دليل إعداد Raspberry Pi

نظام التحكم الأرضي للطائرات — Drone Ground Control System

بث الكاميرا، MAVLink Router، التحكم عن بُعد، والتحكم بالمحركات عبر TX12

مايو 2026

فهرس المحتويات

1. متطلبات الأجهزة
2. الجزء الأول — تثبيت نظام التشغيل Raspberry Pi OS
3. الجزء الثاني — الاتصال بـ Raspberry Pi عبر SSH
4. الجزء الثالث — إعداد الكاميرا (مقارنة الخيارات)
5. الجزء الرابع — بث الفيديو إلى QGroundControl
6. الجزء الخامس — إعداد بث الكاميرا (المتغيرات)
7. الجزء السادس — MAVLink Router للتحكم بالطائرة
8. الجزء السابع — البث عن بُعد عبر Tailscale
9. الجزء الثامن — التحكم بالمحركات عبر TX12 و Sabertooth
10. الجزء التاسع — تحديد قنوات RC على Cube
11. استكشاف الأخطاء وإصلاحها
12. المراجع والروابط

متطلبات الأجهزة

الأساسيات

- Raspberry Pi 5
- بطاقة MicroSD (16 جيجابايت أو أكثر)
- قارئ بطاقة MicroSD عبر USB
- مصدر طاقة (USB-C، 5V/5A لـ RPi 5)
- جهاز كمبيوتر (Mac أو Windows أو Linux)

الكاميرا

- كاميرا Dahua IP Camera (DH-IPC-HFW2431SP-S-S2) — الكاميرا المستخدمة حالياً ✅
- كابل Ethernet RJ45 (توصيل الكاميرا بالـ Pi)
- مصدر طاقة 12V للكاميرا أو POE injector

التحكم الطيراني والمحركات

- Cube Orange+ (وحدة التحكم الطيراني / Flight Controller)
- Radiomaster TX12 (جهاز التحكم عن بُعد)
- Sabertooth HS Hybrid (متحكم المحركات)
- محركات 12V DC (عدد 2)
- مصدر طاقة 12V للمحركات

الجزء الأول — تثبيت نظام التشغيل

الخطوة الأولى: تحميل Raspberry Pi Imager

حمّل وثبّت برنامج **Raspberry Pi Imager** (https://www.raspberrypi.com/software/) على جهازك.

الخطوة الثانية: ضبط الإعدادات قبل الكتابة

افتح البرنامج، اختر الجهاز ونظام التشغيل، ثم انقر على **Next**. سيعرض عليك البرنامج تخصيص الإعدادات — اختر **Edit Settings** وأدخل القيم التالية:

الحقل	القيمة
اسم الجهاز (Hostname)	`lab2` (أو أي اسم تفضله)
اسم المستخدم (Username)	`lab2`
كلمة المرور	كلمة مرور قوية
اسم شبكة WiFi	اسم شبكتك
كلمة مرور WiFi	كلمة مرور شبكتك
تفعيل SSH	✅ مفعّل (تبويب Services)
نوع المصادقة	مفتاح عام — SSH public key (موصى به)

> ملاحظة: لا يشترط أن يكون اسم الجهاز واسم المستخدم متطابقَين، لكن التطابق يجعل الاتصال أسهل.

الخطوة الثالثة: تفعيل Raspberry Pi Connect

في تبويب **General** بنفس نافذة الإعدادات، فعّل خيار **Raspberry Pi Connect** وسجّل دخولك بحساب Raspberry Pi لربط الجهاز تلقائياً.

> هذه الميزة تتيح لك الوصول إلى سطر الأوامر عبر المتصفح من أي مكان — مفيدة جداً إذا لم يعمل SSH بعد أو احتجت للوصول عن بُعد دون إعداد إضافي.

