Power-Efficient Long-Range Drone Networking System for UAV Detection

Final Presentation

Team: BTT

Soonchan Kwon, Gihwan Kim, Nawon Kim, Nahyeong Kim, Karteikay Dhuper, Prakshi Chander

Contents

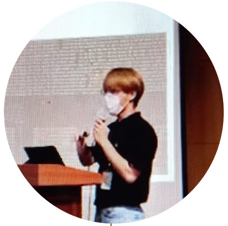
INTRODUCTION

[redacted]

- Team members
- Subject
- Related works

Introduction

Team members



Soonchan Kwon

Team Leader
Networking

Chonnam Nat'l Univ.
Software Engineering

Nawon Kim

Drone Detection

Chonnam Nat'l Univ.
Computer Engineering

Gihwan Kim

Paper
Drone Detection

Chungnam Nat'l Univ.
Computer Science & Engineering

Nahyeong Kim

Networking

Chungnam Nat'l Univ.
Computer Science & Engineering

Karteikay Dhuper

Drone Detection

Purdue Univ.
Computer and Information Technology

Prakshi Chander

Networking

Purdue Univ.
Cybersecurity

Introduction

Subject

Russia launches ‘kamikaze’ drone attack on Kyiv, killing 4 and hitting civilian infrastructure

By Victoria Butenko, Olga Voitovych and Yulia Kesaieva, CNN

Updated 3:09 PM EDT, Mon October 17, 2022

[1]

- UAVs have become more easily accessible to the public, companies, and even terrorists
- This has raised the need of having a Counter Drone System (CDS)

[1] Victoria Butenko, Olga Voitovych and Yulia Kesaieva. "Russia launches ‘kamikaze’ drone attack on Kyiv, killing 4 and hitting civilian infrastructure," CNN: October 17, 2022.

Introduction

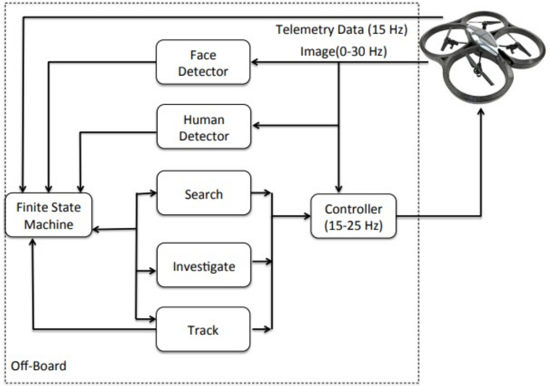
Related Works

Blighter Surveillance Systems, GCDS

[2]

NASA, ACDS

[3]



- Ground Counter Drone Systems (GCDS) have geological restrictions
- Airspace Counter Drone Systems (ACDS) in NASA research uses WiFi mesh network which is fast but consumes high-power

[2] "Anti-UAV Defence System," Verdict Media Limited, London, United Kingdom, 2015. [Online]. Available: https://www.homelandsecurity-technology.com/projects/anti-uav-defence-system-auds/
[3] A. Chakrabarty, R. A. Morris, X. Bouyssounouse, and R. Hunt, "An Integrated System for Autonomous Search and Track with a small Unmanned Aerial Vehicle," AIAA Information Systems-AIAA Infotech @ Aerospace, Jan. 2017, doi: 10.2514/6.2017-0671..

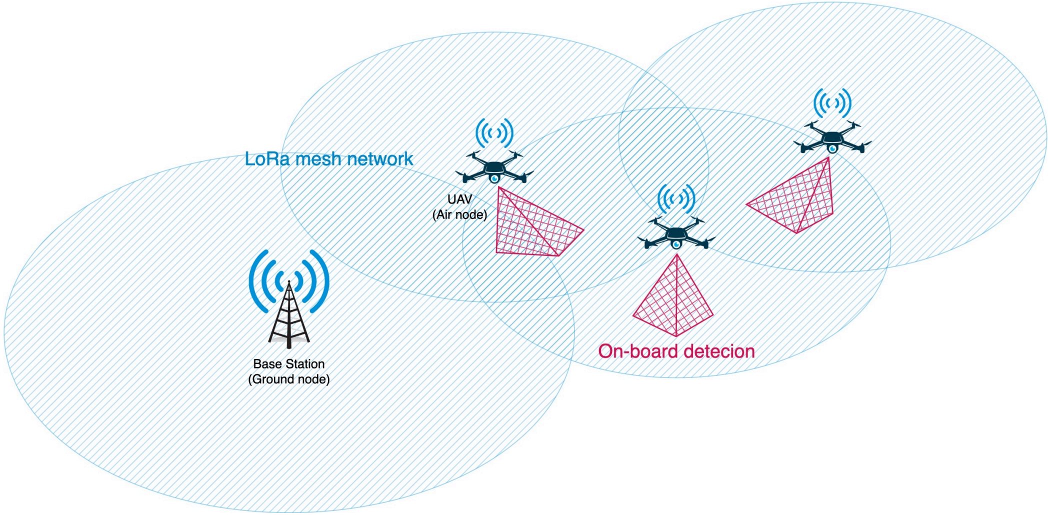
SYSTEM OVERVIEW

[redacted]

- Overview of the system
- Novelty
- Hardware implementation
- Software implementation

