

Sommaire

## But du projet

Ce premier mini-projet avait pour objectif de concevoir et tester un panneau led, nous avons donc à disposition le matériel physique permettant d'assembler le panneau led et le matériel logiciel (CodeBlocks, SmartTTY, console Raspberry) pour faire en sorte que le panneau led puisse fonctionner correctement.

## Liste du matériel utilisé

1 panneau  
1 bornier pour le shield  
2 connecteurs HE10  
2 mètres de nappe  
Toolchain disponible à partir d'une clé USB  
1 clé USB wifi  
1 fer à souder  
1 oscilloscope  
1 voltmètre

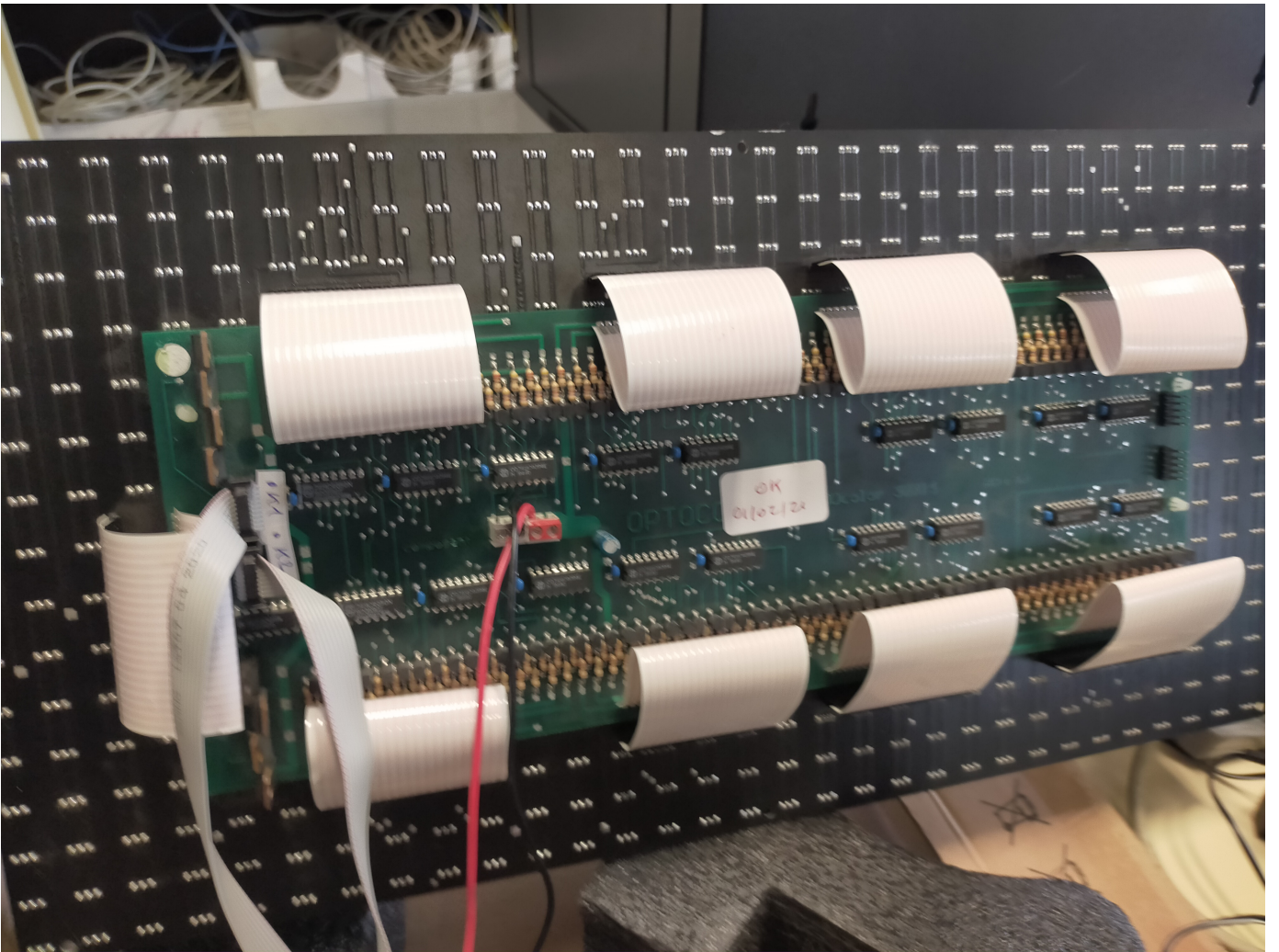
## Tâches réalisées et répartition

Theo Coignet  
Installer/configurer Raspbian Jessie  
Installer la chaîne de compilation  
Tester la chaîne de compilation  
Comprendre la commande d'une LED, d'une ligne de LED, et d'une rangée  
Réaliser des algorithmes, puis coder, tester et déboguer