اختياري الخطوة الرابعة: إضافة مفتاح SSH العام

هذه الخطوة اختيارية وتُغني عن إدخال كلمة المرور في كل مرة تتصل فيها عبر SSH. إذا أردت تفعيلها، افتح الطرفية (Terminal) على جهازك ونفّذ الأمر التالي لعرض مفتاحك العام:

```
cat ~/.ssh/id_rsa.pub
```

إذا لم يكن لديك مفتاح SSH بعد، يمكنك إنشاء واحد باتباع هذا الدليل من GitHub (https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent)، أو بتنفيذ:

```
ssh-keygen -t rsa -b 4096
```

انسخ الناتج والصقه في حقل **مفتاح SSH العام** في إعدادات RPi Imager.

الخطوة الخامسة: كتابة بطاقة SD

انقر على **Save** ثم **Yes** للبدء بالكتابة. انتظر حتى يكتمل النقل، أدخل البطاقة في Raspberry Pi وشغّله.

بعد الإقلاع، يمكنك الوصول إليه من: `connect.raspberrypi.com` (https://connect.raspberrypi.com)

الجزء الثاني — الاتصال عبر SSH

الخطوة الأولى: الاتصال مباشرة باسم الجهاز

بعد إقلاع الجهاز (انتظر دقيقة إلى دقيقتين)، افتح الطرفية واتصل باستخدام:

```
ssh lab2@lab2.local
```

> الصيغة دائماً: `username@hostname.local` — يعمل هذا تلقائياً على نفس شبكة WiFi دون الحاجة لمعرفة عنوان IP.

الخطوة الثانية: إذا لم يعمل اسم الجهاز

ابحث عن عنوان IP يدوياً عبر مسح الشبكة:

```
for i in $(seq 1 254); do (nc -zv -w 1 192.168.0.$i 22 2>&&1 | grep -q succeeded && echo "SSH open: 19
2.168.0.$i") & done; wait
```

أو تحقق من صفحة إدارة الراوتر على `192.168.0.1` وابحث عن الجهاز في قائمة الأجهزة المتصلة.

ثم اتصل باستخدام IP:

```
ssh lab2@192.168.0.127
```

> نصيحة: لضمان نفس IP دائماً، احجز IP ثابتاً في إعدادات DHCP بالراوتر باستخدام MAC address الخاص بـ Pi.

الجزء الثالث — إعداد الكاميرا

يوجد خياران للكاميرا تم توثيقهما. الخيار المستخدم حالياً هو كاميرا Dahua IP Camera.

مقارنة الخيارات

الخاصية	HQ Camera (IMX477)	Dahua IP Camera ✅ مستخدمة حالياً
الاتصال	كابل CSI ribbon → Pi	Ethernet (RJ45)
الترميز	معالج Pi (~35% لكل نواة)	شريحة داخلية (0% من Pi)
الدقة	حتى 720p (لتقليل التأخير)	4MP أصلي
الطاقة	من GPIO الخاص بـ Pi	POE أو 12V
رؤية ليلية IR	لا	نعم
تعقيد الإعداد	متوسط	بسيط
الاستخدام الموصى	الاختبار / التعلم	الاستخدام في الطائرة

الخيار أ — كاميرا Raspberry Pi HQ Camera (IMX477) — قديمة

> هذا كان الإعداد الأصلي. يعمل لكنه يُحمّل عبء الترميز على الـ Pi. تم الانتقال إلى كاميرا Dahua (الخيار ب) بدلاً منه.

1. **توصيل الكاميرا:** وصّل كاميرا HQ بالـ Pi عبر كابل CSI ribbon. تأكد أن الكابل مُدخل بالكامل والمشبك مقفل.

2. **تثبيت الحزم المطلوبة:**

```
sudo apt update
sudo apt install -y rpicam-apps libcamera-apps gstreamer1.0-tools \
  gstreamer1.0-plugins-base gstreamer1.0-plugins-good \
  gstreamer1.0-plugins-bad gstreamer1.0-libav gstreamer1.0-libcamera \
  v4l-utils
```

3. **التحقق من اكتشاف الكاميرا:**

```
rpicam-hello --list-cameras
```

4. **تشغيل البث:**

```
bash scripts/start_stream.sh
```

مسار البث:

```
rpicam-vid → /tmp/h264.pipe → GStreamer → UDP → QGC
```

مشاكل شائعة مع HQ Camera

المشكلة	الحل
صورة ضبابية	أدر العدسة يدوياً أثناء مشاهدة البث
لون بنفسجي/أرجواني	فلتر IR cut مفقود أو غير محاذٍ على العدسة
الفيديو مقلوب	أضف `--rotation 180` إلى `rpicam-vid` في `start_stream.sh`
تأخير عالي	استخدم `preset=ultrafast;tune=zerolatency` في خيارات libx264

