXP_CS0	8	CB_SPL_MISO
8	CB_SPL_MISO	9	CB_SPL_SCLK
9	CB_EXP_CS1	10	CB_EXP_RESET
10	CB_EXP_CS0	11	CB_EXP_RESET
11	CB_SPL_SCLK	12	CB_EXP_GND
12	CB_EXP_RESET	13	CB_EXP_GND
13	CB_EXP_GND	14	CB_EXP_GND
14	CB_EXP_GND	15	CB_EXP_GND
15	CB_EXP_GND	16	CB_EXP_GND
16	CB_EXP_GND	17	CB_EXP_GND
17	CB_EXP_GND	18	CB_EXP_GND
18	CB_EXP_GND	19	CB_EXP_GND
19	CB_EXP_GND	20	CB_EXP_GND
20	CB_EXP_GND	21	CB_EXP_GND
21	CB_EXP_GND	22	CB_EXP_GND
22	CB_EXP_GND	23	CB_EXP_GND
23	CB_EXP_GND	24	CB_EXP_GND
24	CB_EXP_GND	25	CB_EXP_GND
25	CB_EXP_GND	26	CB_EXP_GND
26	CB_EXP_GND	27	CB_EXP_GND
27	CB_EXP_GND	28	CB_EXP_GND
28	CB_EXP_GND	29	CB_EXP_GND
29	CB_EXP_GND	30	CB_EXP_GND
30	CB_EXP_GND	31	CB_EXP_GND
31	CB_EXP_GND	32	CB_EXP_GND
32	CB_EXP_GND	33	CB_EXP_GND
33	CB_EXP_GND	34	CB_EXP_GND

1	CB_EXP +5V	2	CB_EXP +5V
2	CB_SPL_MISO	3	CB_SPL_MISO
3	CB_SPL_SCLK	4	CB_SPL_SCLK
4	CB_EXP_CS2	5	CB_EXP_CS3
5	CB_EXP_CS0	6	CB_EXP_CS1
6	CB_SPL_MISO	7	CB_SPL_MISO
7	CB_EXP_CS1	8	CB_EXP_RESET
8	CB_EXP_CS0	9	CB_EXP_RESET
9	CB_SPL_SCLK	10	CB_EXP_GND
10	CB_EXP_RESET	11	CB_EXP_GND
11	CB_EXP_GND	12	CB_EXP_GND
12	CB_EXP_GND	13	CB_EXP_GND
13	CB_EXP_GND	14	CB_EXP_GND
14	CB_EXP_GND	15	CB_EXP_GND
15	CB_EXP_GND	16	CB_EXP_GND
16	CB_EXP_GND	17	CB_EXP_GND
17	CB_EXP_GND	18	CB_EXP_GND
18	CB_EXP_GND	19	CB_EXP_GND
19	CB_EXP_GND	20	CB_EXP_GND
20	CB_EXP_GND	21	CB_EXP_GND
21	CB_EXP_GND	22	CB_EXP_GND
22	CB_EXP_GND	23	CB_EXP_GND
23	CB_EXP_GND	24	CB_EXP_GND
24	CB_EXP_GND	25	CB_EXP_GND
25	CB_EXP_GND	26	CB_EXP_GND
26	CB_EXP_GND	27	CB_EXP_GND
27	CB_EXP_GND	28	CB_EXP_GND
28	CB_EXP_GND	29	CB_EXP_GND
29	CB_EXP_GND	30	CB_EXP_GND
30	CB_EXP_GND	31	CB_EXP_GND
31	CB_EXP_GND	32	CB_EXP_GND
32	CB_EXP_GND	33	CB_EXP_GND
33	CB_EXP_GND	34	CB_EXP_GND





All rights reserved.  
<https://mieboc.codeberg.page> <mieboc@proton.me>

Sheet: /Audio/  
 File: Audio.sch

**Title: M5**

Size: A4 Date:

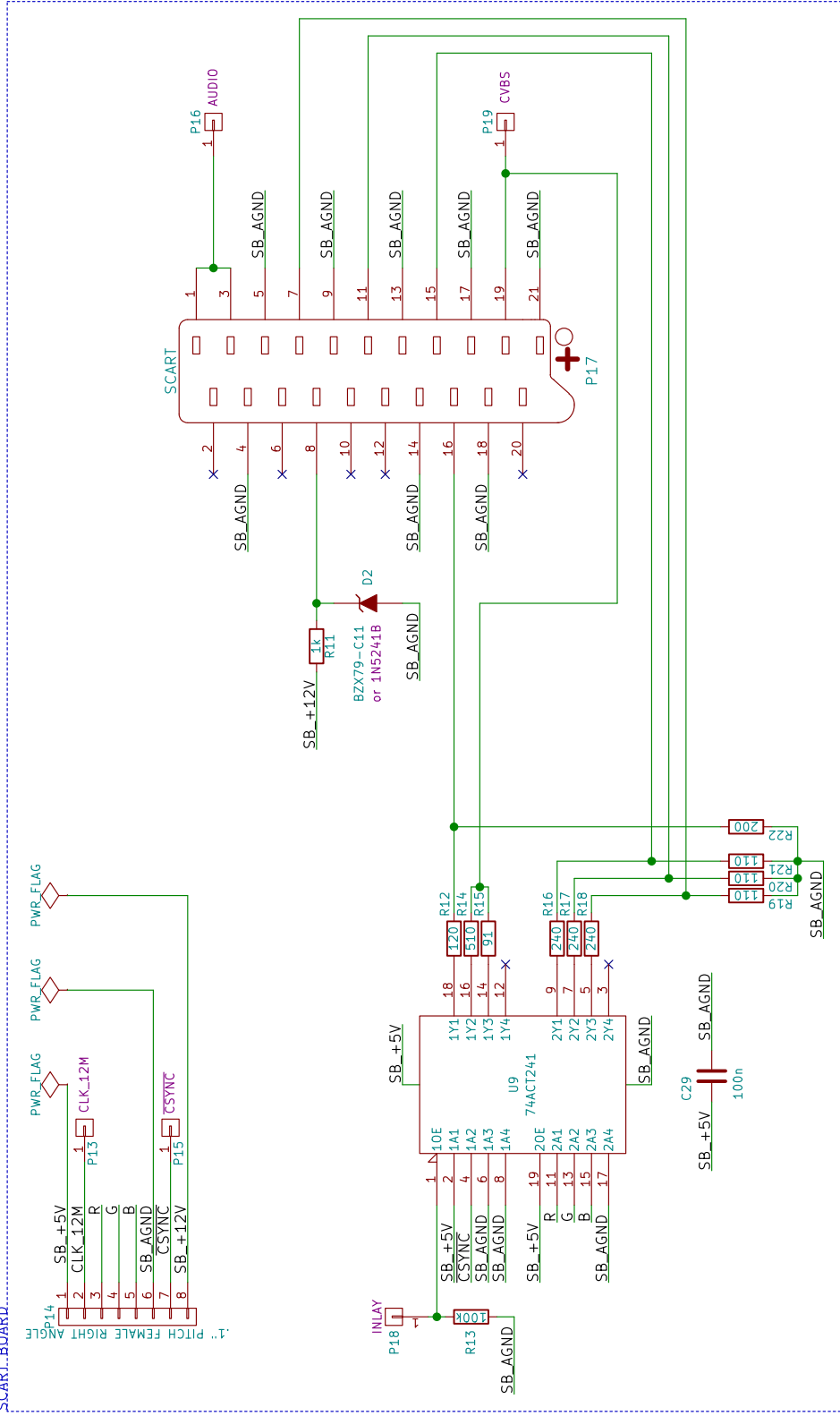
KiCad E.D.A. kicad 0.201512080931+635338ubuntu14.04.1 - stable  
 Id: 6/6

