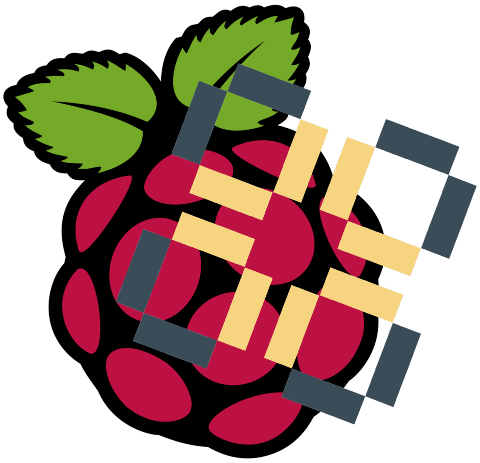
HepiaLight3

Raspberry Pi Pico

Université d'été 2024



Michael Divià (N° 22649552)
Alejandro Escribano (N° 15315914)
Gaspard Le Gouic (N° 19816289)

jeudi 27 juin 2024

Informatique et Systèmes de Communication

Table des matières

Table des figures

1 Introduction

Dans le cadre de l'**Université d'été 2024**, il nous a été proposé de réaliser un projet basé sur **HepiaLight 2**. Dans ce projet, nous devrons évaluer le portage de ce projet sur différentes nouvelles architectures afin de démontrer la possibilité de développement d'une nouvelle carte **HepiaLight 3**. Pour notre groupe, nous nous sommes vus attribué un **RP2040-PICO-HDR** ainsi qu'une matrice de LED numérique **NeoPixel NeoMatrix 8x8 - 64 RGB**. Nous allons donc devoir implémenter, en **MicroPython** toutes les fonctionnalités de base du projet **HepiaLight 2** en nous adaptant aux différentes limitations qu'apporte la nouvelle architecture **RP2040**.

2 Répartition des tâches initiales

Ne connaissant pas la difficulté de codage du **MicroPython** et du **RP2040-PICO-HDR** nous nous sommes initialement réparti les tâches du cahier des charges comme suit :

Set matrix : Michael Divià

Set pixel : Michael Divià

Set texte : Gaspard Le Gouic

Scroll text : Gaspard Le Gouic

init UART : Alejandro Escribano

Write UART : Alejandro Escribano

Read UART : Alejandro Escribano

Nous avions pour but initiale de terminer toute ces tâches avant la fin de la première semaine afin de pouvoir, par la suite, implémenter les fonctionnalités supplémentaires.

Cependant, dû à des limitations de temps externes entre autre, cette répartition à été modifiée de jour en jour afin de pouvoir avancer au mieux pour ce projet.

