Review the assignment due date

IoT Venture Pitch

ESE5180: IoT Wireless, Security, & Scaling

Team Name: Helldivers

Team Number: 3

Team Member Name	Email Address
Linhai Deng	linhaid@seas.upenn.edu
Zhongyu wang	zhongyuw@seas.upenn.edu
Zihao Cai	zihaocai@seas.upenn.edu
Yuhe Zhang	yz016@seas.upenn.edu

GitHub Repository URL: https://github.com/ese5180/iot-venture-f25-helldivers.git

Concept Development

horse

3.2.2 Product Function

Our product is a wearable IoT device for horses that integrates GPS tracking, temperature and humidity monitoring, and gait/balance sensing. It helps prevent horses from getting lost, detects early signs of illness by monitoring environmental and physiological conditions, and provides alerts for potential hoof or leg issues. The device enables real-time monitoring for owners, trainers, and veterinarians.

3.2.3 Target Market & Demographics

1. Who will be using your product?

- Horse owners, stable managers, trainers, and veterinarians who need live monitoring of their animals' safety, health, and locomotion.
- Users of livestock monitoring and animal wearable technologies — and those already interested in pet wearables — are part of the adjacent user base.
- Equine-focused sports professionals (e.g., horse racing and equestrian teams) will be core users.

2. Who will be purchasing your product?

- Individual horse owners, equestrian clubs, ranches, and breeding farms.
- Veterinary clinics and equine hospitals.
- Professional stables and racecourse facilities.

- Potential B2B customers, such as agricultural tech companies deploying large-scale livestock monitoring solutions.

3. Where in the world (or space!) would you deploy your product?

- Initial deployment: the United States and Europe, where horse ownership, equestrian culture, and racing industries are already established.
- Expansion markets: Asia (China, Japan, South Korea, India), the Middle East (Saudi Arabia, UAE), and Australia — all regions with strong horse racing and breeding traditions.
- According to Market Intelo, the equine GPS tracker market is projected to grow significantly worldwide, and Yahoo Finance reports strong growth in equine healthcare demand, supporting global adoption potential.

4. How large is the market you're targeting, in US dollars?

According to Livestock Monitoring Market, the global livestock monitoring market is projected at **USD 5.18 billion in 2024**.
Because our product combines GPS, health/environmental sensing, and gait/balance analytics specifically for horses, we conservatively assume that the equine segment accounts for ~5% of that, giving an addressable market of **USD ~259 million**.

Additionally, the global horse riding equipment market is estimated at **USD 1.60 billion in 2023** (Horse Riding Equipment Market).
Although that number largely covers tack, saddles, and protective gear, it demonstrates that there is already significant spending in the equestrian domain.

5. How much of that market do you expect to capture, in US dollars?

Combining these references, we estimate a TAM between USD 200–400 million for a connected equine monitoring product. If we aim to capture 2–3% in early stages, that corresponds to USD 4–12 million potential revenue.

6. What competitors are already in the space?

- Nightwatch Smart Halter – monitors equine vital signs and stress levels, with automated alerts.
- Trackener Life – provides GPS tracking and behavior monitoring.
- Equimetrics – combines physiological monitoring with GPS for equestrian performance.
- Equestic SaddleClip – tracks gait and performance metrics for training.

Differentiator: None of these competitors combine **GPS tracking, environmental monitoring, and gait/balance detection** in a single integrated system, which is our unique advantage.

3.2.4 Stakeholders

Interview Information

- **Farm name:** Shadow Creek Farm – Ridley Creek
- **Location:** Glen Mills, Pennsylvania, USA
- **Phone:** (484) 301-0560
- **Type:** Horse facility (boarding, show training)
- **Google rating:** 4.7 ★ (71 reviews)

Interview Summary

How many horses do you have right now, and what do you mainly use them for — racing, training, or just riding?

They currently have about 65 horses at the farm. Most are used for show training and boarding — other people board their own horses there for care.

How do you usually keep track of where your horses are and check their health — GPS, just watching them, or something else?

They manually count the horses every time. The farm also partners with a nearby clinic to do regular weekly health checks. When asked what daily health factors they pay most attention to, they said the most important is **water intake**, followed by **sleep quality**.

Have you ever had a horse get lost, hurt, or sick and found out too late? Could you tell me about one time?

Never — it has never happened. They emphasized that horses are very valuable animals, so they are very careful.

If there was a device that could track your horse's location, body temperature, humidity, and leg issues in real time, which features would be most useful for you?

They mentioned that in Florida there have been cases of horses being stolen at night and sold illegally for meat. However, they feel such a device may not need to be running 24/7; constant monitoring might not be necessary.

When picking this kind of device, what matters most to you — price, battery life, comfort for the horse, accuracy, or easy phone access?