الخيار ب — كاميرا Dahua IP Camera ✅ مستخدمة حالياً

> الكاميرا المستخدمة حالياً هي **Dahua DH-IPC-HFW2431SP-S-S2** — كاميرا شبكية تتصل عبر Ethernet وتُشفّر الفيديو داخلياً. لا يتحمّل الـ Pi أي عبء معالجة للفيديو.

مواصفات الكاميرا

الخاصية	القيمة
الموديل	DH-IPC-HFW2431SP-S-S2
الدقة	4MP (2560×1440)
العدسة	3.6mm ثابتة
الطاقة	12V DC أو POE
الاتصال	Ethernet RJ45
الترميز الداخلي	H.264 / H.265
رؤية ليلية	نعم (IR)

الخطوة الأولى: توصيل الكاميرا

وصّل الكاميرا بمنفذ Ethernet الخاص بالـ Pi (`eth0`) باستخدام كابل RJ45. زوّد الكاميرا بالطاقة عبر 12V أو POE injector.

الخطوة الثانية: ضبط IP ثابت على eth0

الكاميرا تعمل افتراضياً على `192.168.1.108` . نضبط IP ثابتاً على الـ Pi ليكون في نفس الشبكة:

```
sudo nmcli connection add type ethernet ifname eth0 con-name eth0-static \
  ip4 192.168.1.100/24 ipv4.method manual ipv6.method ignore
sudo nmcli connection up eth0-static
```

تحقق من الاتصال بالكاميرا:

```
ping -c 3 192.168.1.108
```

الخطوة الثالثة: الإعداد الأولي للكاميرا

افتح نفقاً SSH للوصول إلى واجهة الكاميرا من جهازك:

```
ssh -L 8080:192.168.1.108:80 lab2@lab2.local -N
```

اترك هذا الأمر يعمل في الخلفية، ثم افتح المتصفح على: `http://localhost:8080`

أكمل الإعداد الأولي — اختر المنطقة، المعيار الفيديوي (PAL للشرق الأوسط)، واضبط كلمة مرور قوية.

> إعداد المنطقة يؤثر فقط على التوقيت واللغة الافتراضية — لا يقيّد استخدام الكاميرا في أي مكان في العالم.

الخطوة الرابعة: التحقق من بث RTSP

```
ffprobe -v error -rtsp_transport tcp \
  'rtsp://admin:YOUR_PASSWORD@192.168.1.108/cam/realmonitor?channel=1&subtype=0'
```

> إذا كانت كلمة المرور تحتوي على `@` يجب استبدالها بـ `%40` في الرابط.

الجزء الرابع — بث الفيديو إلى QGroundControl

الكاميرا تُشفّر الفيديو داخلياً وتبثّه عبر RTSP. الـ Pi يُعيد تغليف الحزم فقط ويُرسلها إلى QGC عبر UDP — بدون أي معالجة أو تشفير.

```
كاميرا Dahua (H.264) → RTSP → Pi → RTP/UDP → QGroundControl
```

الخطوة الأولى: استنساخ المستودع على الـ Pi

```
git clone https://github.com/salemaljebaly/rpi-setup-docs.git
cd rpi-setup-docs
```

الخطوة الثانية: تشغيل البث يدوياً (للاختبار)

```
bash scripts/start_ipcam_stream.sh
```

الخطوة الثالثة: إعداد QGroundControl

1. افتح QGroundControl (https://qgroundcontrol.com/) على جهازك
2. انقر على **أيقونة Video** ← **Application Settings** ← **Q**
3. اضبط **Video Source** على `UDP h.264 Video Stream`
4. اضبط **UDP Port** على `5600`

5. أغلق الإعدادات — يجب أن يظهر الفيديو فوراً في واجهة HUD

الخطوة الرابعة: تفعيل البدء التلقائي عند الإقلاع

بعد تحديث كلمة المرور وعنوان IP في `scripts/start_ipcam_stream.sh` ، شغّل سكريبت التثبيت:

```
bash scripts/install_ipcam_service.sh
```

هذا السكريبت يقوم تلقائياً بـ: نسخ ملف الخدمة، تفعيله، وتشغيله. بعد كل إقلاع لـ Pi ستعمل الكاميرا تلقائياً.

