

Projet Collectif

Analyse, résolution et remplissage automatique
d'une grille de Sudoku

FALCONNIER Pierre - PAYSAN Jérémie - GIRARDEY Martin - LIORET Nathan



Contexte et cahier des charges:

Résolution de la grille :

- sans erreur
- temps court (<2min)
- ∇ dimension (< 20cm x 20cm)
- ∇ orientation
- ∇ difficulté
- lisibilité des chiffres

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

Exemple de grille de Sudoku

Matériel et choix de conception:

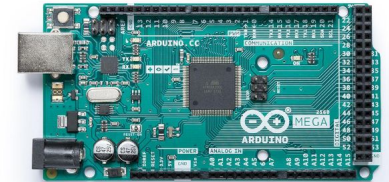
- 1 table traçante XY
- 1 carte Raspberry Pi 4
- 1 disque dur SSD
- 1 carte Arduino Mega 256
- 1 caméra (v2)
- 2 CNA
- 1 bouton poussoir
- 1 plaque électronique



Table traçante



Raspberry Pi 4

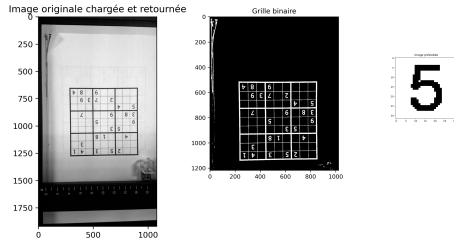


Arduino Mega 2560

Organisation et répartition du travail

Pierre Falconnier
Jérémie Paysan

Traitement d'Image

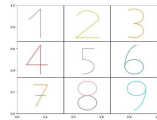
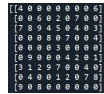


Nathan Lioret

Résolution du Sudoku

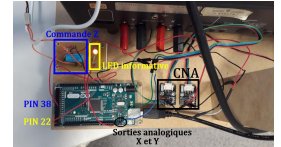
Allure des chiffres

Communication Raspberry-Arduino



Martin Girardey

Partie électronique/mécanique





Traitement d'image



Acquisition de l'image

Caméra Raspberry

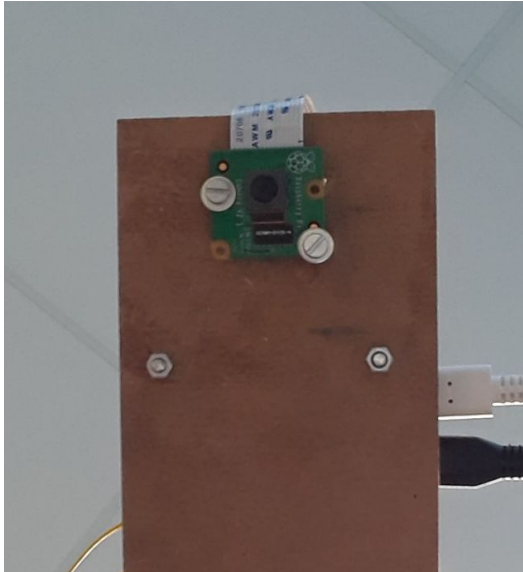
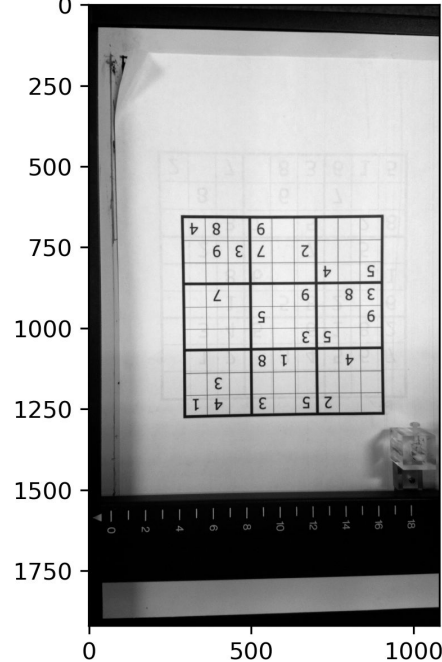
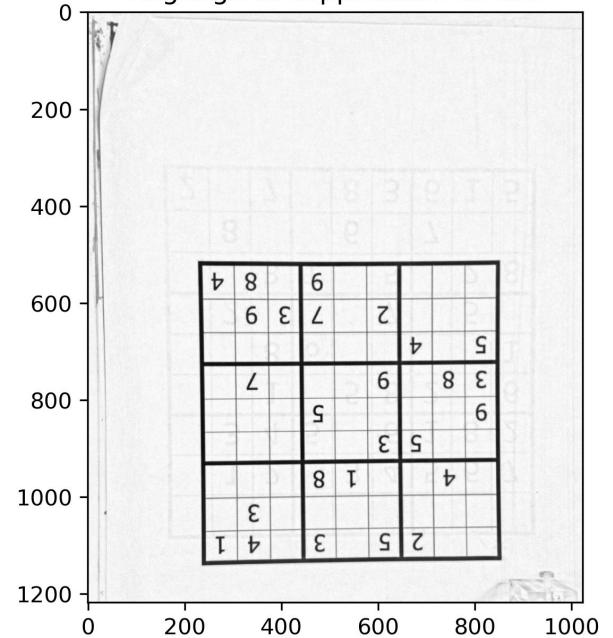


