

PROJET MINITEL

Afficher des écrans Minitel avec un mini-serveur sur ESP32

MATERIEL

Si vous n'avez pas de Minitel, il est facile et peu onéreux de s'en procurer un sur une brocante ou sur internet. Privilégier les Minitels 1B et 2 qui possèdent une prise DIN à 5 broches à l'arrière permettant la connexion de périphériques. Nous utiliserons cette prise pour communiquer avec un microcontrôleur afin de simuler un serveur Minitel.

Cette prise offre une entrée/sortie utilisant des tensions de 0 ou 5 V.

Les broches qui nous intéressent sont les suivantes :

1 : Entrée RX - Réception de données venant du périphérique (Arduino ou ESP32)

2 : Masse (GND)

3 : Sortie TX - Emission de données vers le périphérique (Arduino ou ESP32)

La broche 5 est une sortie alimentation que je n'ai pas utilisé pour mon projet.

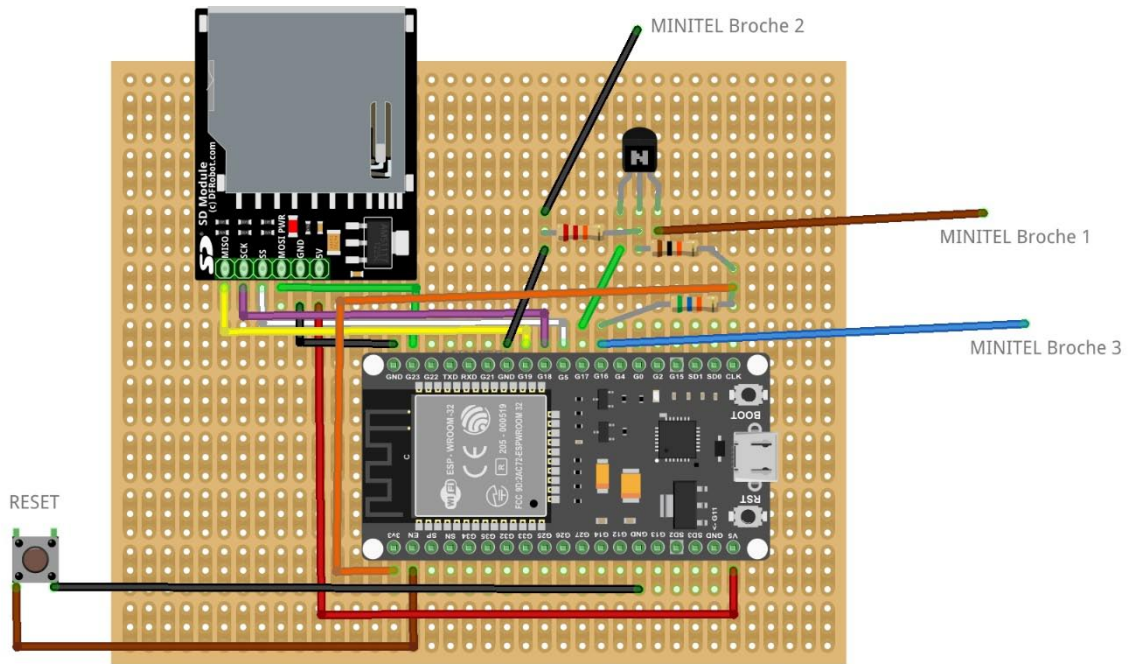


Si l'on utilise un Arduino, le montage est très simple car l'on peut connecter directement ces E/S aux E/S de l'Arduino.