**Rev: 1**

Id: 6/6



SCART\_BOARD



All rights reserved.  
<https://mieboc.codeberg.page> <mieboc@proton.me>

Sheet: /SCART/  
 File: SCART.sch

Title: M5

Size: A4 Date:

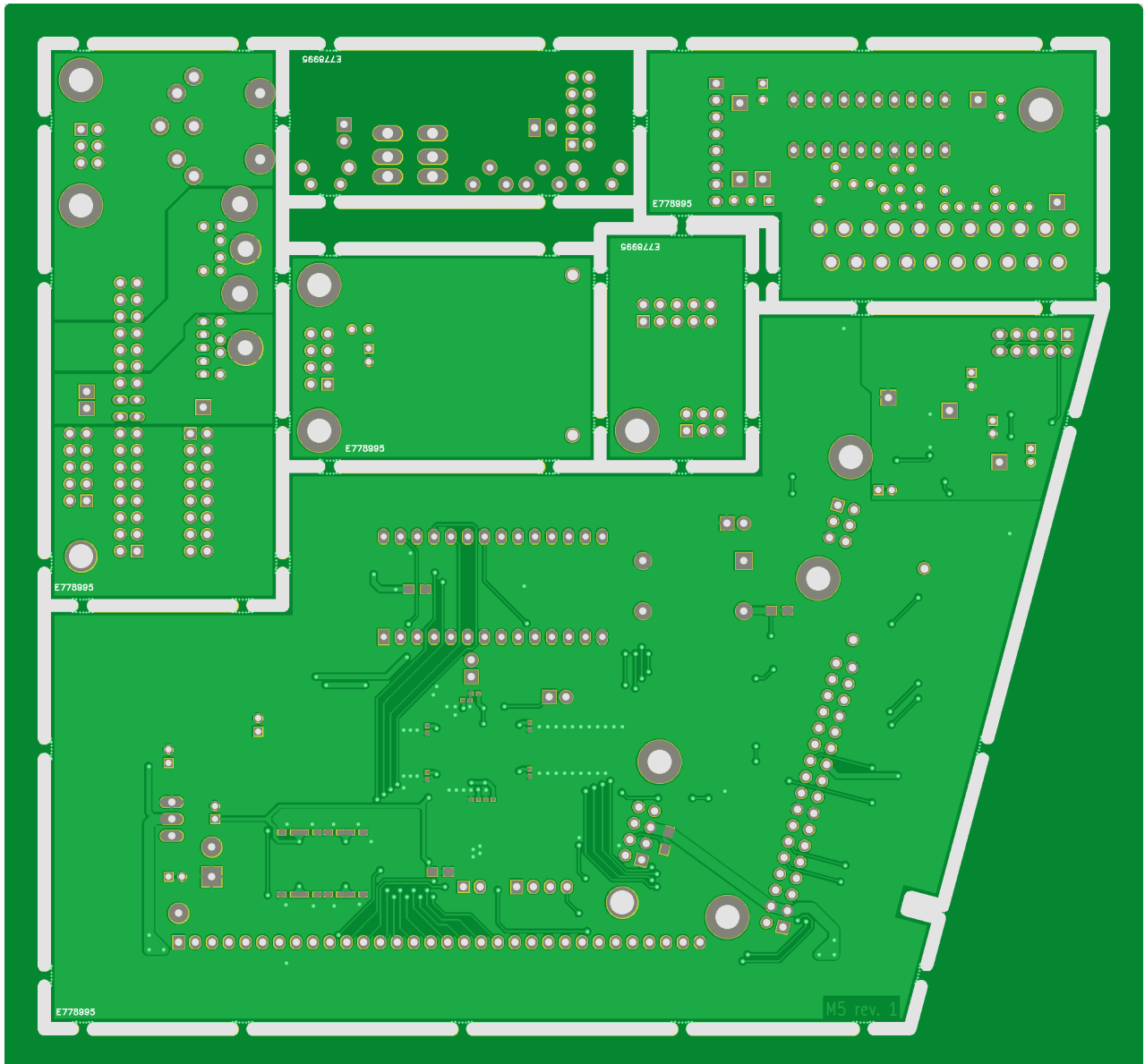
KiCad E.D.A. kicad 0.201512080931+635338ubuntu14.04.1-stable