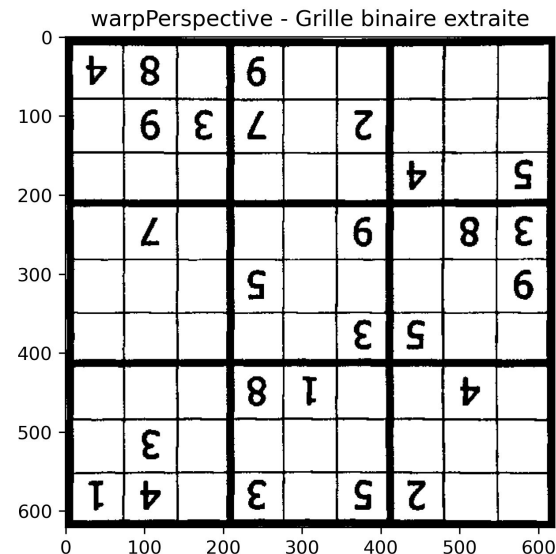
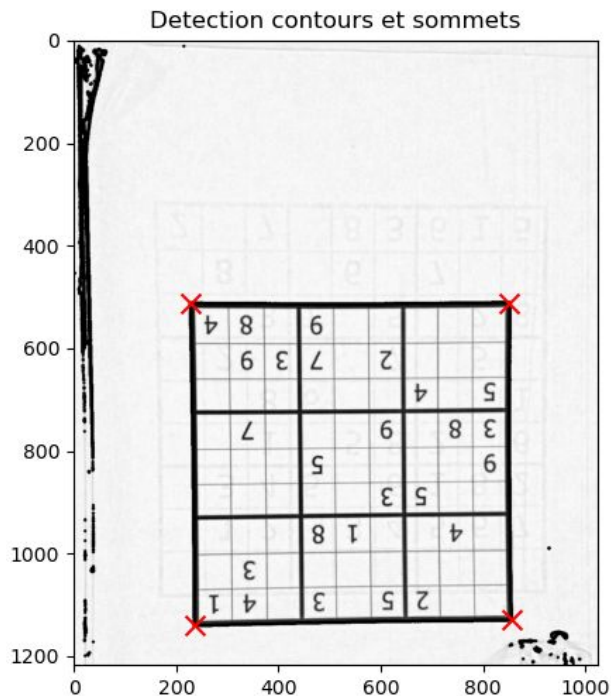
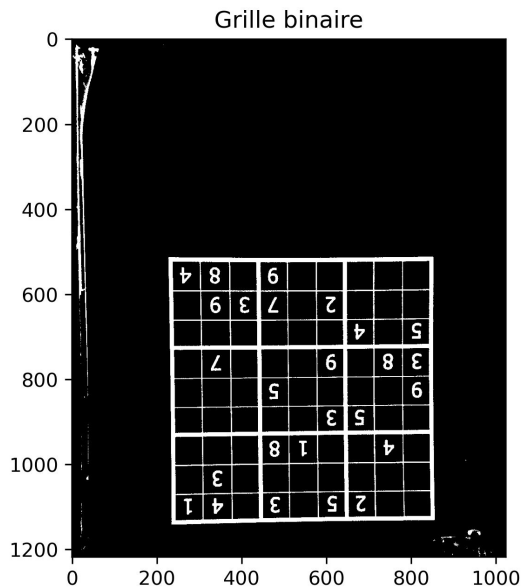
Image originale chargée et retournée