للتحقق من حالة الخدمة أو إعادة تشغيلها:

```
sudo systemctl status ipcam-stream
sudo systemctl restart ipcam-stream
```

الجزء الخامس — إعداد بث الكاميرا (المتغيرات)

عدّل ملف `scripts/start_ipcam_stream.sh` لتحديث عنوان الكاميرا، بيانات الدخول، أو عنوان الجهاز المستهدف:

```
CAMERA_IP=192.168.1.108
CAMERA_USER=admin
CAMERA_PASS='YOUR_PASSWORD'     # استخدم 40% بدلاً من @ في كلمة المرور
TARGET_IP=192.168.0.125         # عنوان IP للجهاز الذي يشغّل QGroundControl
PORT=5600
```

> إذا كانت كلمة المرور تحتوي على `@` يجب استبدال كل `@` بـ `%40` في الرابط. مثال: `my@@password` → `my%40%40password`

الجزء السادس — MAVLink Router للتحكم بالطائرة

لتوصيل وحدة التحكم الطيراني (مثل Cube Orange+) بـ QGroundControl عبر Raspberry Pi، نستخدم برنامج **mavlink-router** (https://github.com/mavlink-router/mavlink-router).

الخطوة الأولى: تثبيت التبعيات

```
sudo apt install -y git meson ninja-build pkg-config gcc g++ \
  libsystemd-dev python3-pip
```

الخطوة الثانية: بناء وتثبيت mavlink-router

```
git clone https://github.com/mavlink-router/mavlink-router.git
cd mavlink-router
git submodule update --init --recursive
meson setup build -Dsystemdsystemunitdir=/lib/systemd/system
ninja -C build
sudo ninja -C build install
```

الخطوة الثالثة: إنشاء ملف الإعداد

```
sudo mkdir -p /etc/mavlink-router
sudo nano /etc/mavlink-router/main.conf
```

محتوى الملف:

```
[General]
Log=/var/log/mavlink-router
MavlinkDialect=ardupilotmega

[UartEndpoint cube]
Device=/dev/ttyACM0
Baud=115200
FlowControl=false

[UdpEndpoint qgc]
Mode=Normal
Address=192.168.0.125
Port=14550
```

> استبدل `192.168.0.125` بعنوان IP جهازك. المنفذ `14550` هو المنفذ الافتراضي لـ QGroundControl للاستماع لاتصالات MAVLink.

الخطوة الرابعة: تشغيل الخدمة

```
sudo systemctl enable mavlink-router
sudo systemctl start mavlink-router
sudo systemctl status mavlink-router
```

الخطوة الخامسة: إعداد QGroundControl

1. افتح QGroundControl
2. انتقل إلى **Application Settings** ← **Comm Links**
3. أضف اتصالاً جديداً من نوع **UDP** على المنفذ `14550`
4. انقر **Connect** — يجب أن يظهر الجهاز الطائر

الخطوة السادسة: إعادة التشغيل التلقائي عند إعادة توصيل USB

إذا لم يكن Cube Orange+ متصلاً عند إقلاع الـ Pi، ستفشل الخدمة وتتوقف عن المحاولة. لإصلاح ذلك، أنشئ ملف تجاوز لـ systemd:

```
sudo mkdir -p /etc/systemd/system/mavlink-router.service.d
sudo nano /etc/systemd/system/mavlink-router.service.d/override.conf
```

الصق المحتوى التالي:

```
[Service]
Restart=always
RestartSec=10
StartLimitIntervalSec=0
```

ثم أعد تحميل الإعدادات:

```
sudo systemctl daemon-reload
sudo systemctl restart mavlink-router
```

> هذا يضمن أن الخدمة تعيد المحاولة كل 10 ثوانٍ إذا لم يكن Cube متصلاً عند الإقلاع، أو إذا تم فصله وإعادة توصيله.