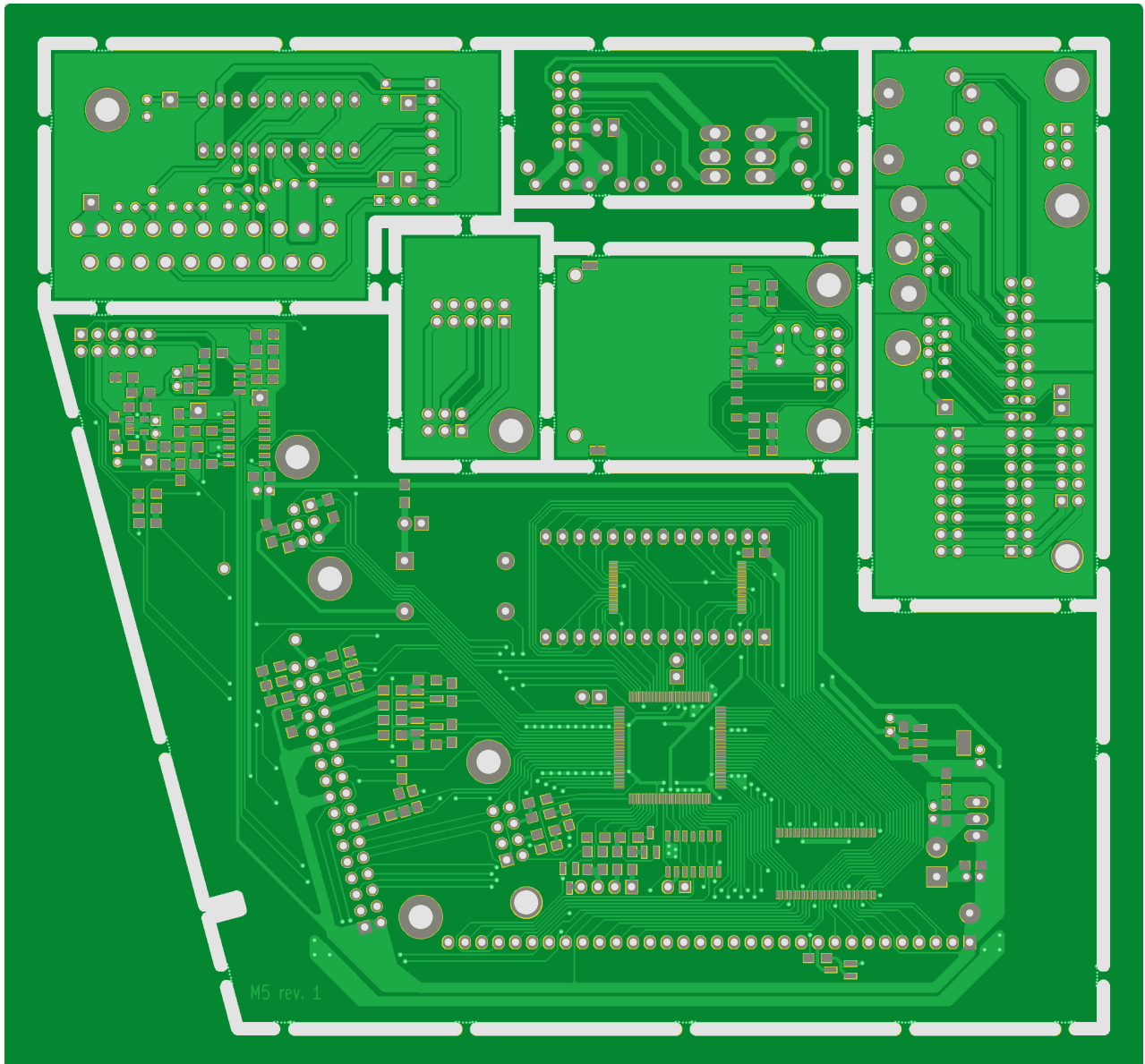
Rev: 1

Id: 4/6

## Annexe D

# Aperçu fourni par le fabricant des circuits imprimés





## Annexe E

# Exemple de fichier SVF programmant l'EEPROM

Ce qui suit est un exemple de fichier SVF tel que généré par le programme que nous avons écrit. Il permet de programmer *in situ* l'EEPROM de la carte M5 *via* la liaison JTAG entre le CPLD et le PC, puis de lire le contenu de l'EEPROM afin de vérifier que la programmation a été effectuée sans erreur. Le fichier ci-dessous programme quatre octets de l'EEPROM, à savoir les couples (*adresse, donnée*) suivants : (0000, a5), (0001, 33), (0a12, 0f), (fff, ac).

Chaque instruction de la forme SDR 240 TDI (...); force les broches du CPLD à un état donné par le contenu des parenthèses.

Les instructions RUNTEST 11E-4 SEC; ordonnent l'attente du délai spécifié avant de passer à l'instruction suivante. Elles sont nécessaires car l'EEPROM a besoin de 10 ms au maximum pour terminer une opération d'écriture. Le délai spécifié devrait être 11E-3 SEC, mais pour une raison que nous ignorons, en pratique, le délai effectivement attendu par l'ordinateur est environ 10 fois plus grand que celui spécifié. Nous ignorons si cela est dû au logiciel qui interprète le fichier SVF, au câble JTAG utilisé, ou à une autre cause encore.

