

# Fusion du visible et de l'invisible

**Yann Ducruy**  
**Rémi Segretain**

**M2GI - M2M - 2019**

# Matériels utilisés

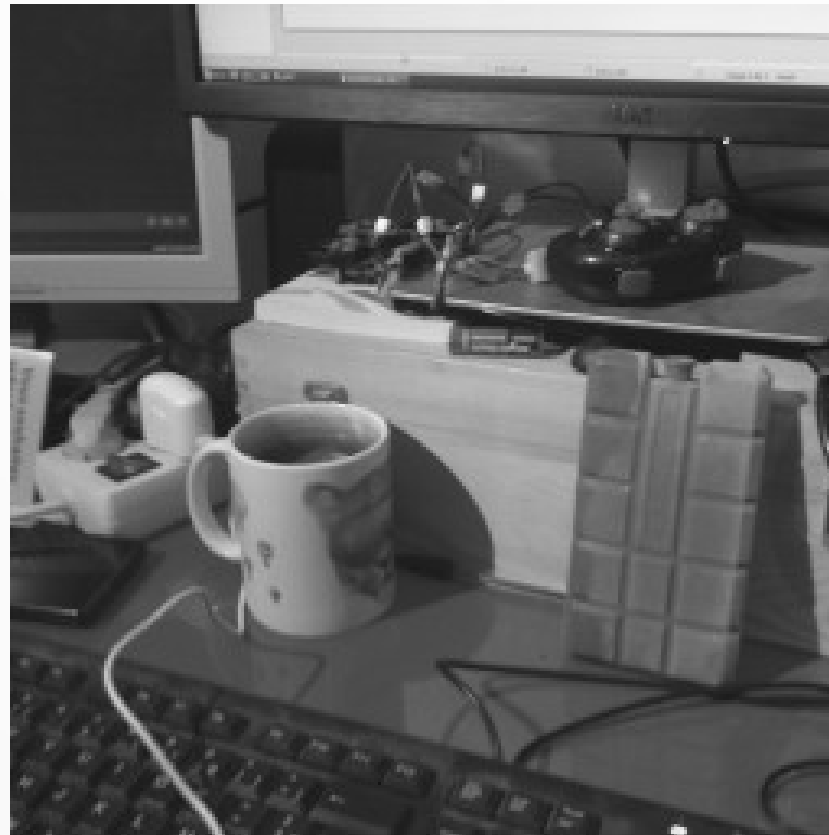
- Raspberry pi 3B
- Caméra de raspberry pi V1.3
- Capteur thermique : SparkFun Grid-EYE Infrared Array Breakout
- ~150 lignes de C++

# Utilisation du capteur thermique

- Difficultés à trouver les librairies nécessaire
- Transformation d'une librairie Arduino en librairie Raspberry (via utilisation de WiringPi) pour le capteur thermique

# Algorithme de fusion

- On prend une photo en niveaux de gris de  $256 \times 256$  pixels (via raspistill) et on lit le fichier généré.