3 Librairie hepialight3

3.1 Branchements

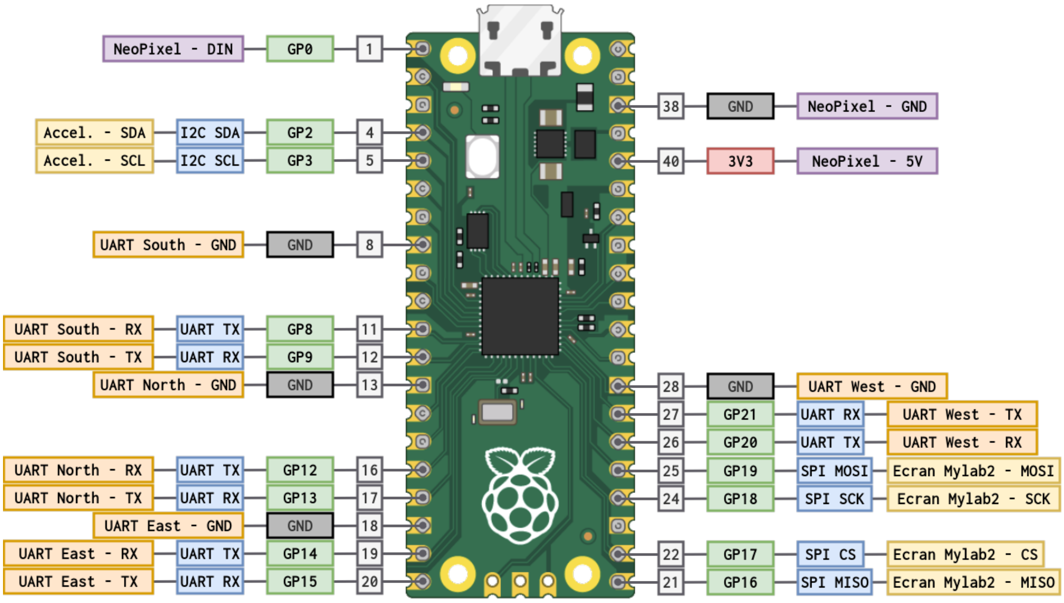


FIGURE 1 – Raspberry Pi Pico Pinout for HepiaLight3

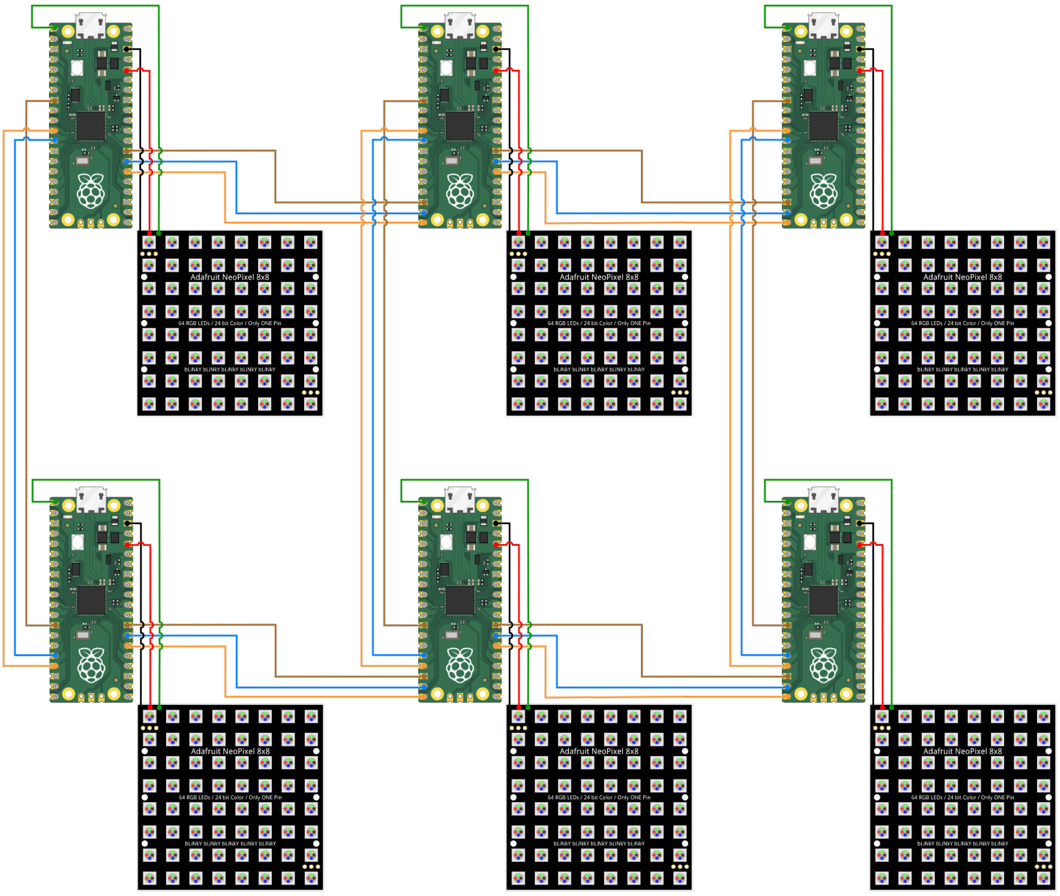


FIGURE 2 – UART communication between Raspberry Pi Pico Pinout for HepiaLight3

3.2 Affichage

Voici les fonctions à disposition en lien avec l'affichage sur la matrice de LED.

3.2.1 Predefined colors

Here are all the predefined colors that come in the class Color with there RGB values :