Easy access through a phone app is the most important, followed by **battery life**. They also noted that in areas with hurricanes, the device should consider extreme weather conditions.

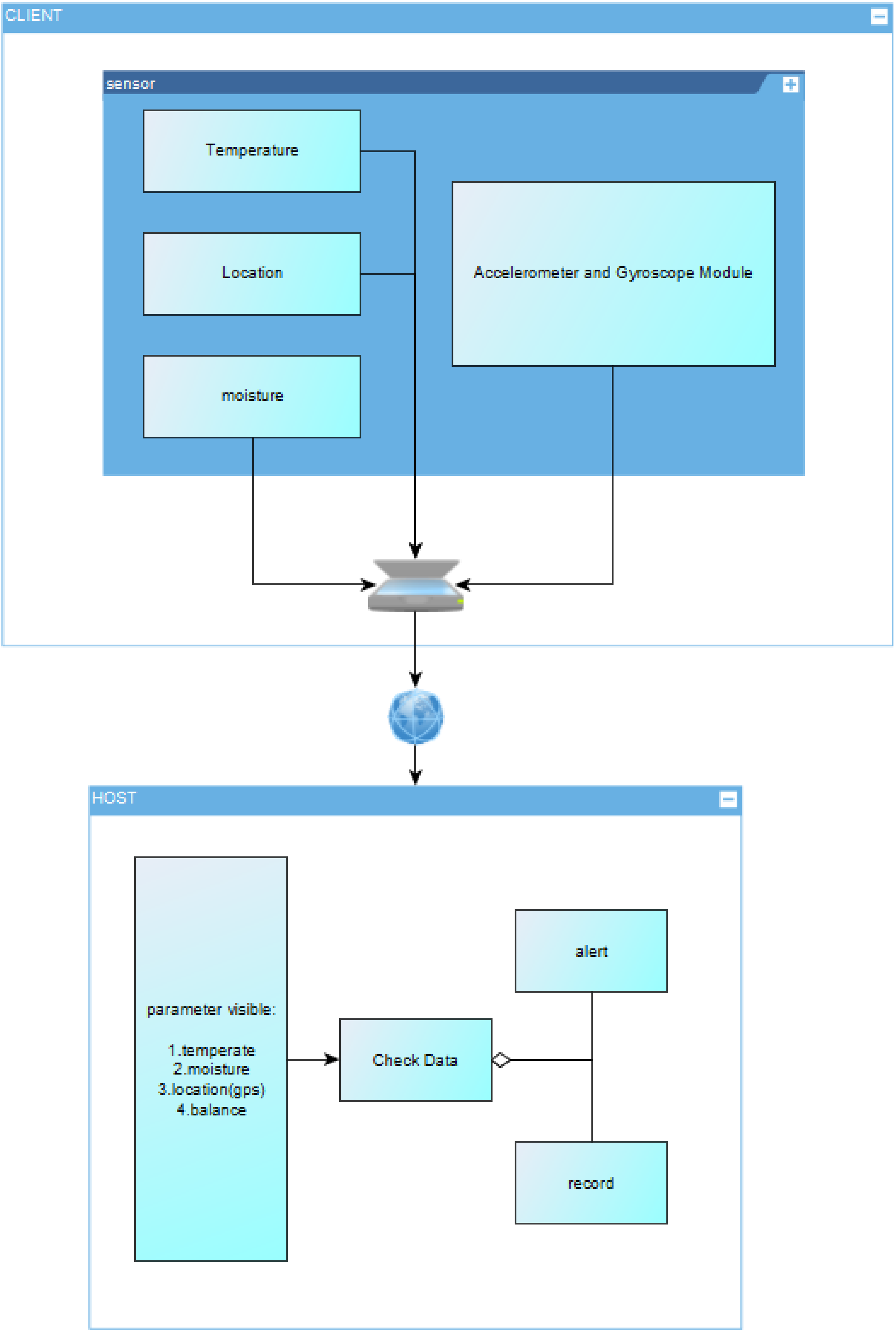
Would you want to share this data with vets or trainers? Do you have any privacy concerns about it?

They think privacy and security are very important in rural areas — people generally don't want to share their personal information or have outsiders know about their property and animals.

How much would you be willing to pay for something like this? Would you rather pay once or pay a subscription?

It depends on the pricing. They prefer a **monthly subscription at first** to try it out. If the device proves useful, they would like the option to switch to a **lifetime purchase with a discount**. Paying a few hundred dollars upfront for something uncertain would feel too risky.

3.2.5 System-Level Diagrams



3.2.6 Security Requirements Specification

Overview

The system will handle sensitive data about horse location, health, and owner property. Security is critical to ensure data integrity, confidentiality, and protection against tampering.