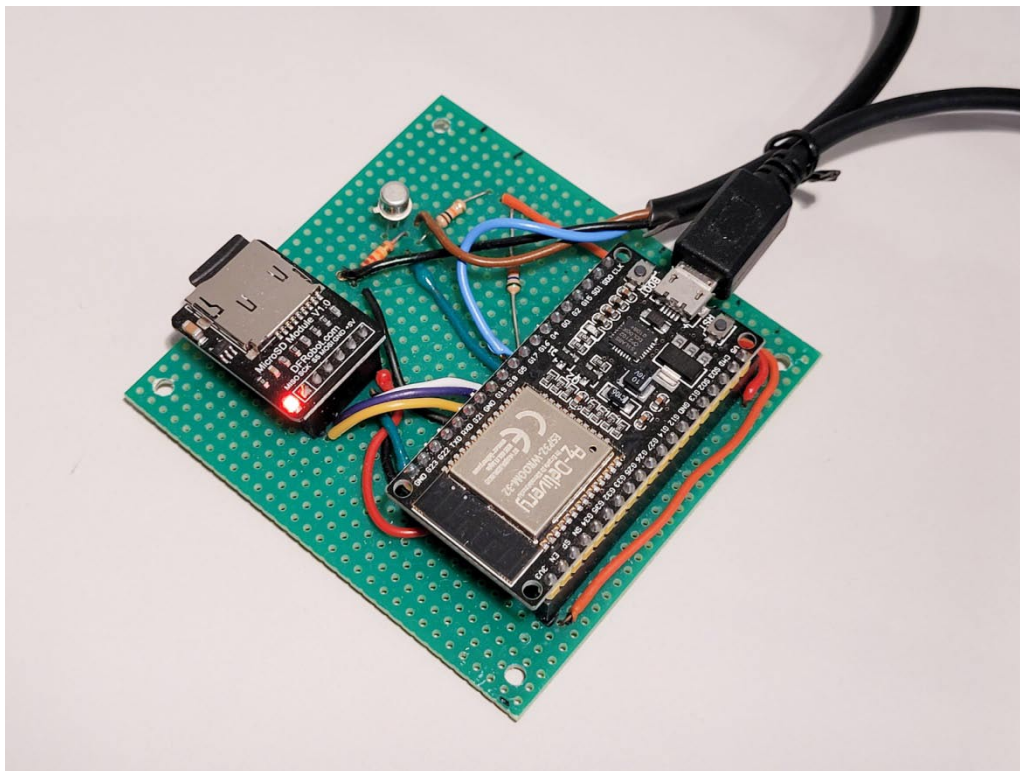
Avec un ESP32, il faut abaisser la tension à 3V. J'ai utilisé le montage proposé sur le site hackaday : <https://hackaday.io/project/180473-minitel-esp32> (qui vend un montage tout fait, mais communique également le schéma ainsi que de nombreux exemples bien utiles). J'ai ajouté un lecteur de carte SD pour stocker des fichiers textes (notamment pour l'annuaire téléphonique et le jeu du pendu).

Liste des composants :

- Transistor 2N2222
- Résistances 10 k Ω , 22 k Ω , 56 k Ω
- Module ESP32 (AZDelivery)
- Lecteur de carte SD (selon projet)



fritzing



LOGICIEL

Communications ESP32-Minitel

Bibliothèque

Pour communiquer avec le Minitel, j'ai utilisé l'excellente bibliothèque Minitel1B_Hard.h d'Eric Sérandour (<https://entropie.org/3615/index.php/software/>) disponible sur Github : https://github.com/eserandour/Minitel1B_Hard

```
#include <Minitel1B_Hard.h>
```

Sélection du port série

Le 2^{ème} port série de l'ESP32 est utilisé pour la communication avec le Minitel (UART2) :

```
RX=GPIO16
```

```
TX=GPIO17=
```

```
Minitel minitel(Serial2);
```

Vitesse de la liaison série

```
minitel.changeSpeed(minitel.searchSpeed());
```

Sans paramètre = vitesse maxi (9600 pour Minitel 2)

Suppression de l'écho à la saisie

```
minitel.echo(false);
```

Pour éviter l'affichage en double avec les instructions `print`

Affichage en mode texte

Exemple affichage du cout de la connexion :

```
minitel.newScreen();
```

```
minitel.textMode();
```

```
minitel.noCursor();
```

```
minitel.moveCursorXY(30, 1); //30ème colonne, 1ère ligne
```

```
minitel.print("COUT:");
```

```
minitel.print(COUT_TXT);
```

```
minitel.println("E");
```

Exemple d'affichage de texte en double hauteur :

```
minitel.attributs(DOUBLE_HAUTEUR); //DOUBLE_LARGEUR ou DOUBLE_GRANDEUR
```

```
minitel.println("SOMMAIRE MINITEL");
```

```
minitel.attributs(GRANDEUR_NORMALE);
```


3615 RETROTECH

COUT: 0.37E

TARIFICATION

Exemple d'affichage de caractères spéciaux et d'effet vidéo :

```
minitel.moveCursorXY(14, 24);
minitel.print("quitter le quizz ");
minitel.printSpecialChar(FLECHE_DROITE);
minitel.print(" ");
minitel.attributs(INVERSION_FOND);
minitel.print("SOMMAIRE");
minitel.attributs(FOND_NORMAL);
```



quitter le quizz → SOMMAIRE

Affichage en mode graphique

Exemple d'affichage d'une ligne horizontale :

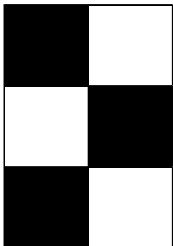
```
minitel.graphicMode();
minitel.hLine(1, 21, 40, CENTER); // TOP, CENTER ou BOTTOM
```

Exemple d'affichage d'un caractère semi-graphique :

En mode graphique, chaque caractère est constitué de 6 gros pixels, codé par un nombre écrit en binaire, 0 pour la couleur d'arrière-plan, 1 pour la couleur du premier plan.

```
minitel.graphicMode();
minitel.graphic(B100110);
```

Affiche le caractère suivant :



Affichage de fichier Videotex

Les fichiers Videotex VDT sont codés en décimal. Pour les afficher avec l'ESP32, il faut les convertir en hexadécimal. Ceci peut être réalisé avec le convertisseur d'Eric Sérandour (https://github.com/eserandour/Conversion_Videotex_Hex) basé sur Processing.