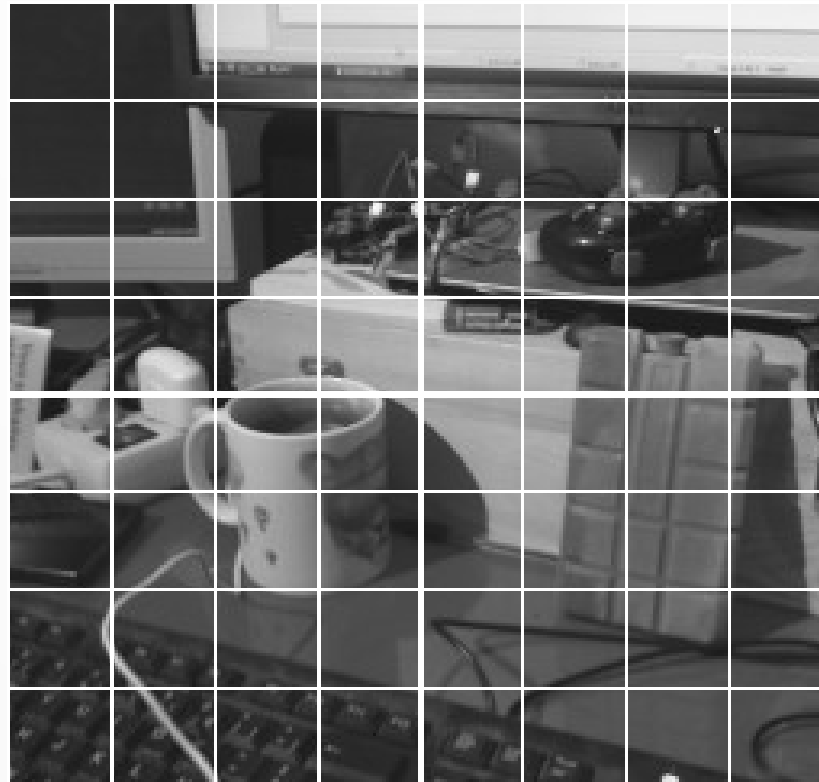
# Algorithme de fusion

- On récupère une image de température de 8\*8 pixels

21.75	22.25	23.00	21.00	19.25	18.50	18.00	18.25
24.00	24.00	23.50	20.75	14.25	13.50	16.25	19.00
25.50	24.50	23.00	20.50	15.25	10.25	14.75	20.00
23.00	23.00	22.75	20.50	23.75	24.25	21.25	21.00
22.25	22.75	22.75	21.50	32.25	43.75	25.50	21.00
24.00	25.50	24.75	24.50	23.00	24.50	23.00	21.25
24.00	26.25	25.00	24.25	21.25	21.75	21.75	21.25
21.50	26.00	25.00	22.75	20.00	20.75	20.50	20.25

# Algorithme de fusion

- Pour chaque pixel de température on associe une valeurs de rouge
- L'image visible est découpé en bloc 64 blocs de  $32 \times 32$  pixels, chaque pixel d'un bloc se voit attribuer en niveau de rouge la valeur calculée par rapport à la température.