Functionality

SEC 01 – All wireless communication (BLE/Wi-Fi) shall be encrypted using industry-standard protocols (e.g., AES-128 or higher).

SEC 02 – User and animal identifiable data shall not be transmitted or stored in plain text.

SEC 03 – The system shall authenticate devices before accepting data, ignoring messages from unregistered or tampered nodes.

SEC 04 – All sensitive information (keys, credentials) shall be stored in a secure hardware region on the Nordic microcontroller.

3.2.7 Hardware Requirements Specification

Overview

The device must be lightweight, durable, and capable of continuous operation in outdoor equestrian environments. Hardware components will support real-time sensing, wireless communication, and energy-efficient power management.

Functionality

HRS 01 – The system shall be based on a Nordic microcontroller (e.g., nRF52840) to ensure Zephyr RTOS compatibility.

HRS 02 – The hardware shall include GPS(for water intake detection as well), IMU (accelerometer + gyroscope), temperature/humidity sensor, and optional ToF module for leg movement monitoring.

HRS 03 – The device shall be powered by a rechargeable Li-Po battery with >24 hours continuous operation, with optional solar trickle charging for extended use.

HRS 04 – The enclosure shall be weather-resistant (IP65 or higher), lightweight (<300 g), and safe for equine wear without causing discomfort.

HRS 05 – The hardware should include a IMU that detects the balance of horses, monitoring if they are in sleep.

3.2.8 Software Requirements Specification

Overview

The software will collect sensor data (temperature, moisture, GPS, and leg movement from IMU/ToF modules) from wireless nodes attached to the horse, transmit it to the central host, and provide real-time monitoring, alerts, and data logging for horse health and movement analysis.

Users

Primary users are horse farm managers, veterinarians, and researchers who need to monitor horse leg movement and health parameters in real time. Secondary users include software developers and system maintainers who will manage the system.

Functionality

SRS 01 – Each wireless sensor node (per leg) will measure distance to the ground using a ToF sensor at 25–50 Hz.

SRS 02 – Sensor data (IMU + ToF + temperature + moisture + GPS) will be transmitted wirelessly via BLE/Wi-Fi to the central host.

SRS 03 – The host software will synchronize data from 4 legs and compute relative height differences to determine whether the knees are at the same level.

SRS 04 – The system will generate alerts if any leg shows abnormal vibration amplitude or asymmetry > X mm threshold.

SRS 05 – All data will be logged with timestamps in a database for later analysis.

SRS 06 – The user interface will provide real-time visualization of leg movement and health parameters on a PC dashboard.

3.4 Wireless Demo

The demo is uploaded on Github in the File "wireless_demo".

Lihai Deng can pull the repository from GitHub, build, and flash the code to their device.

Zhongyu Wang can pull the repository from GitHub, build, and flash the code to their device.

Yuhe Zhang can pull the repository from GitHub, build, and flash the code to their device.

Zihao Cai can pull the repository from GitHub, build, and flash the code to their device.