Exemple affichage écran 1 du quizz :

```
const int LONGUEUR_TRAME_Q1 = 202;

const prog_uchar IMAGE_Q1[] PROGMEM =
{0x0C,0x1B,0x47,0x14,0x1E,0x0F,0x33,0x36,0x31,0x35,0x20,0x52,0x45,0x54,0x52
,0x4F,0x54,0x45,0x43,0x48,0x1F,0x42,0x41,0x60,0x12,0x67,0x1F,0x44,0x41,0x1B
,0x4F,0x51,0x55,0x49,0x5A,0x5A,0x20,0x4D,0x49,0x4E,0x49,0x54,0x45,0x4C,0x1F
,0x46,0x41,0x1B,0x5D,0x51,0x55,0x45,0x53,0x54,0x49,0x4F,0x4E,0x20,0x31,0x2F
,0x31,0x30,0x1B,0x5C,0x1F,0x48,0x41,0x4C,0x65,0x20,0x4D,0x69,0x6E,0x69,0x74
,0x65,0x6C,0x20,0x19,0x42,0x65,0x74,0x61,0x69,0x74,0x20,0x69,0x6E,0x73,0x70
,0x69,0x72,0x19,0x42,0x65,0x20,0x64,0x27,0x75,0x6E,0x65,0x0D,0x0A,0x65,0x78
,0x70,0x19,0x42,0x65,0x72,0x69,0x65,0x6E,0x63,0x65,0x0D,0x0A,0x1F,0x4D,0x43
,0x1B,0x4E,0x1B,0x5D,0x31,0x1B,0x5C,0x1B,0x4C,0x20,0x41,0x6C,0x6C,0x65,0x6D
,0x61,0x6E,0x64,0x65,0x1F,0x4F,0x43,0x1B,0x4E,0x1B,0x5D,0x32,0x1B,0x5C,0x1B
,0x4C,0x20,0x41,0x6D,0x19,0x42,0x65,0x72,0x69,0x63,0x61,0x69,0x6E,0x65,0x1F
,0x51,0x43,0x1B,0x4E,0x1B,0x5D,0x33,0x1B,0x5C,0x1B,0x4C,0x20,0x4A,0x61,0x70
,0x6F,0x6E,0x61,0x69,0x73,0x65,0x1F,0x54,0x43,0x56,0x4F,0x54,0x52,0x45,0x20
,0x43,0x48,0x4F,0x49,0x58,0x20,0x3A
};

for (int i = 0; i < LONGUEUR_TRAME_Q1; i++) {
minitel.writeByte(pgm_read_byte_near(IMAGE_Q1 + i));
}
}
```



Récupération d'informations sur le Minitel

```
minitel.identifyDevice()
```

Renvoie 3 octets XXYYZZ (exemple 0x427639)

XX = constructeur

- Matra : 0x41
- Philips : 0x42

- Telic-Alcatel : 0x43

YY = type

- Minitel 1 : 0x62, 0x63 ou 0x72 suivant les modèles
- Minitel 1 Couleur : 0x73
- Minitel 1 Dialogue : 0x72
- Minitel 10 : 0x64 ou 0x66 suivant les modèles
- Minitel 1 Bistandard : 0x75
- Minitel 10 Bistandard : 0x77
- Minitel 2 : 0x76
- Minitel 12 : 0x7A
- Minitel 5 : 0x79

ZZ = version du logiciel

Saisies clavier

```
unsigned long TOUCHE = 0;      // Code décimal de la touche pressée
```

```
TOUCHE = minitel.getKeyCode();
```

Les touches numériques et alphanumériques suivent le codage ascii habituel. Les touches spécifiques du Minitel sont définies dans la bibliothèque Minitel1B_Hard et peuvent donc être testées facilement :

- ENVOI
- RETOUR
- REPETITION
- GUIDE
- ANNULATION
- SOMMAIRE
- CORRECTION
- SUITE
- CONNEXION_FIN

Exemple :

```
if (TOUCHE == SOMMAIRE) {  
  FIN = true;  
}
```