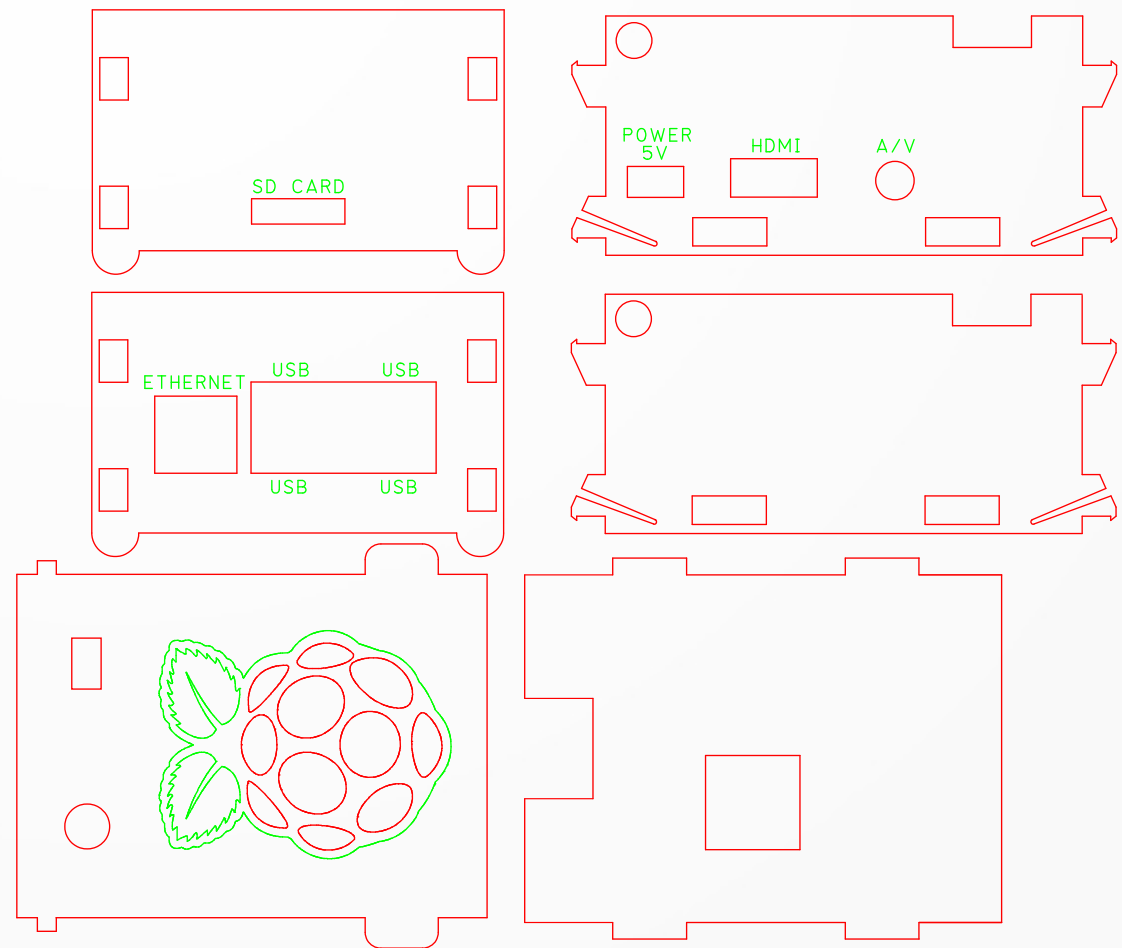


# Algorithme de fusion

- Écriture de l'image dans un nouveau fichier.
- Récupérable via SFTP.
- Lecture et écriture des images via les librairies stb : <https://github.com/nothings/stb>

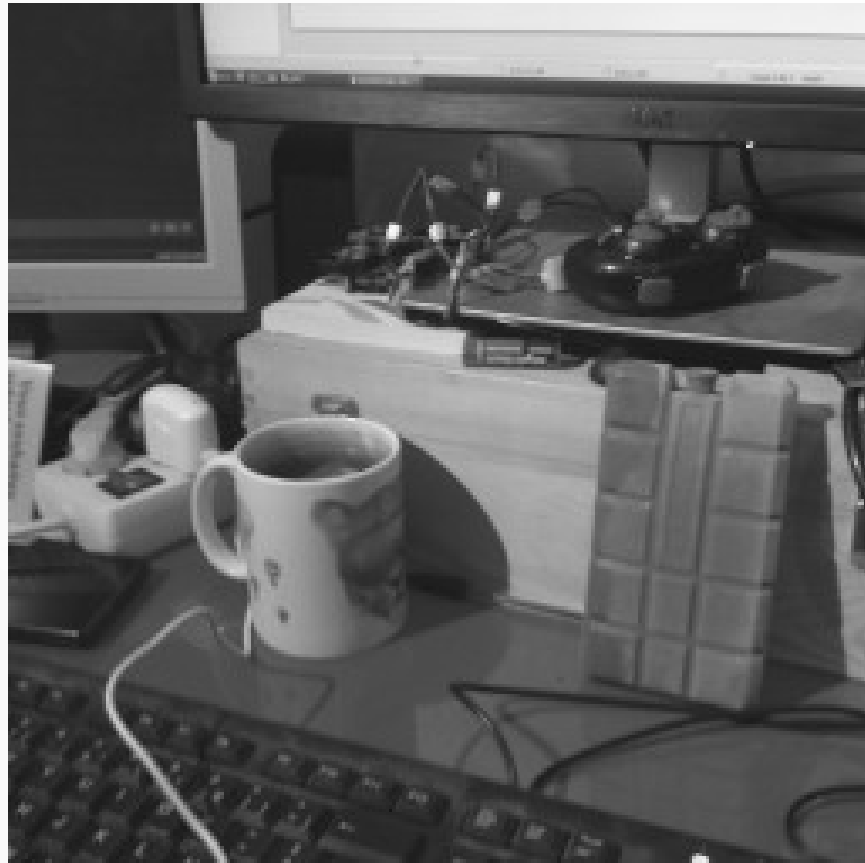


# Le boîtier





# Images résultat



# Conclusion

- Amélioration : convolution de l'information de température pour éviter l'effet pixel