

Überwachungseinheit

Die Überwachungseinheit besteht aus zwei Komponenten: der Objekterkennung, welche Rauch und Feuer erkennt und einem intelligenten Schaltsystem, das einen energetisch effizienten Haushalt ermöglicht.

Objekterkennung

Benötigt wird: mind. Raspberry Pi Zero 2 W, am Besten - Camera Module 3 Wide sowie USB-C power; 5 V mit 3 A (15W)

Das Yolo-Modell kann über Github heruntergeladen werden (https://github.com/FIRENET-AI/FIRENET-AI.github.io/main/code/FASDD_nano_val_interval_2_80ep_best.pt):

Nachdem das Modell heruntergeladen wurde, ist es notwendig, ein Betriebssystem für den Pi herunterzuladen. Für den Raspberry Pi Zero 2 W bietet sich die 32-bit Variante an (<https://www.raspberrypi.com/software/operating-systems/>) für den Pi 4 oder höher kann man auch auf die 64-bit Variante zurückgreifen. Danach muss eine Kamera an den Pi angeschlossen werden, dies gelingt etwa über USB oder über ein Flachbandkabel, abhängig von der Kamera.

Die Kamera kann auf Funktion über `rpicam-hello -t 0` getestet werden.

Anschließend lässt sich das Hauptskript über Github herunterladen (https://github.com/FIRENET-AI/FIRENET-AI.github.io/main/code/Fire_Detection_BOT.py).

Um das Skript nutzen zu können, muss ein Telegram-Bot erstellt werden. Dieser lässt sich über Telegram unter BotFather konfigurieren (<https://core.telegram.org/bots/tutorial>). Nachdem man ein solchen Bot erstellt hat, muss man die Variable `Bot_Token` und `Chat_ID` im Skript mit dem eigenem `Bot_Token` und `Chat_ID` ergänzen.

Nachdem das Skript heruntergeladen und auch auf Funktion getestet wurde, ist es nun möglich, ein Bild auf Rauch und Flammen zu detektieren. Über den Chat ist es ebenfalls möglich über `sendphotos=true` einen Testmodus

einzuschalten, in dem jede Detektion unabhängig vom Ergebnis gesendet wird.

Damit das Skript autonom beim Reboot auf dem Pi läuft, muss es unter crontab -e hinzugefügt werden (<https://de.wikipedia.org/wiki/Cron>).

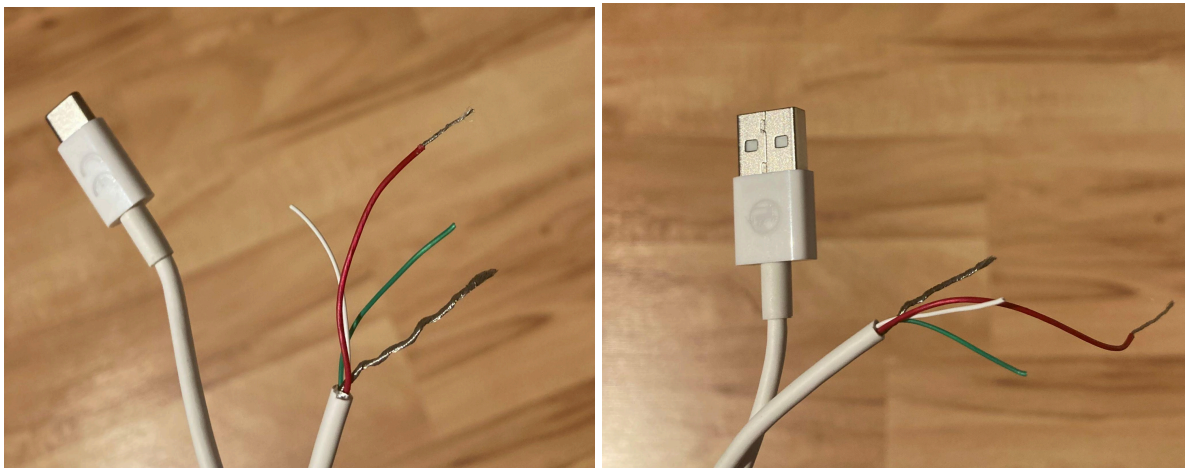
Intelligenten Schaltsystem

Benötigt wird : Raspberry Pi Pico, Jump-Wires, Breadboard (oder Protoboard), Relay Modul, USB-A zu USB-C bzw. USB-A zu Mikro-USB (abhängig vom Pi) und ein USB-C zu Mikro-USB Kabel.

Der Code für den Pico lässt sich über Github herunterladen (<https://github.com/FIRENET-AI/FIRENET-AI.github.io/main/code/pico.py>).

Damit der Code funktioniert, muss MicroPython (<https://micropython.org/>) auf dem Pico installiert werden. Die Installation gelingt über Thonny (<https://thonny.org/>). Danach kann der Code als `main.py` auf dem Pico lokal gespeichert werden.

Nachdem das Skript auf dem Pico läuft, muss der Pico mit dem Relay Modul verbunden werden. Dafür schneidet man ein USB-C Kabel in der Mitte durch.



VON	ZU
-----	----

Über die ON_Time Variable kann die Zeit, die der Raspberry Pi an ist, gesteuert werden. Gleichmaßen kann über OFF_Time die Zeit, die der Pi aus gesteuert werden.

PS: Zukünftig wird eine weitere Variante veröffentlicht, die die Schaltung mithilfe von Sensorik adaptiv anpassen kann.