Les commentaires sont signalés par un point d'exclamation.

```
! M5 REV 1 EEPROM INDIRECT PROGRAMMING SVF FILE
! This file was automatically generated by the eeprom_svfggen_byte_mode program,
  version 0.1.

TRST ABSENT;
ENDDR IDLE;
ENDIR IDLE;
STATE IDLE;

! Check device ID

SIR 10 TDI (006);
SDR 32 TDI (FFFFFFFF) TDO (020A50DD) MASK (FFFFFFFF);

! Check usercode
```

```

SIR 10 TDI (007);
SDR 32 TD0 (4D353130) MASK (FFFFFF00);

! Force the CPLD pins as required and put MCU in reset state

SIR 10 TDI (005);
SDR 240 TDI (F7B6C926DE7DF6DB7FA4DBDFFDFFFFFFD1FFFFFFFFFFFFE01B69B6FF6DB6F);
SIR 10 TDI (00F);

! Wait in case a write operation in the EEPROM is already in progress

RUNTEST 11E-4 SEC;

! Have the CPLD take over the bus and assert the EEPROM chip enable line
SDR 240 TDI (F7B6C926DE7DF6DB7FA4DBDFFDFFFFFFDB6DB6DB6DB6DA29B69A2FF6DB6F);

! Program

! 0000 a5
SDR 240 TDI (F7B6C926DE7DF6DB7FA4DBDFFDFFFFFFDA00000145028A29A01007E00A2F);
SDR 240 TDI (F7B6C926DE7DF6DB7FA4DBDFFDFFFFFFDA00000145028A29001007E00A2F);
SDR 240 TDI (F7B6C926DE7DF6DB7FA4DBDFFDFFFFFFDA00000145028A29A01007E00A2F);
RUNTEST 11E-4 SEC;
! 0001 33
SDR 240 TDI (F7B6C926DE7DF6DB7FA4DBDFFDFFFFFFDA00000005A05A29A01007E00B6F);
SDR 240 TDI (F7B6C926DE7DF6DB7FA4DBDFFDFFFFFFDA00000005A05A29001007E00B6F);
SDR 240 TDI (F7B6C926DE7DF6DB7FA4DBDFFDFFFFFFDA00000005A05A29A01007E00B6F);
RUNTEST 11E-4 SEC;
! 0a12 0f
SDR 240 TDI (F7B6C926DE7DF6DB7FA4DBDFFDFFFFFFDA28A0000016DA29A0102FE05A2F);
SDR 240 TDI (F7B6C926DE7DF6DB7FA4DBDFFDFFFFFFDA28A0000016DA2900102FE05A2F);
SDR 240 TDI (F7B6C926DE7DF6DB7FA4DBDFFDFFFFFFDA28A0000016DA29A0102FE05A2F);
RUNTEST 11E-4 SEC;
! ffff ac
SDR 240 TDI (F7B6C926DE7DF6DB7FA4DBDFFDFFFFFFDB6DB6DB45168029B69A2FF6DB6F);
SDR 240 TDI (F7B6C926DE7DF6DB7FA4DBDFFDFFFFFFDB6DB6DB45168029169A2FF6DB6F);
SDR 240 TDI (F7B6C926DE7DF6DB7FA4DBDFFDFFFFFFDB6DB6DB45168029B69A2FF6DB6F);
RUNTEST 11E-4 SEC;
RUNTEST 11E-4 SEC;

! Verify

SDR 240 TDI (F7B6C926DE7DF6DB7FA4DBDFFDFFFFFFDA000001D70BAE01A01007E0002F); ! 0000
a5
SDR 240 TDI (F7B6C926DE7DF6DB7FA4DBDFFDFFFFFFDA00000097A97E01A01007E0016F) TD0 (
F7B6C926DE7DF6DB7FA4DBDFFDFFFFFFDA000001D70BAE01A01007E0016F) MASK
(49249200000000000000); ! 0001 33
SDR 240 TDI (F7B6C926DE7DF6DB7FA4DBDFFDFFFFFFDA28A000921FFE01A0102FE0502F) TD0 (
F7B6C926DE7DF6DB7FA4DBDFFDFFFFFFDA28A00097A97E01A0102FE0502F); ! 0a12 0f
SDR 240 TDI (F7B6C926DE7DF6DB7FA4DBDFFDFFFFFFDB6DB6DBD71FA401B69A2FF6D16F) TD0 (
F7B6C926DE7DF6DB7FA4DBDFFDFFFFFFDB6DB6DA921FFE01B69A2FF6D16F); ! ffff ac

! Verify last byte and force pins back to idle state

SDR 240 TDI (F7B6C926DE7DF6DB7FA4DBDFFDFFFFFFD1FFFFFFFFFFFFE01B69B6FF6DB6F) TD0 (
F7B6C926DE7DF6DB7FA4DBDFFDFFFFFFDB6DB6DBD71FA401B69A2FF6D16F);

! End