Rognage et suppression ombre



Détection et extraction de la grille



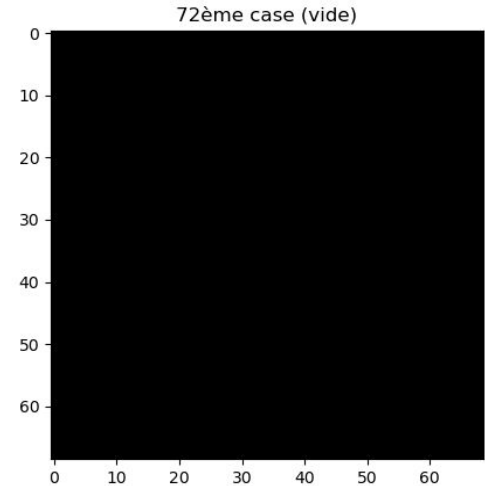
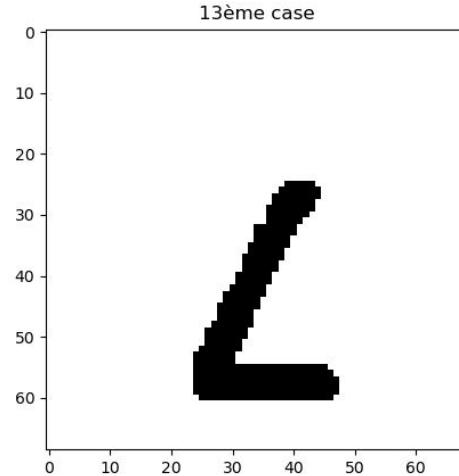
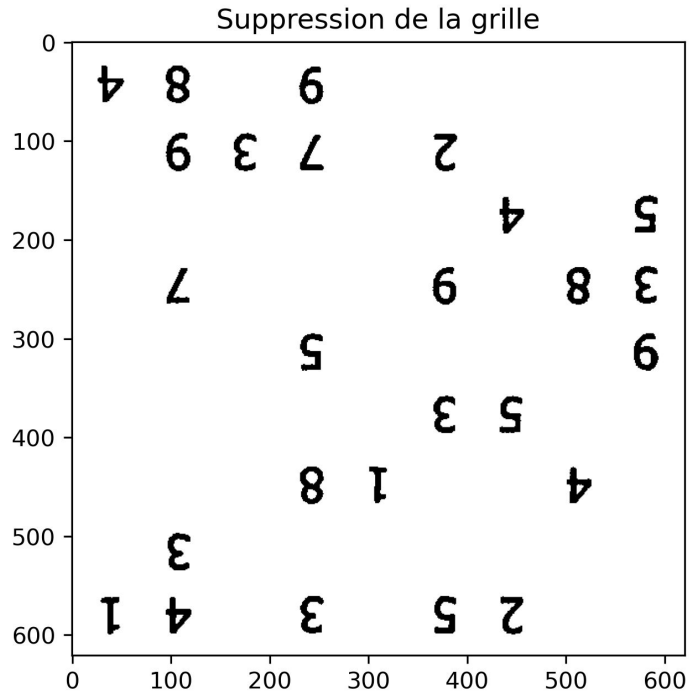
Position des coins :

```
[[ 228. 513.]
 [ 851. 512.]
 [ 235. 1141.]
 [ 855. 1130.]]
```