Demo video: Wireless Demo

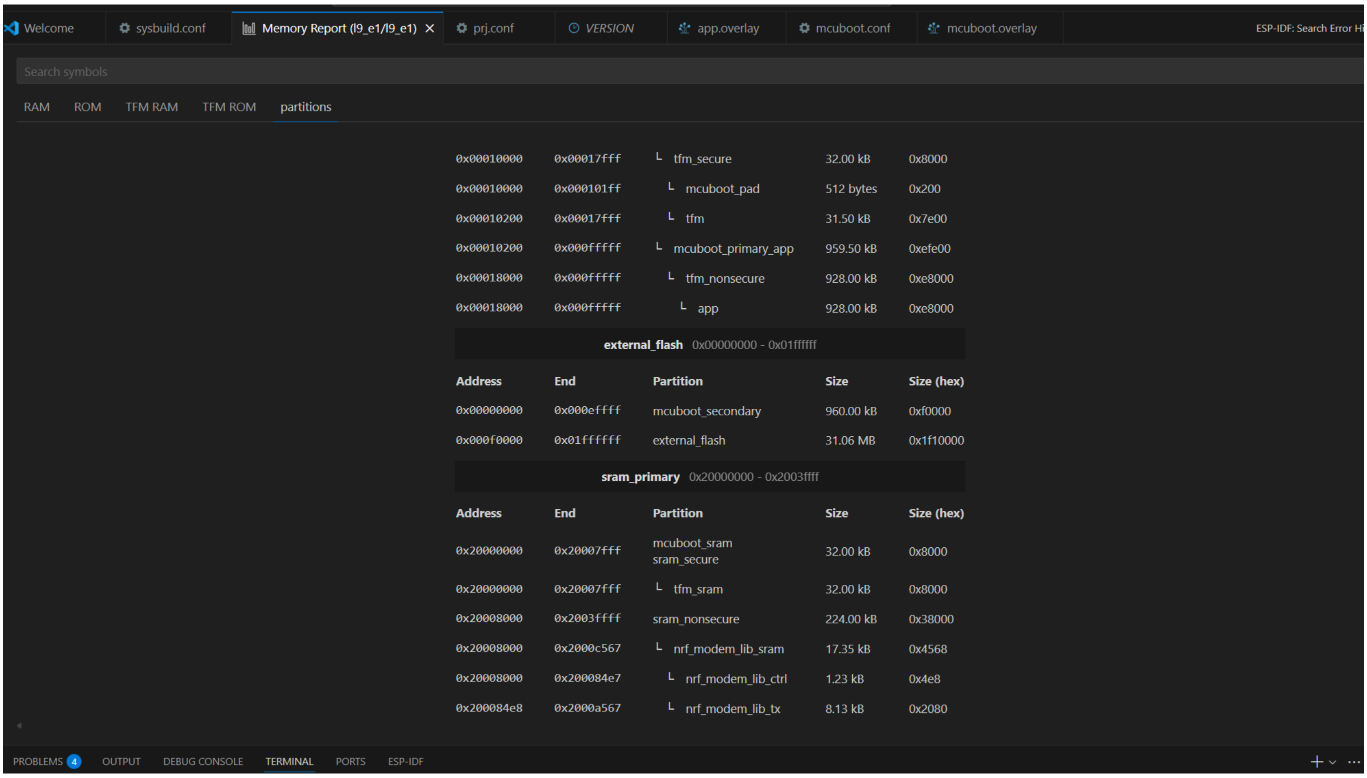
3.5 Secure Firmware Updates

(3.5.1) Bootloading Process Description

1. Bootloader Size

From the memory map:
Partition: mcuboot_primary_app
Address range: 0x00010200 – 0x00017FFF
Size: ~64 KB (0x8000)

The MCUboot bootloader occupies approximately 64 KB of internal flash memory.
It is responsible for validating firmware signatures and managing firmware swapping between slots.



0x00010000	0x00017fff	└ tfm_secure	32.00 kB	0x8000
0x00010000	0x000101ff	└ mcuboot_pad	512 bytes	0x200
0x00010200	0x00017fff	└ tfm	31.50 kB	0x7e00
0x00010200	0x000fffff	└ mcuboot_primary_app	959.50 kB	0xefe00
0x00018000	0x000fffff	└ tfm_nonsecure	928.00 kB	0xe8000
0x00018000	0x000fffff	└ app	928.00 kB	0xe8000

external_flash 0x00000000 - 0x01ffffff

Address	End	Partition	Size	Size (hex)
0x00000000	0x000effff	mcuboot_secondary	960.00 kB	0xf0000
0x000f0000	0x01ffffff	external_flash	31.06 MB	0x1f10000

sram_primary 0x20000000 - 0x2003ffff

Address	End	Partition	Size	Size (hex)
0x20000000	0x20007fff	mcuboot_sram sram_secure	32.00 kB	0x8000
0x20000000	0x20007fff	└ tfm_sram	32.00 kB	0x8000
0x20008000	0x2003ffff	sram_nonsecure	224.00 kB	0x38000
0x20008000	0x2000c567	└ nrf_modem_lib_sram	17.35 kB	0x4568
0x20008000	0x200084e7	└ nrf_modem_lib_ctrl	1.23 kB	0x4e8
0x200084e8	0x2000a567	└ nrf_modem_lib_tx	8.13 kB	0x2080

2. Application Code Size

From the same memory report:
Partition: app
Address range: 0x00018000 – 0x000FFFFF
Size: 928 KB

The main application image occupies approximately 928 KB of flash memory.
This includes the Zephyr-based user application and the MCUmgr subsystem used for firmware updates.

---

3. Who Handles Firmware Image Download

The firmware image download is handled by the application, not the bootloader.

- MCUboot validates image signatures and performs slot swaps during boot.
- The application uses the MCUmgr subsystem via the SMP (Simple Management Protocol) over UART to receive and write firmware images.

Handled by: Application (MCUmgr over UART)

---

4. Wireless Communication Used for Image Download

In this lab setup, firmware images are transferred through:
UART + SMP v2 protocol (Serial DFU)

Wireless method used: UART (Serial DFU)

---

5. Why This Communication Method Was Chosen

- UART DFU provides a simple and reliable interface for testing MCUboot and MCUmgr integration.
- It does not require configuring LTE or BLE stacks.
- It allows fast debugging and consistent transfers during development.

Reason for choice: simplicity, reliability, and ease of debugging.