الجزء السابع — البث عن بُعد عبر Tailscale

للوصول عن بُعد عبر الإنترنت (عندما يكون Pi وجهازك على شبكات مختلفة)، استخدم **Tailscale VPN** (https://tailscale.com/).

الخطوة الأولى: تثبيت Tailscale على Raspberry Pi

```
curl -fsSL https://tailscale.com/install.sh | sh
sudo tailscale up
```

الخطوة الثانية: تثبيت Tailscale على جهازك

حمّل التطبيق من tailscale.com/download (https://tailscale.com/download) وسجّل دخولك بنفس الحساب المستخدم في الـ Pi.

الخطوة الثالثة: معرفة Tailscale IP لجهازك

نفّذ هذا الأمر على جهازك:

```
tailscale ip -4
```

ستحصل على عنوان يبدأ بـ `100.x.x.x` — هذا هو العنوان الذي ستستخدمه.

الخطوة الرابعة: التحقق من اتصال كلا الجهازين

على الـ Pi، تحقق من أن جهازك يظهر متصلاً:

```
tailscale status
```

> يجب أن يظهر جهازك بعنوان `100.x.x.x` بدون كلمة "offline". إذا ظهر offline افتح Tailscale على جهازك وتأكد من الاتصال بنفس الحساب.

الخطوة الخامسة: تحديث ملفات الإعداد على الـ Pi

يجب تغيير العنوان في موضعين فقط:

1. **بث الكاميرا** — في `scripts/start_ipcam_stream.sh` :

```
TARGET_IP=100.x.x.x   # Tailscale IP لجهازك
```

2. **MAVLink Router** — في `/etc/mavlink-router/main.conf` :

```
[UdpEndpoint qgc]
Mode=Normal
Address=100.x.x.x     # Tailscale IP لجهازك
Port=14550
```

ثم أعد تشغيل خدمة البث:

```
sudo systemctl restart ipcam-stream
```

> كل شيء آخر يبقى كما هو — نفس المنافذ، نفس إعدادات Tailscale. QGroundControl يتولى التوصيل بشفافية عبر الإنترنت بغض النظر عن موقع الجهازين.

> إذا كان البث متقطعاً على إنترنت بطيء، غيّر `subtype=0` إلى `subtype=1` في رابط RTSP داخل `start_ipcam_stream.sh` للحصول على دقة أقل وبيانات أخف.

الجزء الثامن — التحكم بالمحركات عبر TX12 و Sabertooth

هذا الجزء يشرح مسار التحكم لاستخدام **Radiomaster TX12** لتشغيل **محركَين 12V** عبر **Cube Orange+** و **Sabertooth HS Hybrid**.

بنية النظام

لقيادة المحركات مباشرة بالأذرع (joystick)، مسار الإشارة هو:

```
TX12 transmitter → RC receiver → Cube Orange+ RC input → Cube PWM outputs → Sabertooth S1 / S2 → Motor
s
```

> الـ Raspberry Pi و `mavlink-router` **ليسا** المسار الرئيسي لحركة أذرع TX12. دورهما هو: القراءة عن بُعد (telemetry)، الوصول عبر QGroundControl، أتمتة عبر Python أو MAVLink، ومراقبة المخرجات والمعاملات.

الافتراضات

- Sabertooth في وضع **RC/PWM input**
- `S1` = مدخل القناة الأولى للمحرك
- `S2` = مدخل القناة الثانية للمحرك
- طرف `0V` في Sabertooth متصل بأرضي إشارة Cube
- فقط **الإشارة + الأرضي** متصلان من Cube إلى Sabertooth
- طرف `5V` في Sabertooth غير مستخدم

توصيل الأسلاك

```
Cube output GND   → Sabertooth 0V
Cube output SIG1  → Sabertooth S1
Cube output SIG2  → Sabertooth S2
```

> سلك 5V الأحمر (servo) عادةً **غير مطلوب**. Sabertooth لديه طاقته الخاصة من مصدر 12V. قراءة multimeter على إشارة PWM ستُظهر جهداً متوسطاً صغيراً مثل `0.17V` إلى `0.8V` — هذا طبيعي ولا يعني ضعف الإشارة.