Angle initiale :

```
1.0164313830751661
```

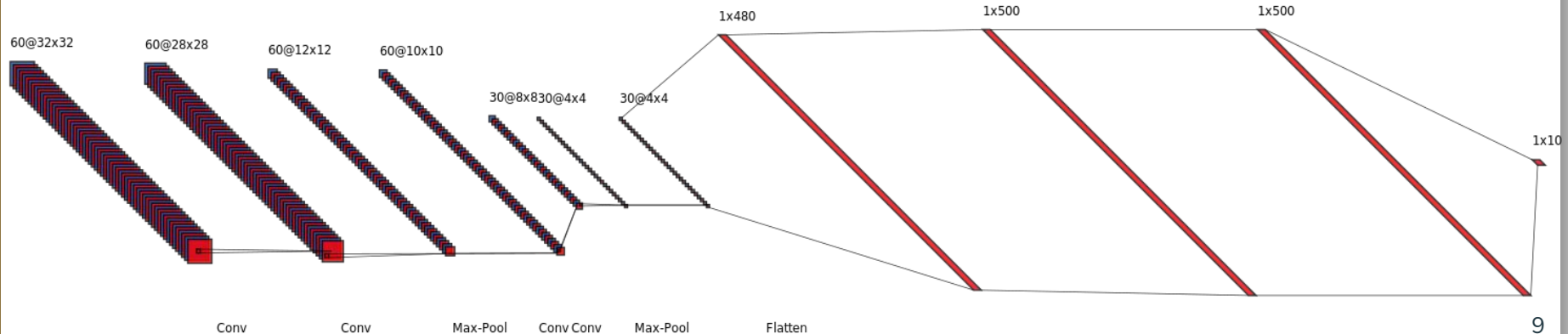
Extraction des cases et orientation de la grille



Réseaux de neurones convolutif (CNN)

2 réseaux :

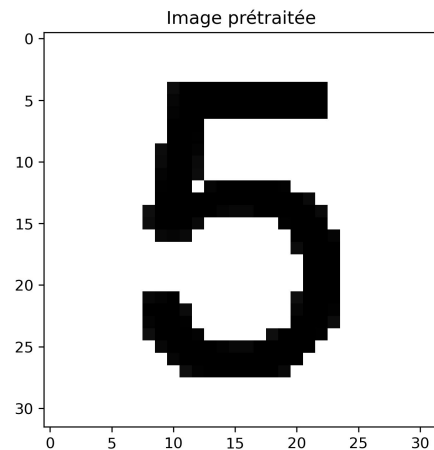
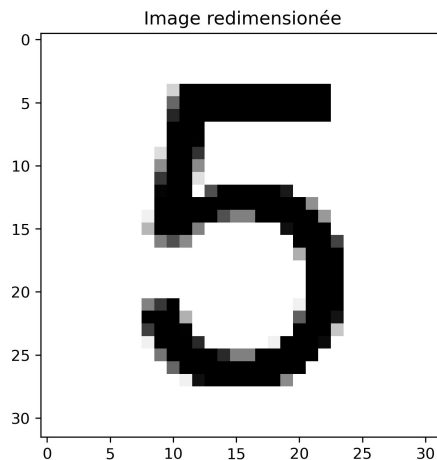
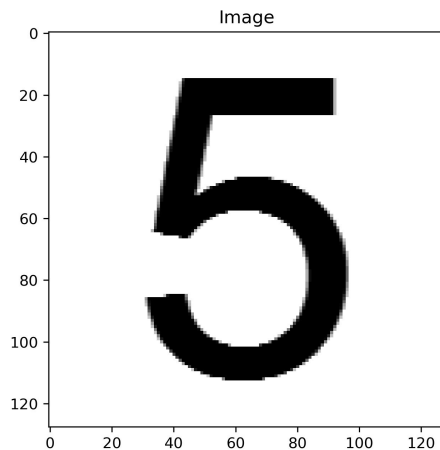
- Classification des chiffres : 0 1 2 3 4 5 6 7 8 9
- Classification de l'orientation : 0° ou 180°



Données :