```

## Annexe F

# Code source du module crayon optique

```
module light_pen (clk, vsync_n, hsync_n, sensor,
                 hold, line, column);

    parameter line_counter_size = 8;
    parameter column_counter_size = 8;

    input clk;
    input vsync_n;
    input hsync_n;
    input sensor;
    input hold;
    output reg [line_counter_size-1:0] line;
    output reg [column_counter_size-1:0] column;

    reg [line_counter_size-1:0] line_counter;
    reg [column_counter_size-1:0] column_counter;

    reg [line_counter_size-1:0] line_tmp;
    reg [column_counter_size-1:0] column_tmp;

    reg sensed;

    always @(negedge vsync_n, negedge hsync_n)
        if (~vsync_n) line_counter <= 0;
        else          line_counter <= line_counter + 1'b1;

    always @(negedge hsync_n, posedge clk)
        if (~hsync_n) column_counter <= 0;
        else          column_counter <= column_counter + 1'b1;

    always @(posedge sensor, negedge vsync_n)
```

```
    if (~vsync_n)
    begin
        sensed <= 1'b0;
        column_tmp <= 2'd2**column_counter_size - 1'b1;
    end
    else if (column_counter < column_tmp)
    begin
        sensed <= 1'b1;
        line_tmp <= line_counter;
        column_tmp <= column_counter;
    end

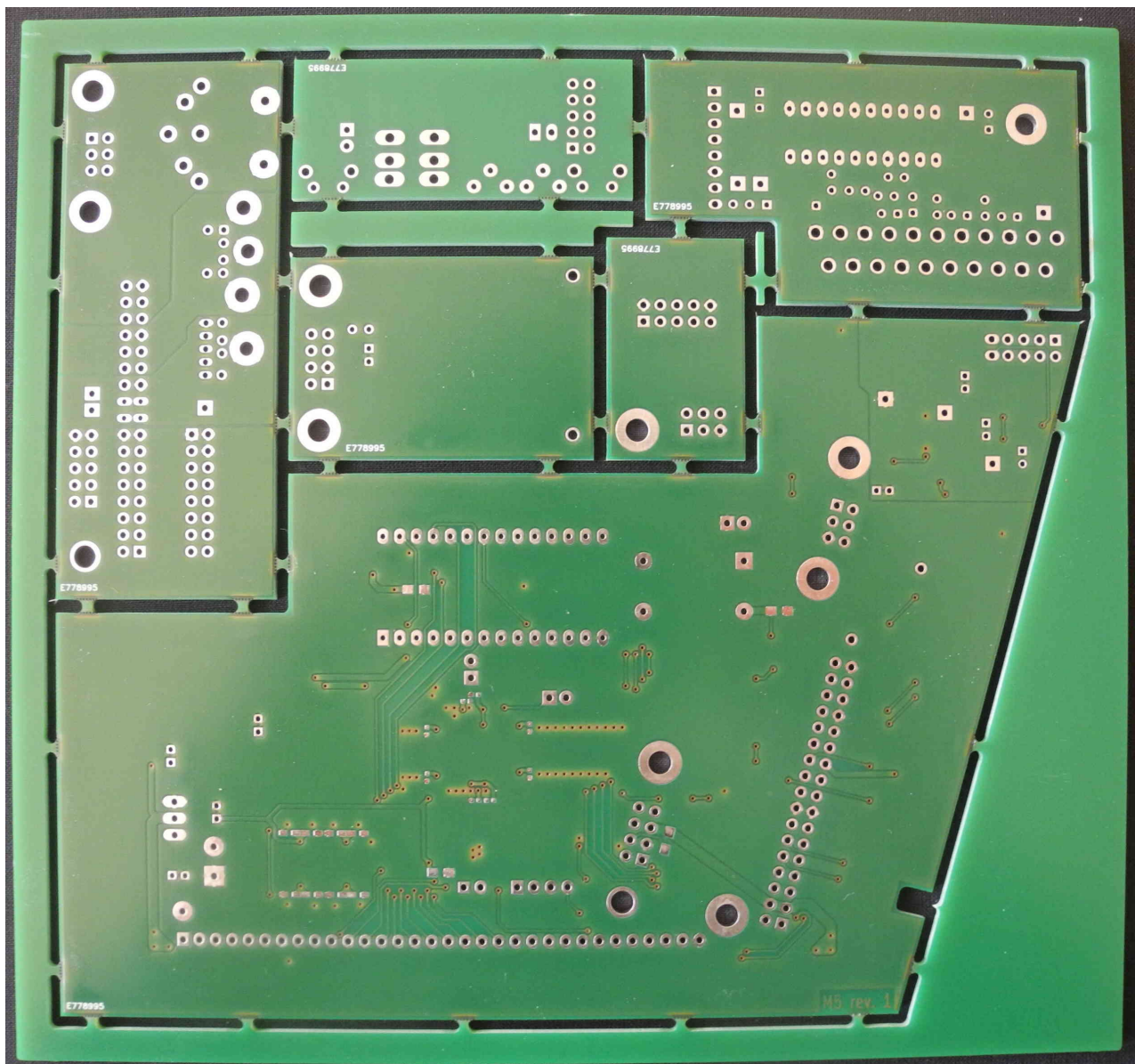
    always @(negedge vsync_n)
    if (~hold)
    begin
        line <= sensed ? line_tmp : 0;
        column <= sensed ? column_tmp : 0;
    end

endmodule
```