---

6. Where the Downloaded Firmware Images Are Stored

The new firmware images are written into the secondary slot before activation.
Partition: mcuboot_secondary
Address range: 0x000E0000 – 0x000EFFFF
Size: 960 KB

Stored in: mcuboot_secondary partition (in external flash)

---

7. Features Enabled to Handle Firmware Update Failures

MCUboot implements several mechanisms to protect against faulty or invalid updates.

Failure Type	Handling Feature
Invalid signature	ECDSA P-256 digital signature verification using the private/public key pair
Faulty but signed code	Two-step "Test → Confirm" update process (new image only confirmed after successful boot)
Corrupted firmware	SHA-256 hash check ensures data integrity
Interrupted transfer	Size and CRC validation before marking the image as ready
Version mismatch	Version comparison prevents downgrades
Power failure during swap	Atomic image swapping and rollback support

Enabled safety features:

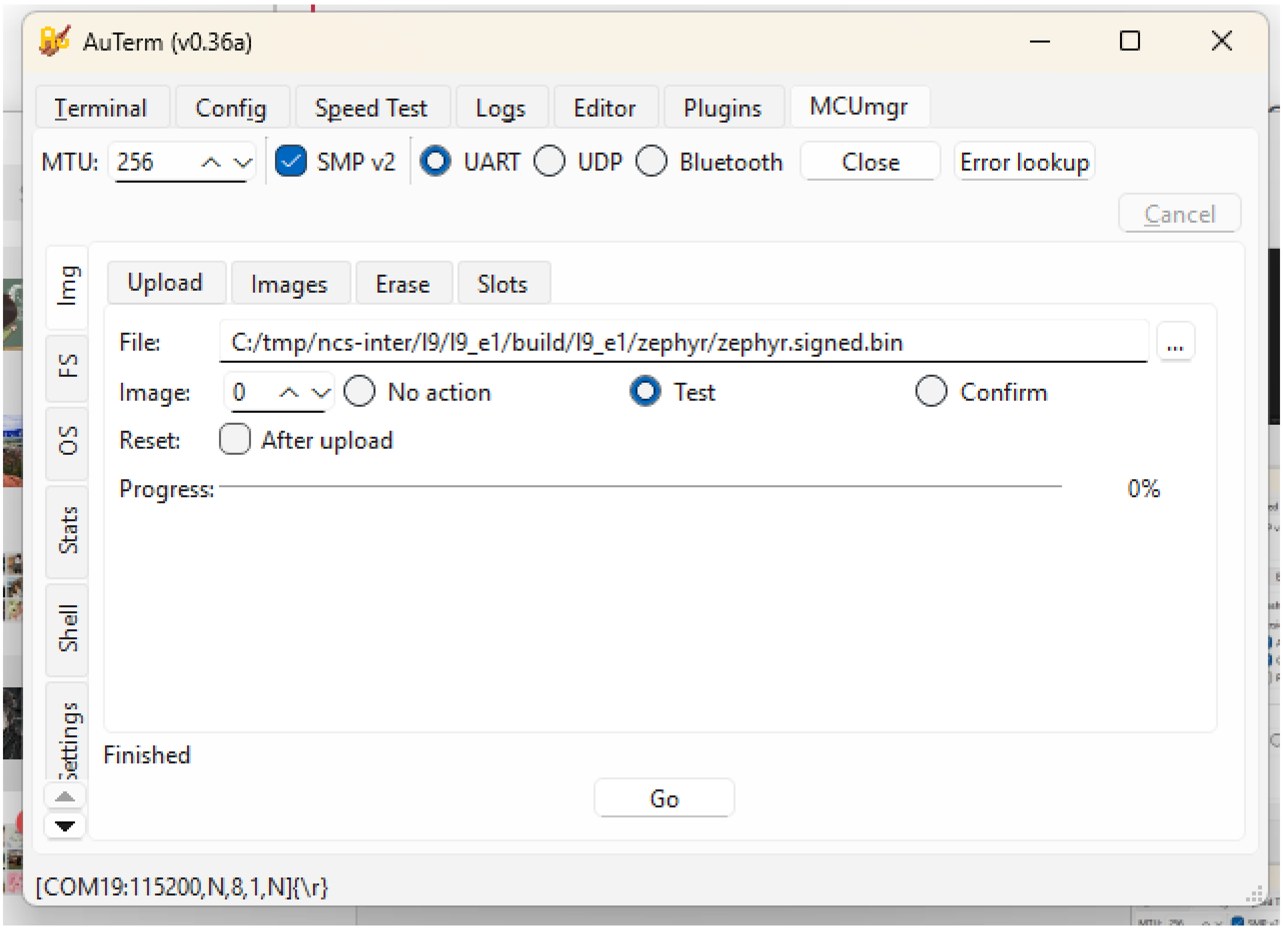
- Digital signature verification (ECDSA P-256)
- SHA-256 integrity checking
- Slot rollback and atomic swap recovery
- Test-before-confirm update process