3 8 7 5 9

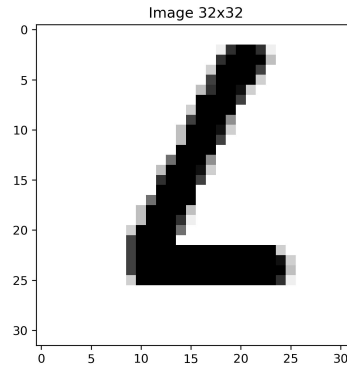
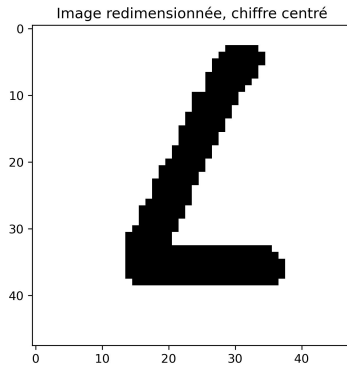
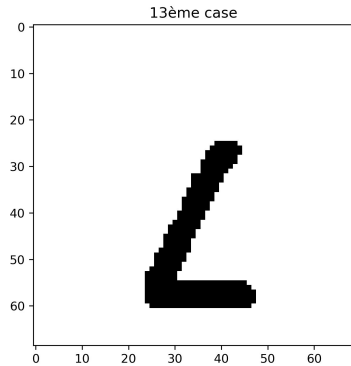
Prétraitement :



Entraînement et résultats :

- Classification des chiffres : 94,64 %
- Classification de l'orientation : 98,89 %

Retour au sudoku : prétraitement et classifications



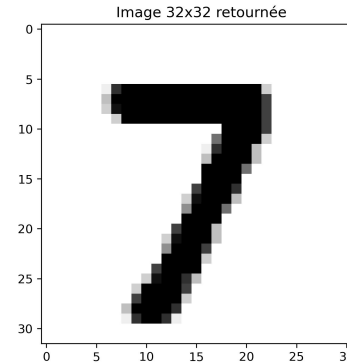
CNN 1

Reconnaissance orientation :

```
[[1 0 0 0 0 0 0 0]
 [0 0 1 1 0 1 0 0]
 [0 0 0 0 0 0 1 0]
 [0 1 0 0 0 0 0 1]
 [0 0 0 1 0 0 0 0]
 [0 0 0 0 0 1 1 0]
 [0 0 0 0 0 0 0 1]
 [0 1 0 0 0 0 0 0]
 [1 1 0 1 0 1 1 0]]
```

Grille à l'endroit

		2	5		3		4	1
							3	
	4			1	8			
		5	3					
9					5			
3	8		9				7	
5		4						
			2		7	3	9	
				9		8	4	



CNN 2

Reconnaissance chiffres :

```
[[0 0 2 5 0 3 0 4 1]
 [0 0 0 0 0 0 0 3 0]
 [0 4 0 0 1 8 0 0 0]
 [0 0 5 3 0 0 0 0 0]
 [9 0 0 0 0 5 0 0 0]
 [3 8 0 9 0 0 0 7 0]
 [5 0 4 0 0 0 0 0 0]
 [0 0 0 2 0 7 3 9 0]
 [0 0 0 0 0 9 0 8 4]]
```