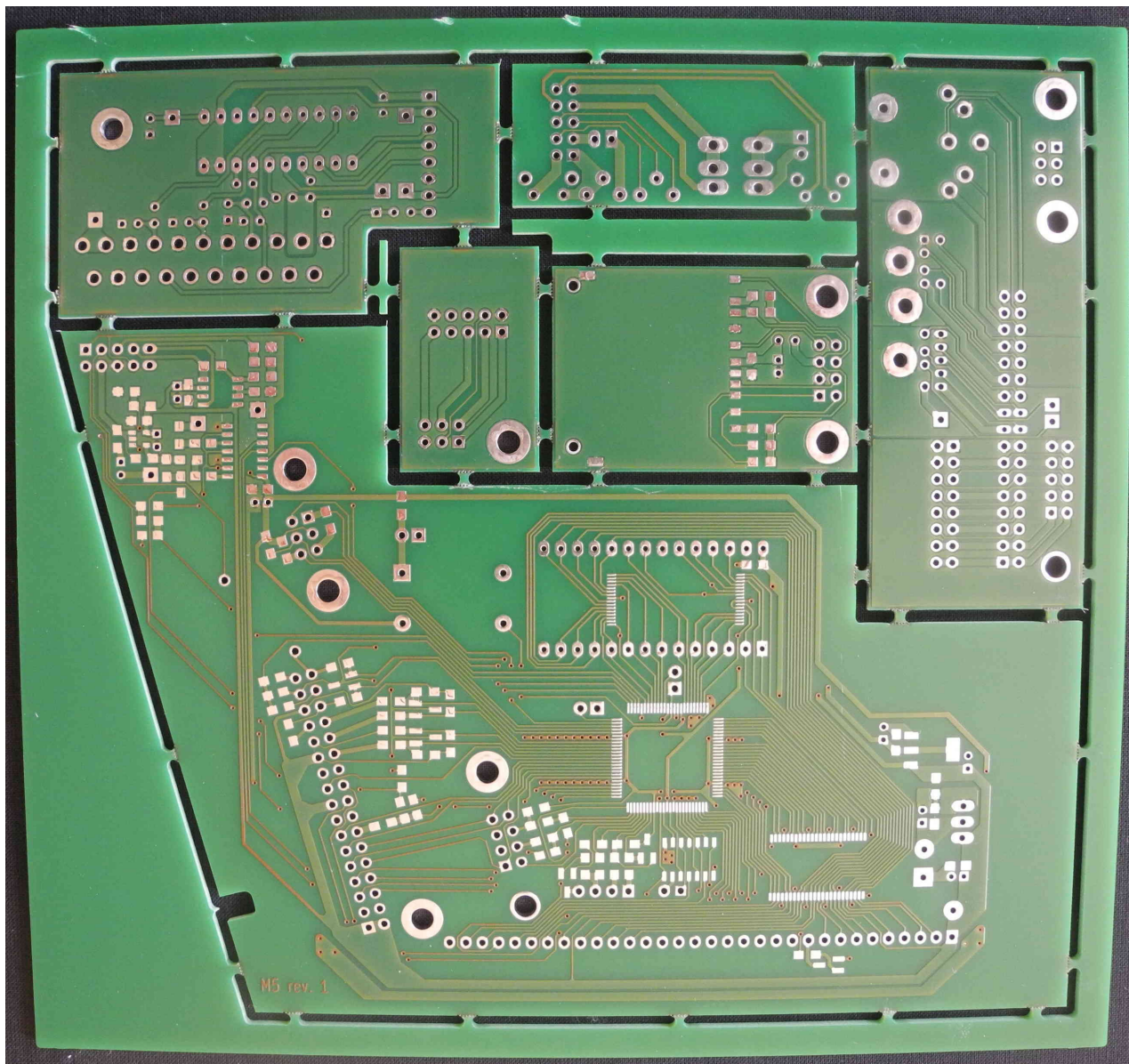


**Annexe G**

# **Photographies**

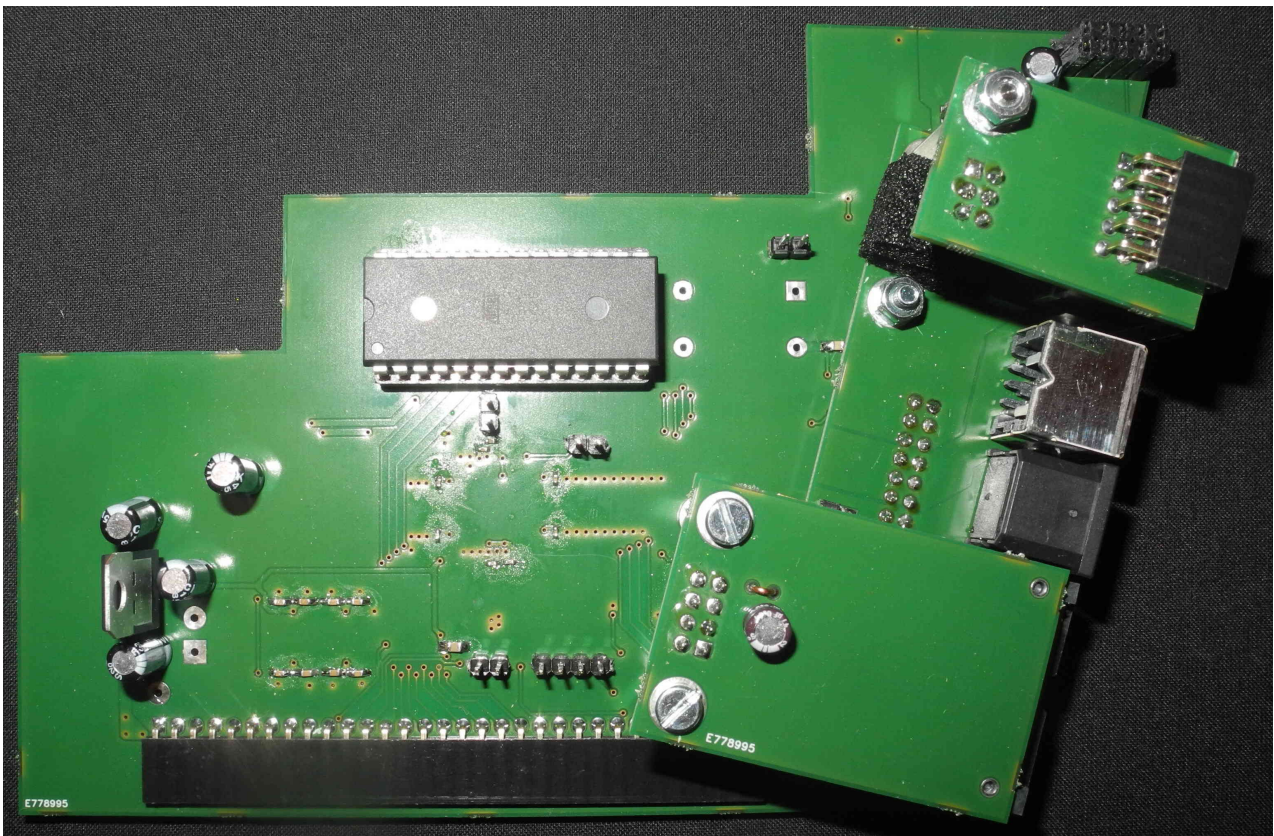


Photographie du panneau livré par l'usine, face de dessus

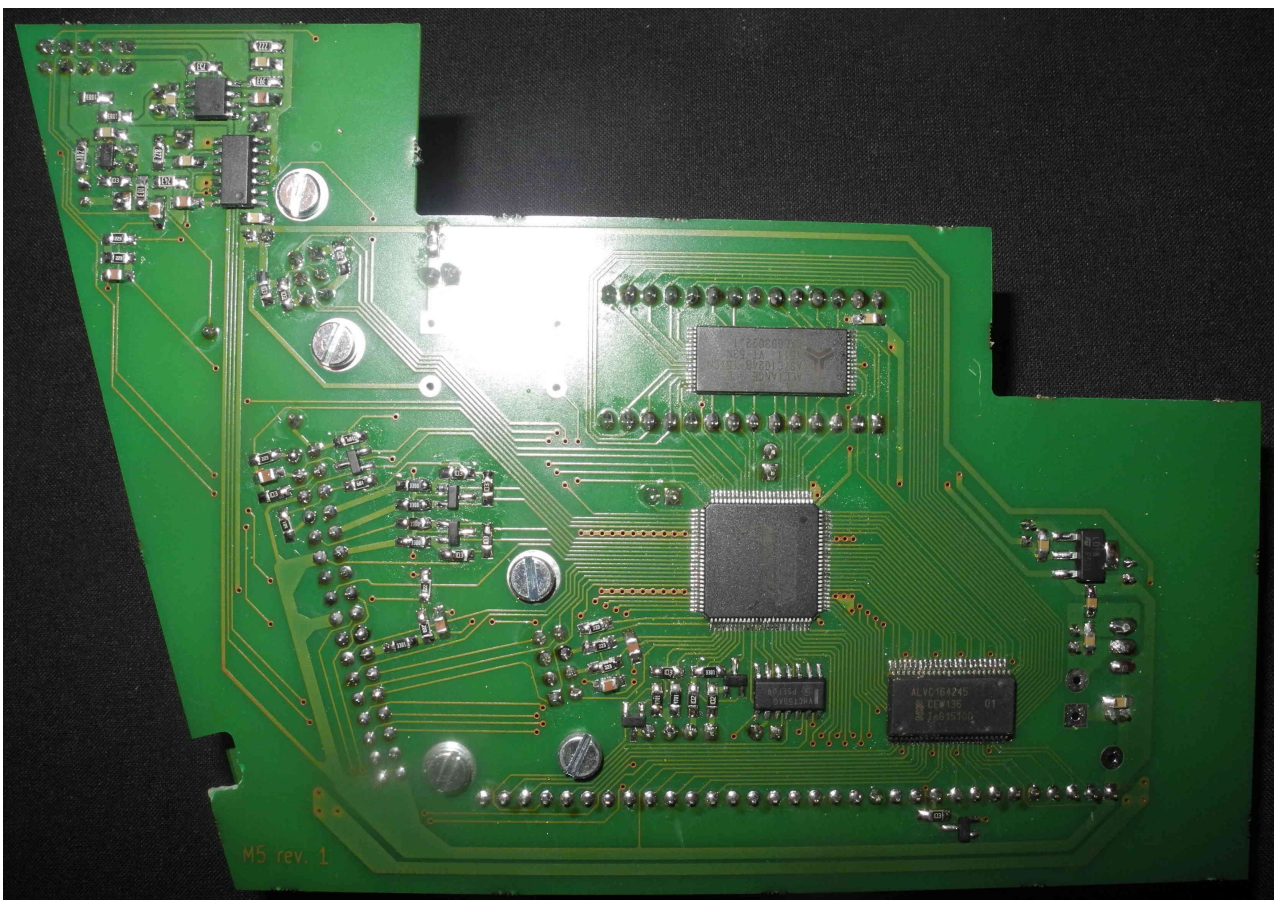


Photographie du panneau livré par l'usine, face de dessous



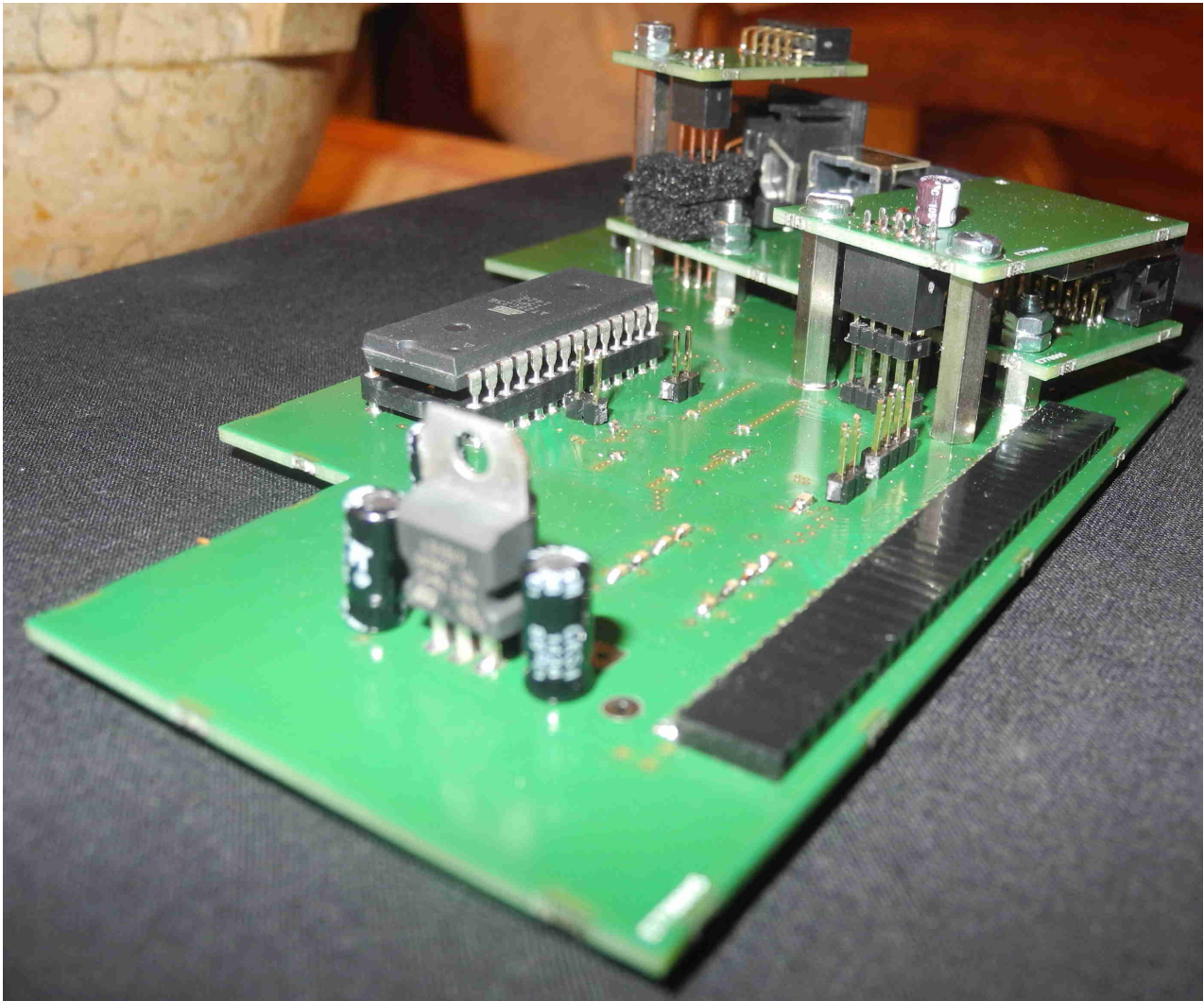


Photographie des circuits imprimés peuplés et assemblés, face de dessus

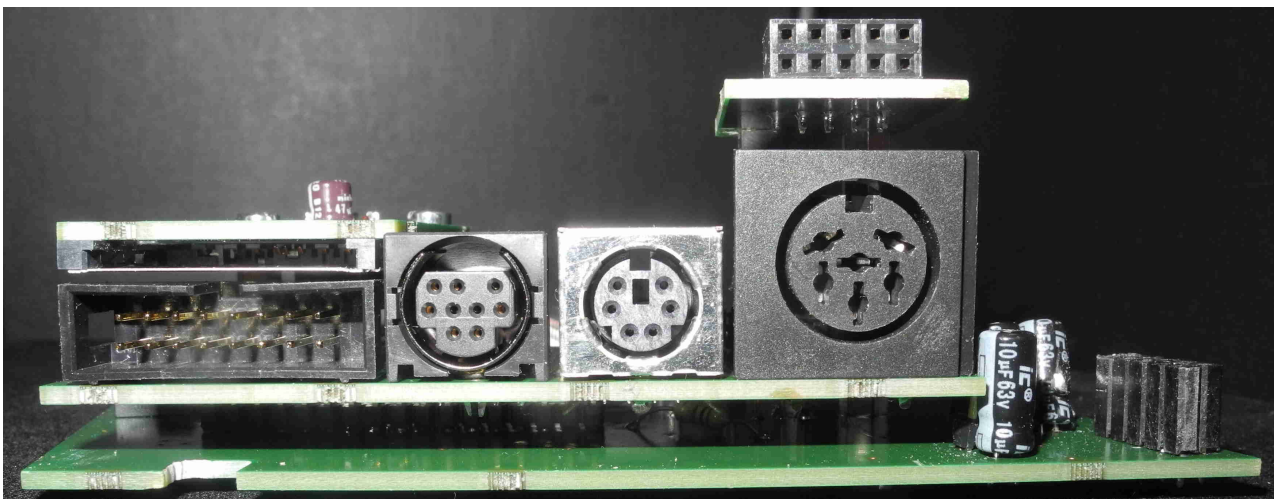


Photographie des circuits imprimés peuplés et assemblés, face de dessous