---

8. Summary

Item	Description
Bootloader size	~64 KB
Application size	~928 KB
Who downloads firmware	Application (MCUmgr subsystem)

Item	Description
Communication method	UART (Serial DFU using SMP v2)
Reason for method	Simple, reliable, and easy to debug
Image storage	mcuboot_secondary partition (external flash)
Failure recovery	Signature verification, hash check, rollback, confirm-test mechanism



MTU: 256 ☑ SMP v2 ● UART ○ UDP ○ Bluetooth

File: C:/tmp/ncs-inter/l9/l9_e1/build/l9_e1/zephyr/zephyr.signed.bin

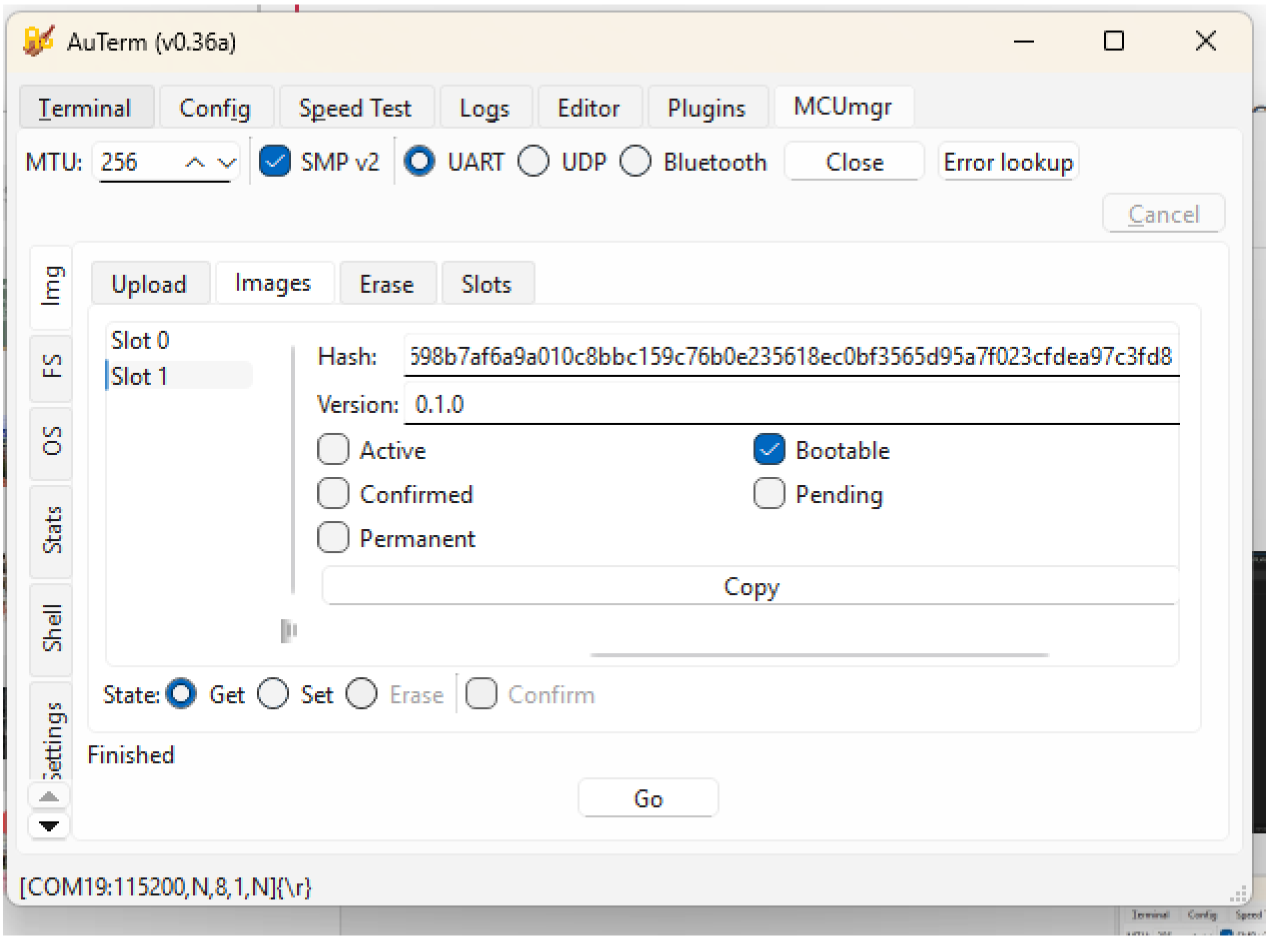
Image: 0 ○ No action ● Test ○ Confirm

Reset: □ After upload

Progress: 0%

Finished

[COM19:115200,N,8,1,N]{\r}



MTU: 256 ☑ SMP v2 ● UART ○ UDP ○ Bluetooth