Angle final :
181.01643138307517

Résolution du Sudoku par Backtracking

3			
			2
4		2	3
			1

Grille à résoudre

L'algorithme
test le 1

Pas le choix

3	1	4	/
			2
4		2	3
			1

Aucune
possibilité

Pas le choix

On test donc
avec le 2

Pas le choix

3	2	1	4
1	4	3	2
4	1	2	3
2	3	4	1

Pas le choix

Une possibilité à chaque fois
=> on obtient la solution



Communication entre les différents dispositifs



Communication RaspBerry → Table traçante



RaspBerry Pi 4

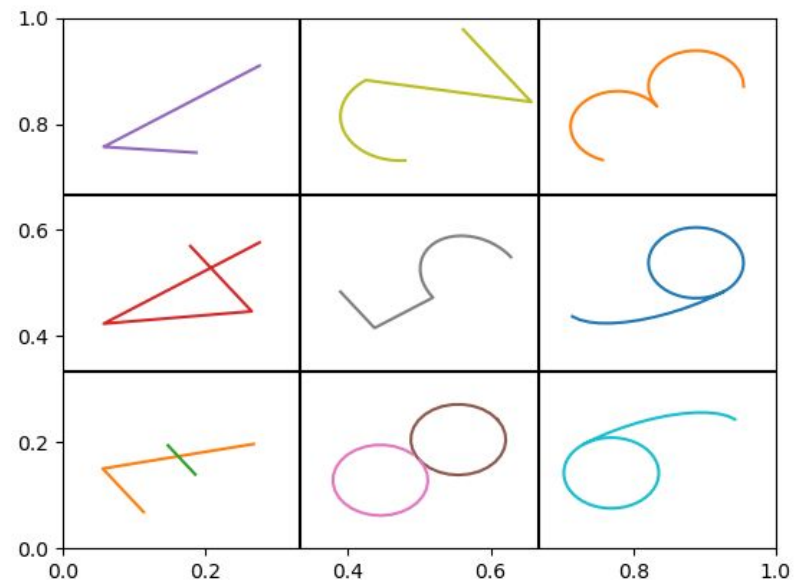
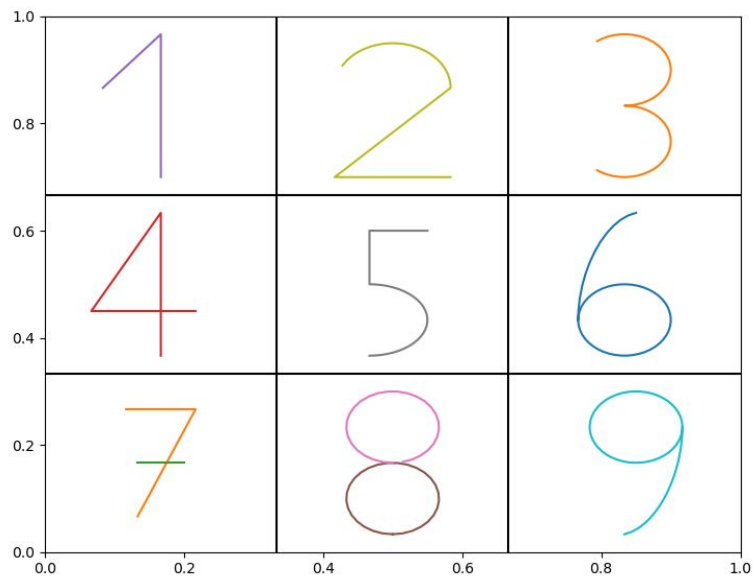