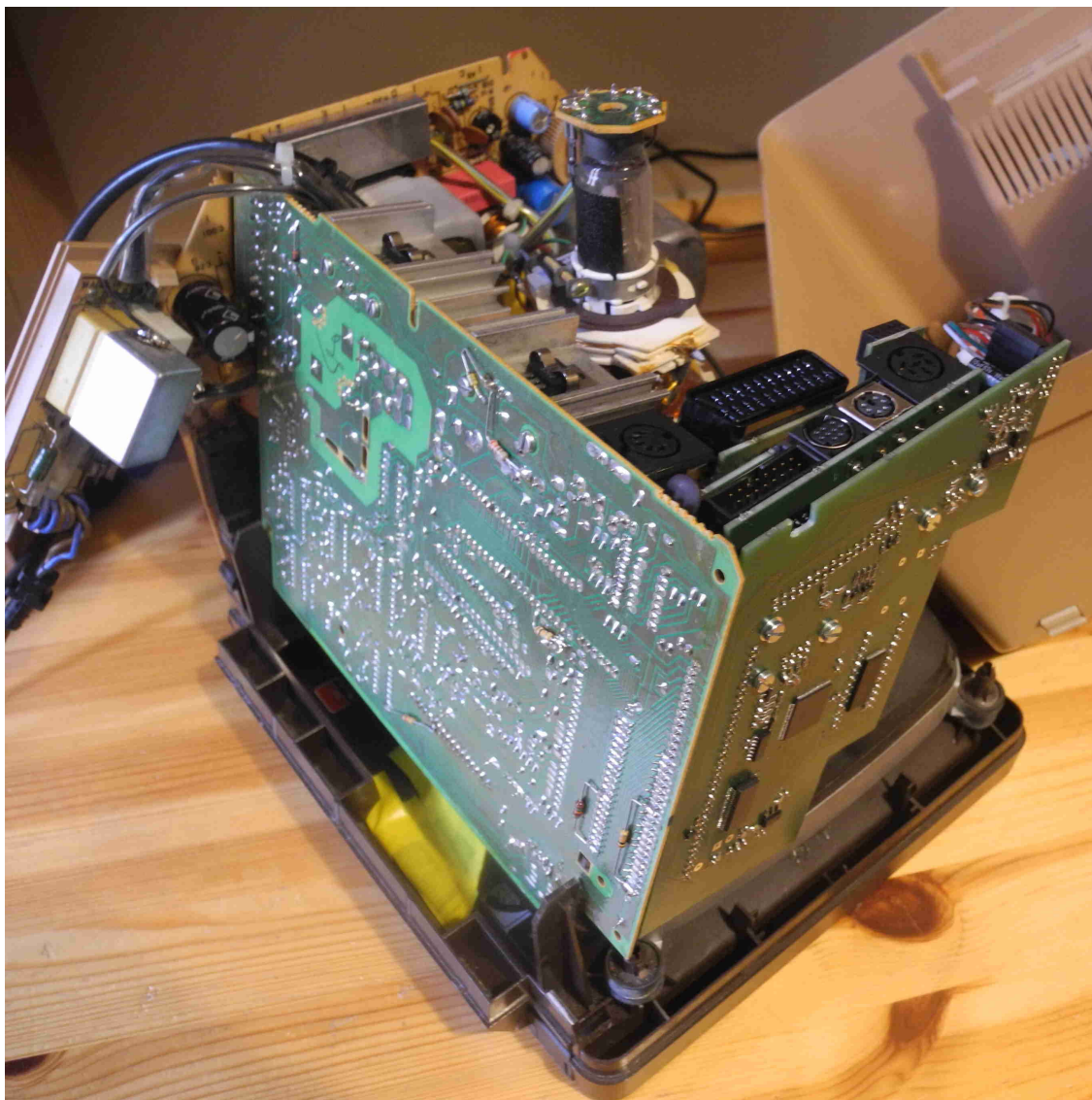
Photographie des circuits imprimés peuplés et assemblés, vue d'ensemble



Photographie des circuits imprimés peuplés et assemblés, vue des connecteurs



Photographie du haut-parleur fixé dans la coque du Minitel



Photographie de la carte M5 enfichée sur la carte UT



Photographie des connecteurs à l'arrière du Minitel modifié





Photographie d'un programme de test du crayon optique en cours d'exécution

Le programme permet à l'utilisateur de dessiner grossièrement à l'écran à l'aide du crayon optique. En bas à gauche de l'écran sont affichées les coordonnées du crayon.

La grossièreté du dessin n'est pas due à un manque de précision du crayon optique, mais au fait que, pour simplifier la programmation du logiciel, nous avons limité la granularité du dessin à la taille des cases de caractères de texte. Il est possible d'obtenir un dessin plus fin en utilisant les caractères semi-graphiques du VGP ou des caractères personnalisés.