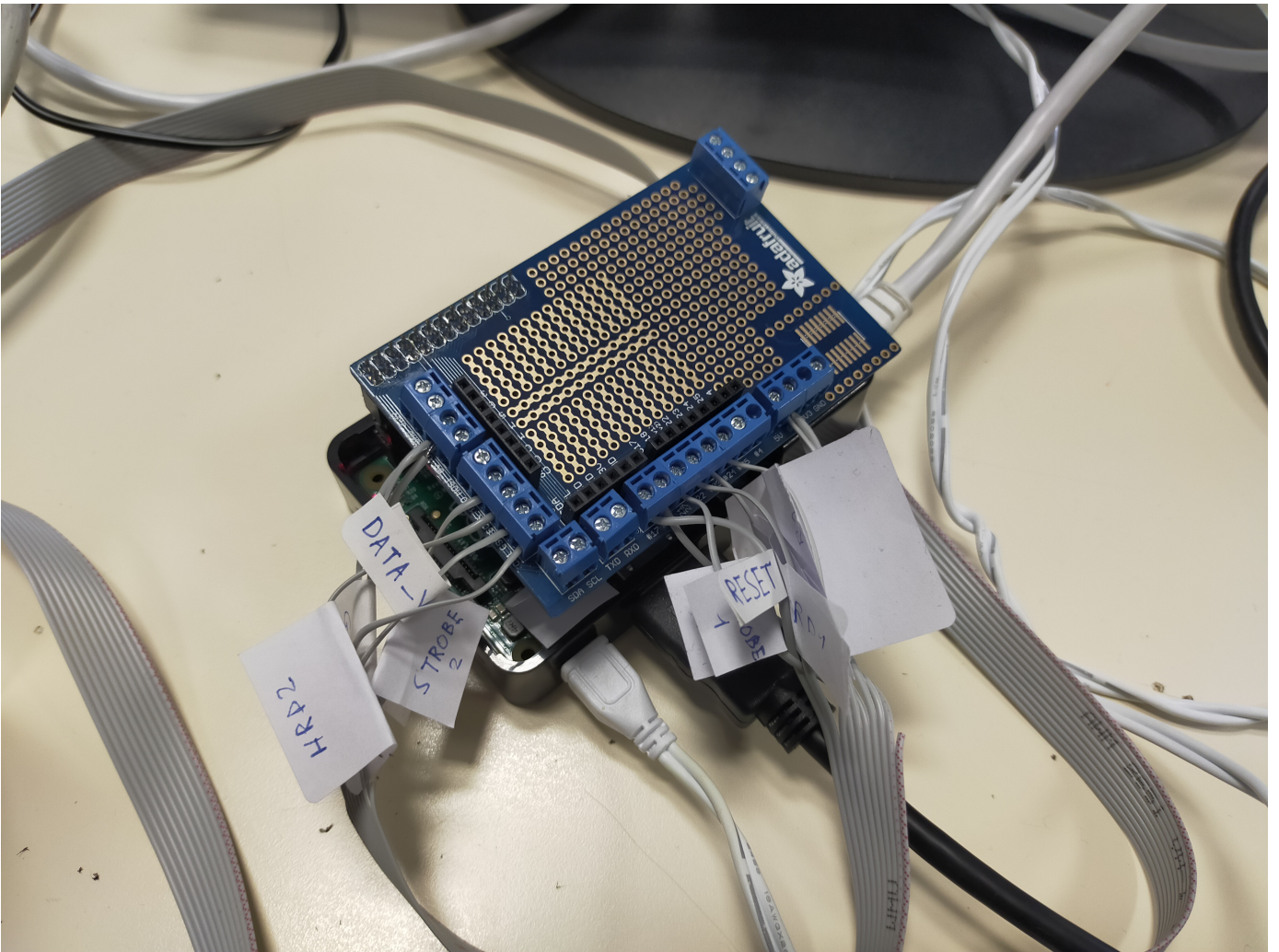
Hugo Cheveau  
Tester les entrées/sorties  
Comprendre l'interconnexion des différents matériels  
Fabrication du boîtier LED  
Mesurer/relever des signaux électriques