Arduino Mega 2560

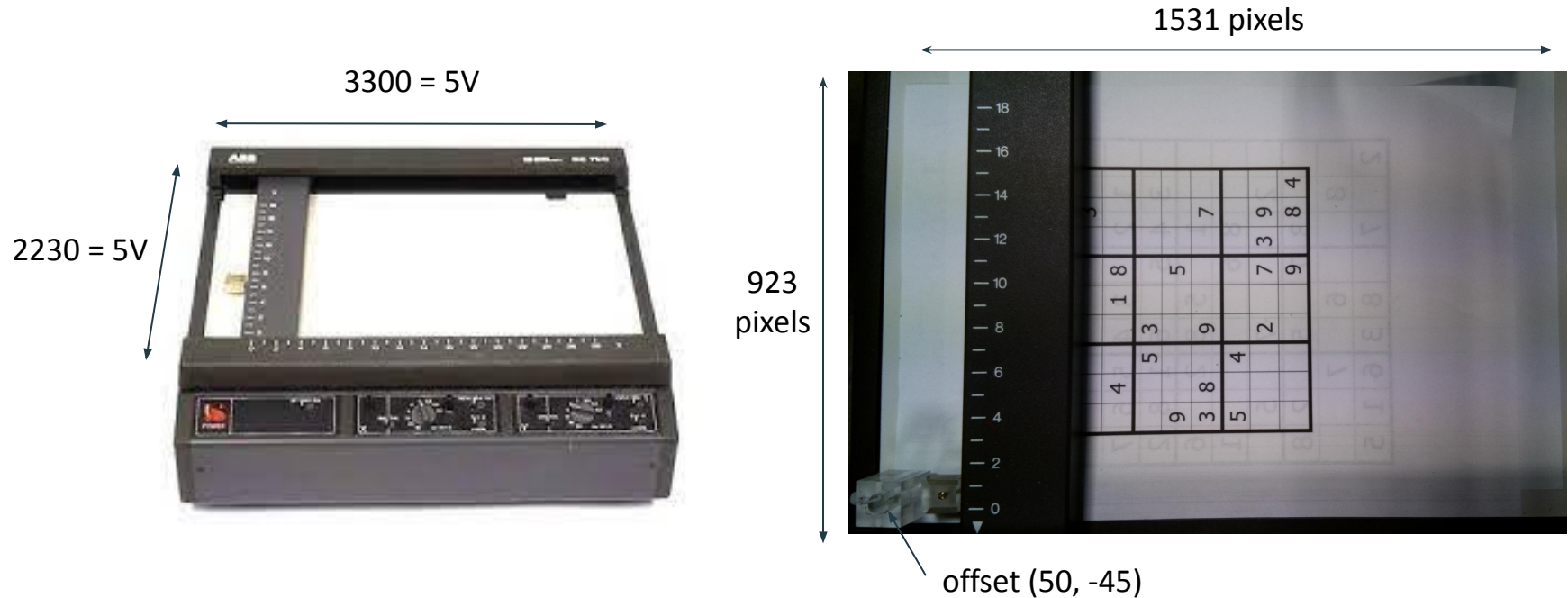


Table traçante

Allure des chiffres

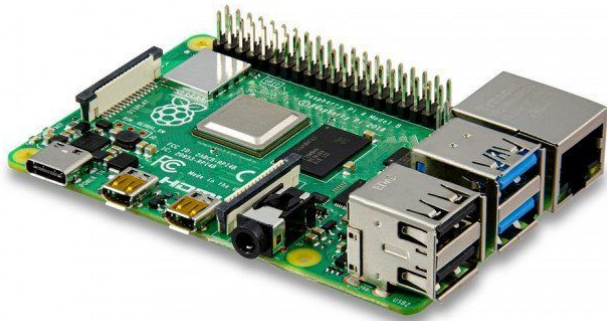


Correspondance image / tensions



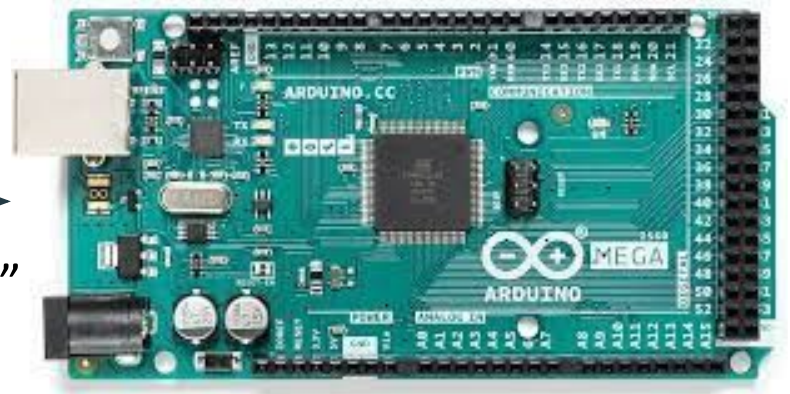
$$(x_{table}, y_{table}) = (3300 \times (x_{image} + 50)/1531, 2230 \times (y_{image} - 45)/923)$$