متطلبات ما قبل الاختبار

1. `mavlink-router` يعمل على الـ Pi و QGroundControl يرى Cube
2. مستقبل TX12 متصل بـ Cube ومدخل RC مرئي في QGroundControl
3. معايرة الراديو في QGroundControl مكتملة
4. إعداد أمان Cube مُعدّل لاختبار الطاولة (`BRD_SAFETYOPTION = 0`)
5. المركبة يمكنها التسليح بنجاح بدون خطأ `non-zero throttle`

نموذج التحكم الموصى

لمركبة أرضية بمحركَين:

- **أعلى/أسفل** = دفع (throttle)
- **يسار/يمين** = توجيه (steering)

يقوم Cube بخلط المدخلات إلى مخرجَين: أحدهما يُشغّل `S1` والآخر `S2` .

إجراء اختبار الطاولة

افصل الحمل عن نظام الدفع قبل الاختبار إن أمكن.

1. **تحقق من مدخل RC:** افتح **Radio** → **Vehicle Setup** → **QGroundControl**. حرّك الأذرع وتأكد أن القيم تتغير بشكل صحيح.
2. **تحقق من وصول المخرج إلى Sabertooth:** عند تحريك الأذرع يجب أن يتغير PWM للمخرج.
3. **سلّح المركبة:** Cube يمنع المخرجات حتى يتم التسليح. إذا كانت المركبة غير مسلّحة فالمخرجات قد تبقى ثابتة.
4. **اختبر اتجاهاً واحداً في كل مرة:**
   - أذرع محايدة: المحركان متوقفان
   - أمام: المحركان يدوران للأمام
   - خلف: المحركان يدوران للخلف

- يمين: سرعات المحركَين تنقسم بشكل صحيح
- يسار: سرعات المحركَين تنقسم بشكل صحيح

إذا تحرّك TX12 في QGC لكن المحركات لا تتحرك

1. المركبة لا تزال غير مسلّحة
2. معايرة RC غير مكتملة أو حد أدنى للدفع خاطئ
3. Sabertooth ليس في وضع RC/PWM المتوقع
4. `S1` و `S2` غير متصلين بمخرجات Cube النشطة الفعلية
5. الأرضي بين Cube و Sabertooth مفقود
6. جانب طاقة المحرك في Sabertooth يعمل لكن جانب المخرج غير مُفعّل

طريقتان مدعومتان لتشغيل المحركات

الطريقة	المسار	الاستخدام
قيادة RC مباشرة	`TX12 → receiver → Cube → Sabertooth → motors`	عندما يمسك المشغّل بـ TX12
قيادة برمجية / QGC	`Python` أو `QGC → MAVLink → Cube → Sabertooth → motors`	تحكم آلي أو مُبرمَج

> **التوصية:** استخدم TX12 للحركة اليدوية، والـ Raspberry Pi + MAVLink للمراقبة والإعداد والأتمتة لاحقاً. هذا يُبقي مسار التحكم بسيطاً ويتجنب إدخال الـ Pi في حلقة RC المباشرة.

الجزء التاسع — تحديد قنوات RC على Cube

عندما يقرأ QGroundControl إشارة TX12 عبر USB، قد لا تتطابق أسماء القنوات على الشاشة مع قيم `RC_CHANNELS` الفعلية التي تصل إلى Cube. المنظور الصحيح هو **ما يراه Cube عبر MAVLink**.