## Déroulement des activités

- Installer/configurer Raspbian Jessie  
Tout d'abord, il faut installer l'image iso de la Raspberry qui va lui être implémenter par une carte SD.  
On installe donc l'utilitaire win32 afin de pouvoir procéder à l'installation de l'image sur la carte SD.  
Suite à cela l'accès à la console de la Raspberry est disponible, mais il faut tout de même vérifier que la Raspberry soit bien à jour et que le SSH soit bien présent.  
Pour faire ceci il faut d'abord entrer la commande "apt-get update" qui met à jour les données de la Raspberry et "service ssh status" pour vérifier le SSH.
- Tester les entrées/sorties  
Nous avons tester la connectivité du panneau en utilisant un Oscilloscope  
Oscilloscope.png
- Comprendre l'interconnexion des différents matériels  
Vérification des liaisons effectué entre le panneau et la Raspberry
- Fabrication du boîtier LED  
Pour commencer nous nous sommes servi du carton comme support pour le panneau led en lui créant un espace suffisant en découpant certaines parties du carton et en enlevant le polystyrène.  
Ensuite nous avons relié les nappes à la carte mère du panneau, puis nous avons dénuder et souder des câbles qui font partis de l'alimentation.



Après cela nous avons mis des étiquettes sur chacune des broches de la Raspberry pour les identifier plus facilement et nous avons placé les deux câbles que nous avons préparés sur les broches 22 et 23



- Installer la chaîne de compilation

Afin de pouvoir mettre des programmes servant à afficher des messages sur le panneau led, on installe la chaîne de compilation CodeBlocks qui va permettre de réaliser les programmes ainsi que SmartTY qui va servir à entrer dans la console Raspberry pour pouvoir exécuter le programme dans le panneau led

- Tester la chaîne de compilation

A partir de Codeblocks, nous testons des programmes déjà préparés par les professeurs.

Comme par exemple un programme permettant au panneau d'afficher le message bonjour.

Pour cela il faut déjà télécharger le fichier c contenant ce programme, le mettre sur CodeBlocks et cliquer sur "build" puis faire un "rebuild" afin d'obtenir l'exécutable du fichier contenant le programme.

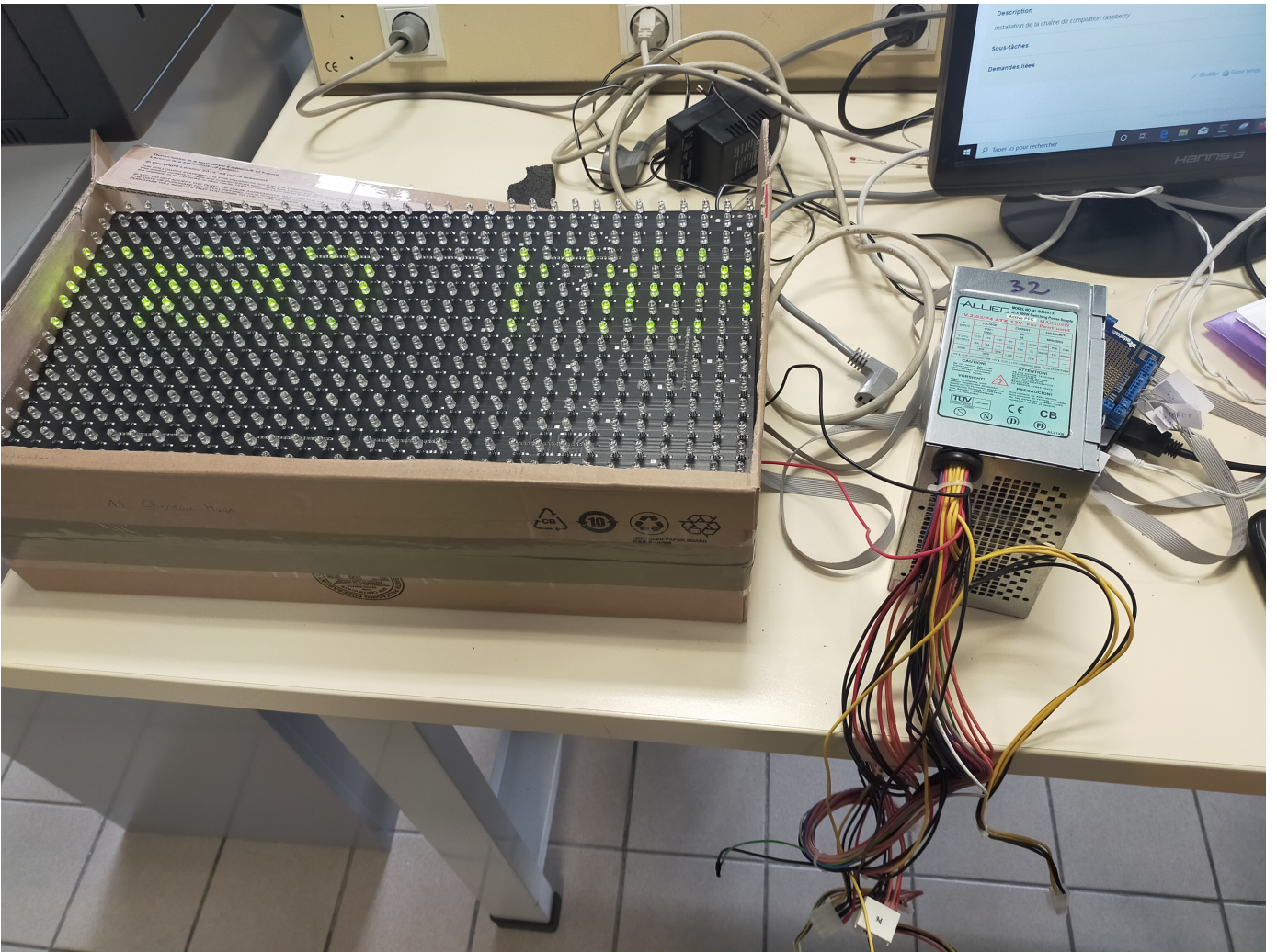
Ensuite il faut glisser l'exe du fichier sur la console de la raspberry et lui assigner les droits d'exécutions avec "chmod +x bonjour", le panneau affiche alors "bonjour".