Communication RaspBerry -> Arduino



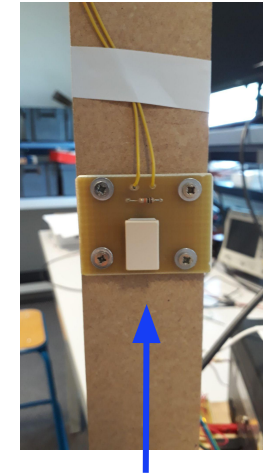
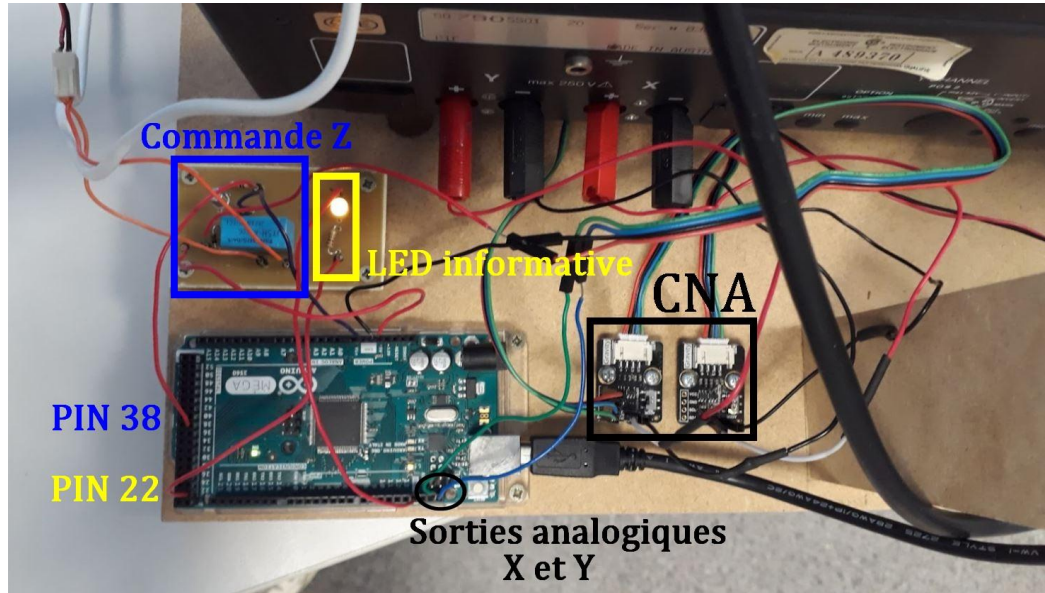
RaspBerry Pi 4

→
“xxxx,yyyy,z”



Arduino Mega 2560

Communication RaspBerry → Table traçante



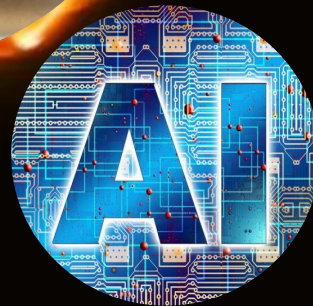
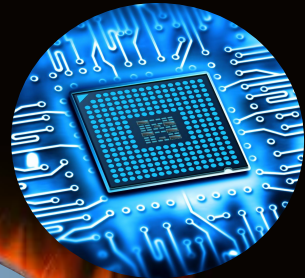
Bouton démarrage



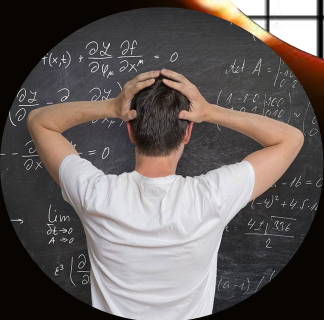
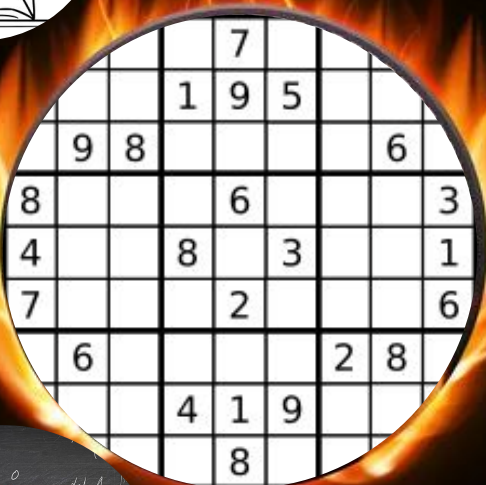
Conclusion



SOLVEUR DE SUDOKU



VS



RÉSOLVÉZ VOS SUDOKUS EN UNE MINUTE SANS ÉCRIRE NI RÉFLÉCHIR²⁰ !

Démonstration