class Color :

```
Color.BLACK = (0, 0, 0)
Color.BLUE = (0, 0, 255)
Color.CYAN = (0, 255, 255)
Color.GREEN = (0, 255, 0)
Color.MAGENTA = (255, 0, 255)
Color.RED = (255, 0, 0)
Color.YELLOW = (255, 255, 0)
Color.WHITE = (255, 255, 255)
Color.RED_DARKER = (153, 0, 0)
Color.RED_DARK = (204, 0, 0)
Color.RED_LIGHT = (255, 102, 102)
Color.RED_LIGHTER = (255, 153, 153)
Color.BLUE_DARKER = (0, 0, 153)
Color.BLUE_DARK = (0, 0, 204)
Color.BLUE_LIGHT = (102, 102, 255)
Color.BLUE_LIGHTER = (153, 153, 255)
Color.GREEN_DARKER = (0, 153, 0)
Color.GREEN_DARK = (0, 204, 0)
Color.GREEN_LIGHT = (102, 255, 102)
Color.CYAN_DARK = (0, 204, 204)
Color.CYAN_LIGHT = (102, 255, 255)
Color.MAGENTA_DARKER = (153, 0, 153)
Color.MAGENTA_DARK = (204, 0, 204)
Color.MAGENTA_LIGHT = (255, 102, 255)
Color.YELLOW_DARKER = (153, 153, 0)
Color.YELLOW_DARK = (204, 204, 0)
Color.YELLOW_LIGHT = (255, 255, 102)
Color.GRAY_DARK = (64, 64, 64)
Color.GRAY = (128, 128, 128)
Color.GRAY_LIGHT = (192, 192, 192)
Color.ORANGE_DARK = (204, 102, 0)
Color.ORANGE = (255, 128, 0)
Color.ORANGE_YELLOW = (255, 204, 0)
```

3.2.2 Clear the matrix

Set the whole LED matrix to the specified color.

Usage :

```
Matrix.clear(Color)
```

Example :

```
Matrix.clear(Color.RED)
Matrix.clear((255, 0, 0))
Matrix.clear(0xFF0000)
```

Clearing the matrix in red.

Special case :

```
Matrix.clear(0)
```

Turn off the whole matrix.

3.2.3 Set a line of the matrix

Set a specific line of the LED matrix to the specified color.

Usage :

```
Matrix.set_line(line, Color)
```

Line must be between 0 and 7.

Example :

```
Matrix.set_line(5, Color.GREEN)
```

Setting line 5 (the 6th line) in green.

3.2.4 Set a column of the matrix

Set a specific column of the LED matrix to the specified color.

Usage :

```
Matrix.set_column(line, Color)
```

Column must be between 0 and 7.

Example :

```
Matrix.set_column(0, Color.YELLOW)
```

Setting line 0 (the 1st column) in yellow.

3.2.5 Set a LED of the matrix

Set a specific LED of the matrix to the specified color.

Usage :

```
Matrix.set_led(column, line, Color)
```

Position is represented by a column and line, where both must be between 0 and 7.

Example :

```
Matrix.set_led(3, 5, Color.ORANGE)
```

Setting LED at position column 3, line 5 in orange.

3.2.6 Get an LED color

Get the current color of a specific LED of the matrix.

Usage :

```
Matrix.get_led(column, line)
```

Position is represented by a column and line, where both must be between 0 and 7.

Example :

```
color = Matrix.get_led(7, 0)
```

Getting the color of the LED at column 7, line 0.

3.2.7 Display text

Display a scrolling message of the specified color using the specified delay in seconds.

Usage :

```
show_text(text, Color, speed)
```

Speed is in seconds.

Example :

```
show_text("PiPo", Color.RED, 0.1)
show_text("PiPo", Color.RED, 1/10)
```

scrolling «Pipo» message in red at a delay of 0.1 second.

3.2.8 Display image

Display a full matrix of colors.

Usage :

```
set_img(matrix)
```

Example :

```
set_img(""" .RR..RR.
            RRRRRRRR
            RRRRRRRR
            RRRRRRRR
            RRRRRRRR
            .RRRRRR.
            ..RRRR..
            ...RR...""")
```

Available colors :

```
R : red
G : green
B : blue
C : cyan
V : violet
Y : yellow
W : white
. : black
```

3.3 Communication

3.3.1 UART

Transmit or reveice data via UART with up to 4 other cards.

Usage :

```
Uart(Direction, baudrate, parity, bits, stop)
```

Create a UART communication

```
x.send(string)
```

Send data

```
x.sendline(string)
```

Send line of data. '\n' will be added at the end of the string automaticaly.

```
x.receive(length)
```

Wait until you received the ask length of data.

```
x.receiveline(length)
```

Wait until you received the full line of data (ending with a '\n').

Example :

```
uart_north = Uart(Direction.NORTH)
uart_south = Uart(Direction.SOUTH)
uart_east = Uart(Direction.EAST)
uart_west = Uart(Direction.WEST)

uart_east.sendline("123")
uart_west.send("45")
data_north = uart_north.receive(3)
data_south = uart_south.receiveline()
print(f"Data received from North: {data_north}")
print(f"Data received from South: {data_south}")
```

Default values are :

baudrate : 9'600

parity : None

bits : 8

stop : 1

Each Direction Pin Out can be found on figure 1

3.4 Capteurs externes

4 Conclusion