Hash: 598b7af6a9a010c8bbc159c76b0e235618ec0bf3565d95a7f023cfdea97c3fd8

Version: 0.1.0

□ Active ☑ Bootable

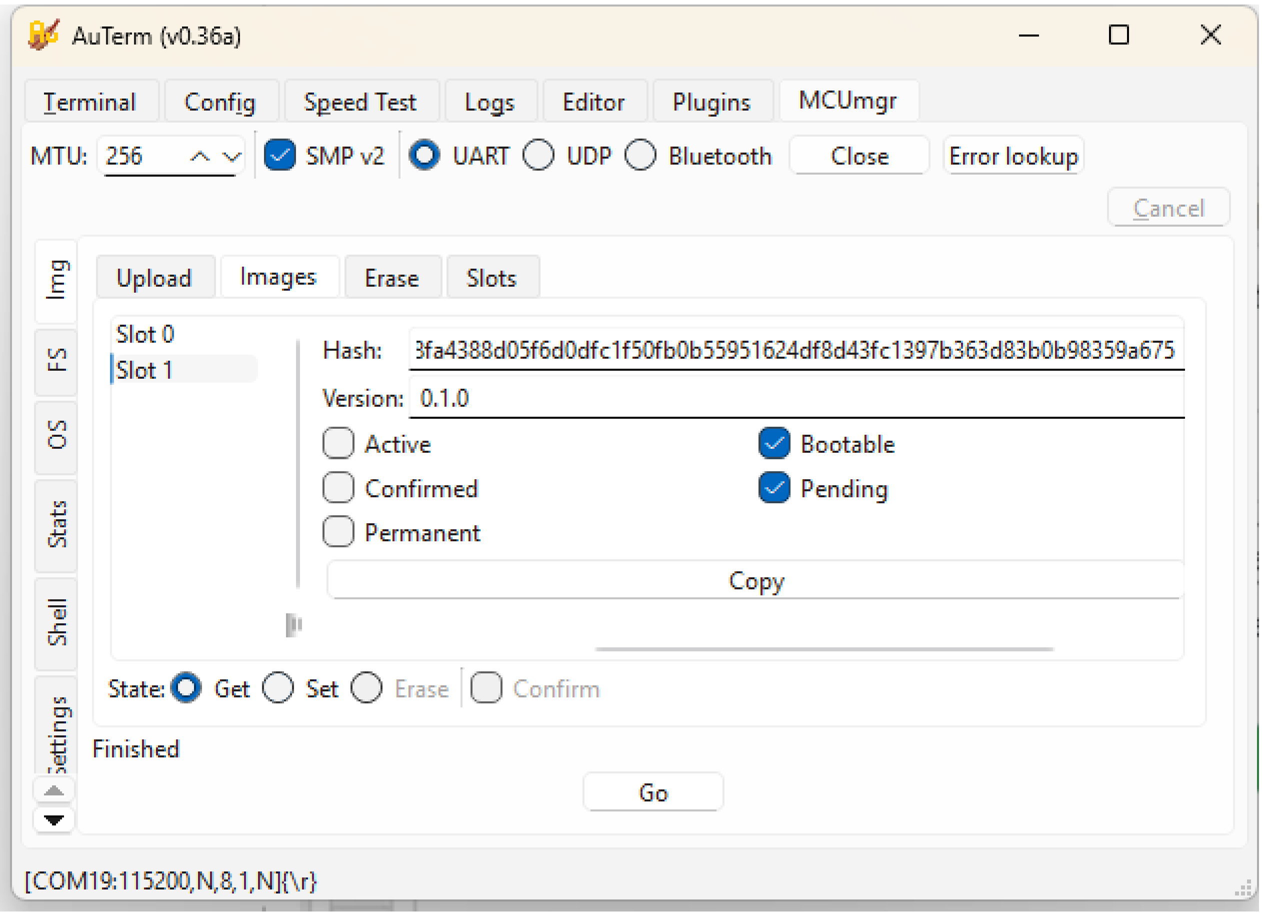
□ Confirmed □ Pending

□ Permanent

State: ● Get ○ Set ○ Erase □ Confirm

Finished

[COM19:115200,N,8,1,N]{\r}



MTU: 256

☑ SMP v2 ● UART ○ UDP ○ Bluetooth

Hash: 3fa4388d05f6d0dfc1f50fb0b55951624df8d43fc1397b363d83b0b98359a675

Version: 0.1.0

□ Active ☑ Bootable

□ Confirmed ☑ Pending

□ Permanent

State: ● Get ○ Set ○ Erase □ Confirm

Finished

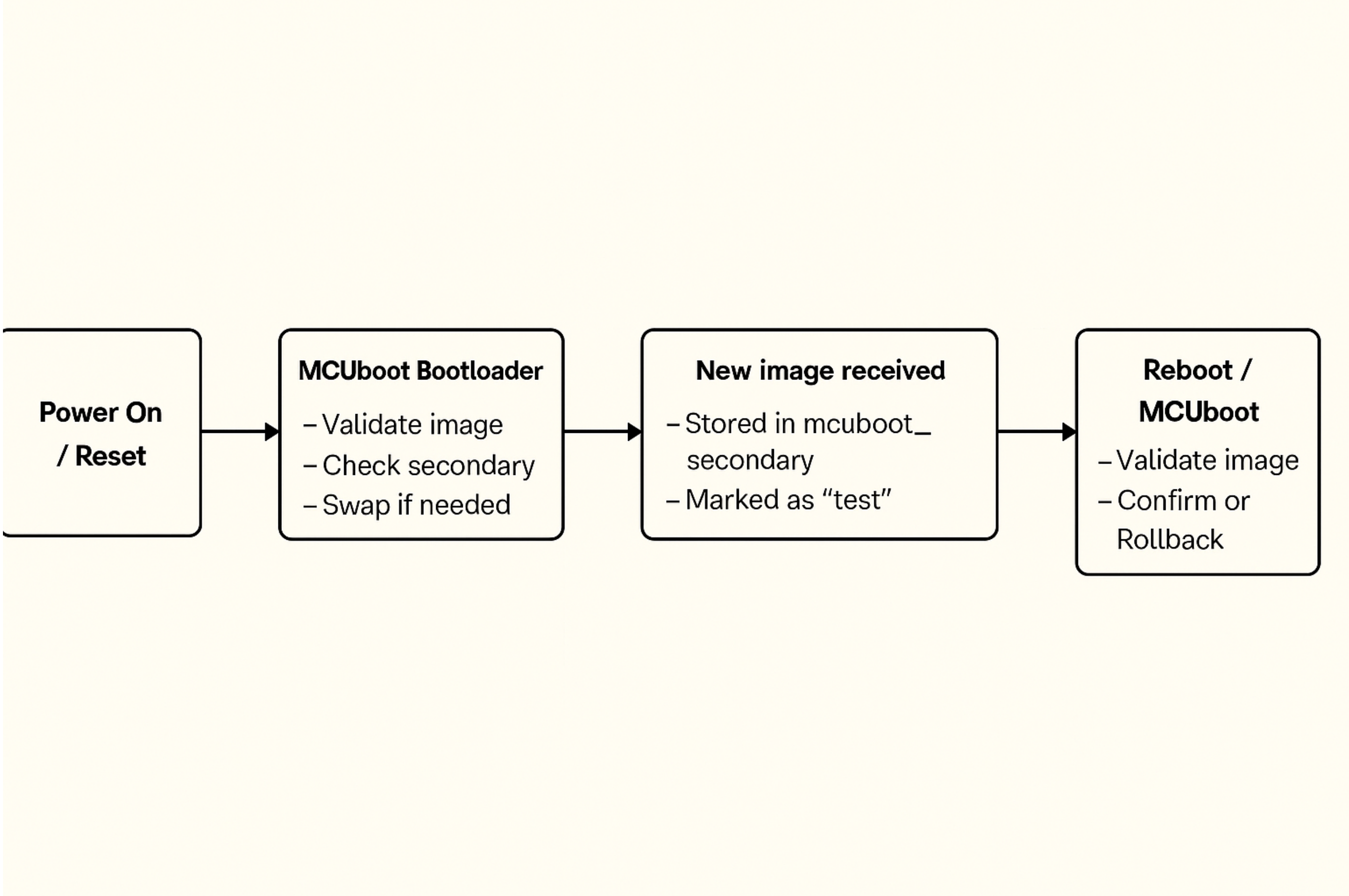
[COM19:115200,N,8,1,N]{\r}

9. Bootloading Process Overview

1. Device powers on and runs MCUboot.
2. MCUboot checks the primary slot for a valid, signed image.
3. If a new image is found in the secondary slot, MCUboot verifies it and swaps it into the primary slot.
4. MCUboot boots the main application.
5. The application runs normally and can use MCUmgr to download new firmware images.
6. When a new image is received, it is written to the secondary slot and marked as "test."
7. On reboot, MCUboot validates and boots the test image.
8. If it runs successfully, the application confirms it; otherwise, MCUboot rolls back to the previous version.

---

10. Boot Flow Diagram



(3.5.2) FOTA

APPROVED BY NICK, since our celluar is not working as expected, ignored for now.

https://devzone.nordicsemi.com/f/nordic-q-a/125395/rf9151-dk-connects-to-nrf-cloud-via-coap-but-mqtt-public-internet-not-accessible-conexa-sim