codeblock1.png

codeblock2.png

codeblock3.png

- Mesurer/relever des signaux électriques  
On a utilisé un voltmètre pour être sûr que le panneau led fonctionne correctement  
Voltmètre.png
- Comprendre la commande d'une LED, d'une ligne de LED, et d'une rangée  
Nous avons dû comprendre comment un programme fonctionnait à travers le panneau led en visualisant si le chemin faisait une ligne, une rangée et aussi vérifier le temps de pause entre chaque allumage de led.
- Réaliser des algorithmes, puis coder, tester et déboguer  
En réalisant les mêmes étapes qu'avec les fichiers c de tests, il a fallut que nous modifions le programme afin de pouvoir afficher d'autres messages



## Programme en C/C++

Voici le code en C/C++ du programme correspondant au cas d'utilisation numéro 1

```

#include "panneauLeds2.h"

int main(int argc, char* argv[])
{
    //Index tableau 2D
    int i = 0;
    int j = 0;
    //Index du caractère + 1
    int k = 1;
    //Message : tableau 1D de caractères

    bool tabPanel[NLIGNES][NCOLONN];
    char message[100] = "BONNE ANNEE";

    for(i = 0; i < NLIGNES; i++)
    {
        for(j = 0; j < NCOLONN; j++)
        {
            tabPanel[i][j] = 0;
        }
    }
    //Index colonne pour un caractère de font5x7
    unsigned char cptRotation = 0;

    configPanel();
    clearPanel();

    while(1)
    {
        //Pour commencer à la première ligne
        pulse(RESET);
        //Mémoriser les états des leds avec les valeurs du tableau
        digitalWrite(STROBE1,HIGH);
        digitalWrite(STROBE2,HIGH);
        //Boucle de traitement d'une ligne
        for(i = 0; i < NLIGNES; i++)
        {
            for(j = 0; j < NCOLONN; j++)
            {
                if(tabPanel[i][j] == 1)
                {
                    digitalWrite(DATA_V1,LEDON);
                    digitalWrite(DATA_R1,LEDOFF);
                    pulse(HRD1);
                }
                else
                {
                    digitalWrite(DATA_V1,LEDOFF);
                    digitalWrite(DATA_R1,LEDOFF);
                    pulse(HRD1);
                }
            }

            //On active les leds de la ligne
            digitalWrite(OE,HIGH);
            delay(DELAY);
            //On change de ligne
            digitalWrite(OE,LOW);
        }
        /*CONVERSION CHAINE DE CARACTERES EN TABPANEL UTILISANT FONTSX7*/
        if(k < strlen(message) + 1)
        {
            processTab2DFromMessage(tabPanel,message,k,cptRotation);
            cptRotation++;
            //Chaque caractère est codé sur 5 colonnes et 7 lignes
            if(cptRotation % 5 == 0)
            {
                k++;
            }
        }
        //Permet l'affichage en boucle
        else
        {
            k = 1;
        }
    }
    return(0);
}

```



## Files

---

Raspberry.png	9.43 MB	20/02/2021	Theo Coignet
NappesLed.png	7.82 MB	20/02/2021	Theo Coignet
PanneauLed2.png	11.2 MB	20/02/2021	Theo Coignet
Voltmètre.png	31.8 KB	20/02/2021	Theo Coignet
Oscilloscope.png	27.2 KB	21/02/2021	Theo Coignet
codeblock1.png	53.3 KB	28/02/2021	Theo Coignet
codeblock2.png	48.3 KB	28/02/2021	Theo Coignet
codeblock3.png	43.5 KB	28/02/2021	Theo Coignet
codeUtilisateur1.png	63.4 KB	28/02/2021	Hugo Cheveau