سكريبت مراقبة القنوات

يتضمن هذا المستودع سكريبت مساعد: `scripts/monitor_rc_channels.py`

نسخ السكريبت إلى الـ Pi:

```
scp scripts/monitor_rc_channels.py lab2@192.168.0.127:/home/lab2/
```

تشغيله على الـ Pi:

```
ssh lab2@192.168.0.127
python3 /home/lab2/monitor_rc_channels.py --channels 4
```

مثال على المخرجات:

```
Connecting to tcp:127.0.0.1:5760...
Watching RC_CHANNELS. Move one stick at a time. Press Ctrl+C to stop.
(1467, 1433, 1100, 1496)
```

كيف تقرأ النتائج

حرّك محور ذراع واحد في كل مرة وراقب أي قناة تتغير:

القناة التي تغيّرت	قيمة SERVOx_FUNCTION
RC1	51
RC2	52
RC3	53
RC4	54

مثال: إذا تغيّرت `RC3` عند تحريك الذراع الأيمن للأعلى/الأسفل، وأردت أن `MAIN OUT 5` يتبعها، اضبط `SERVO5_FUNCTION = 53`.

لماذا قد تبقى RC5 إلى RC8 عند الصفر

في مسار USB الحالي:

```
TX12 → USB → QGroundControl → MAVLink → Cube
```

من الطبيعي أن تصل فقط أول 4 قنوات إلى Cube. القنوات RC5 إلى RC8 قد تبقى `0`.

إذا احتجت RC5 إلى RC8: الطريقة الصحيحة هي إضافة مستقبل راديو حقيقي متصل مباشرة بـ Cube: `TX12 → receiver → Cube RC input` . هذا يوفر قنوات إضافية، مفاتيح، وسلوك failsafe أفضل.

استكشاف الأخطاء وإصلاحها

المشكلة	السبب	الحل
لا يمكن العثور على Pi في الشبكة	لم يكتمل الإقلاع أو بيانات WiFi خاطئة	انتظر دقيقتين، راجع إعدادات WiFi في RPi Imager
رفض SSH (Permission denied)	اسم مستخدم خاطئ أو مفتاح غير مضاف	راجع اسم المستخدم، أعد إضافة المفتاح العام
الكاميرا لا تستجيب لـ ping	eth0 بدون IP أو الكاميرا بدون طاقة	نفّذ `nmcli connection up eth0-static` وتحقق من الطاقة
واجهة الكاميرا لا تفتح في المتصفح	الجهاز ليس في نفس شبكة الكاميرا	افتح نفق SSH: `ssh -L 8080:192.168.1.108:80 lab2@lab2.local -N`
"Waiting for video" في QGC	خدمة البث متوقفة أو إعدادات خاطئة	نفّذ `sudo systemctl restart ipcam-stream`
البث لا يعمل تلقائياً عند الإقلاع	الخدمة غير مفعّلة	نفّذ `sudo systemctl enable ipcam-stream`
خطأ 401 عند الاتصال بـ RTSP	كلمة مرور خاطئة أو @ غير مشفّرة	استبدل كل `@` في كلمة المرور بـ `%40`
QGC لا يستلم MAVLink	IP خاطئ في إعداد mavlink-router	تحقق من IP في `/etc/mavlink-router/main.conf`
mavlink-router فشل عند الإقلاع	Cube لم يكن متصلاً عند بدء الخدمة	أضف تجاوز systemd بـ `Restart=always` (راجع الجزء السادس، الخطوة السادسة)
TX12 يتحرك في QGC لكن المحركات لا تعمل	المركبة غير مسلّحة أو التوصيلات غير صحيحة	راجع قائمة التحقق في الجزء الثامن
RC5 إلى RC8 تبقى صفر	قيد مسار USB joystick — طبيعي	أضف مستقبل راديو حقيقي متصل مباشرة بـ Cube

المراجع والروابط

المرجع	الرابط
Raspberry Pi Imager — أداة تثبيت نظام التشغيل	https://www.raspberrypi.com/software/
Raspberry Pi Connect — الوصول عن بُعد عبر المتصفح	https://connect.raspberrypi.com
GitHub — دليل إنشاء مفتاح SSH	https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent
QGroundControl — برنامج التحكم الأرضي	https://qgroundcontrol.com/
mavlink-router — توجيه بروتوكول MAVLink	https://github.com/mavlink-router/mavlink-router
Tailscale — شبكة VPN للوصول عن بُعد	https://tailscale.com/download
مستودع هذا الدليل	https://github.com/salemaljebaly/rpi-setup-docs