System overview

Overview of the system



ACGS using observer drones in LoRa mesh network

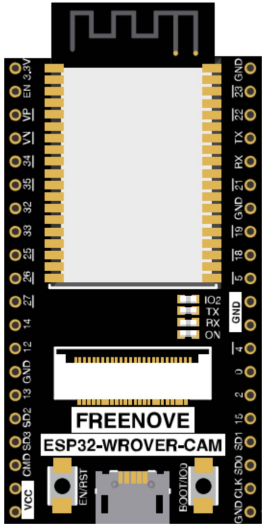
System overview

Novelty

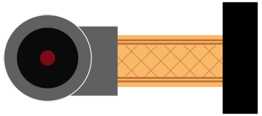
- **Observer drones to detect unidentified aerial object**
  - ➡ Immune to a geological restriction
- **On-board detection**
  - ➡ Low-power and low-cost Machine Learning (ML) detection
  - ➡ Low data-rate and long-range communication technology
- **LoRa Mesh network via multi-hop communication**
  - ➡ Easily buildable, expandable and sustainable network

Hardware implementation

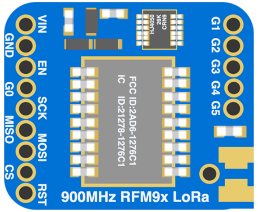
Components



ESP32-WROVER
Microcontroller Unit (MCU)



OV2460
Camera Module

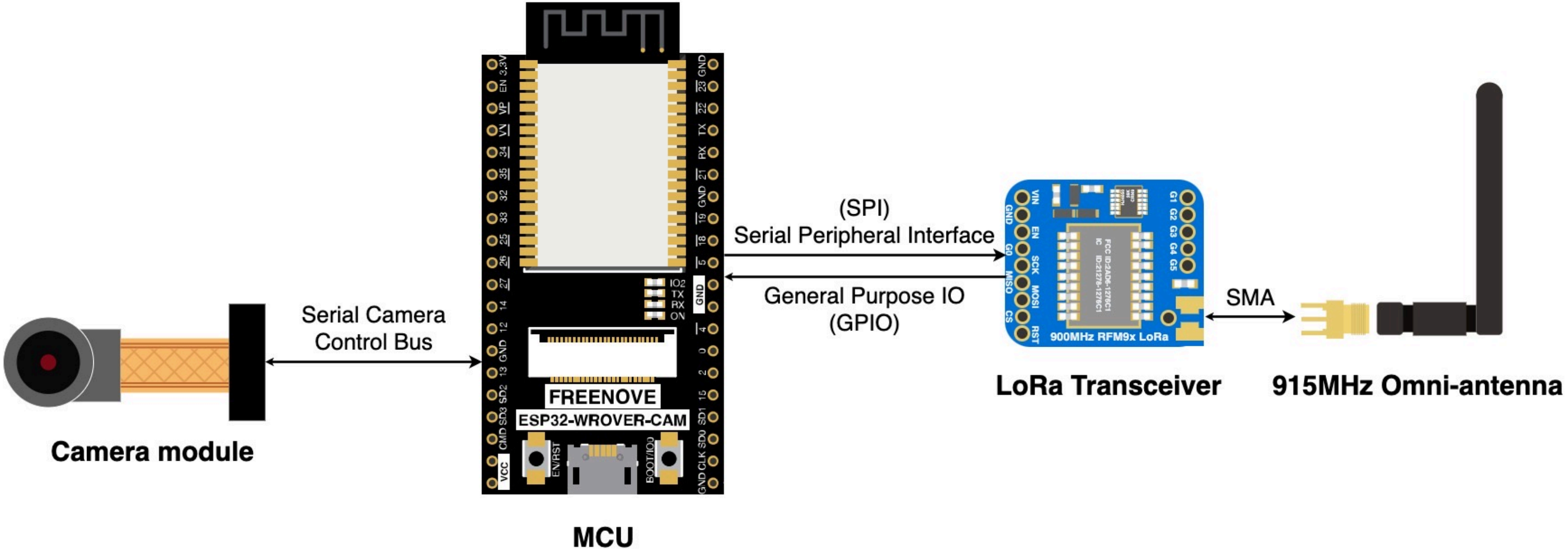


RFM95W
LoRa Transceiver

SMA Adapter & 915MHz Antenna
5dBi Omni-directional antenna

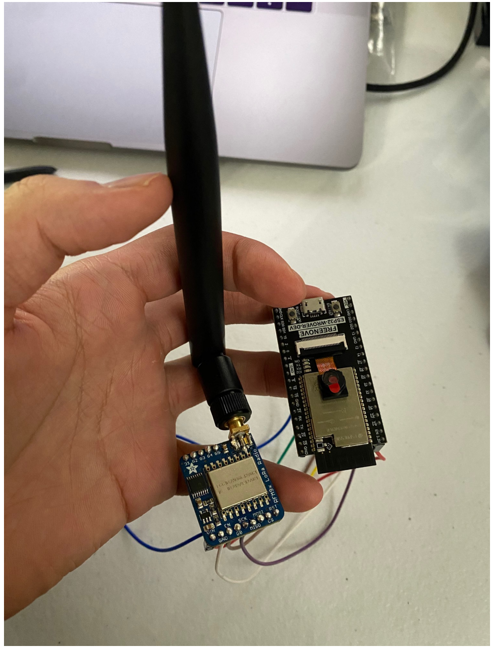
Hardware implementation

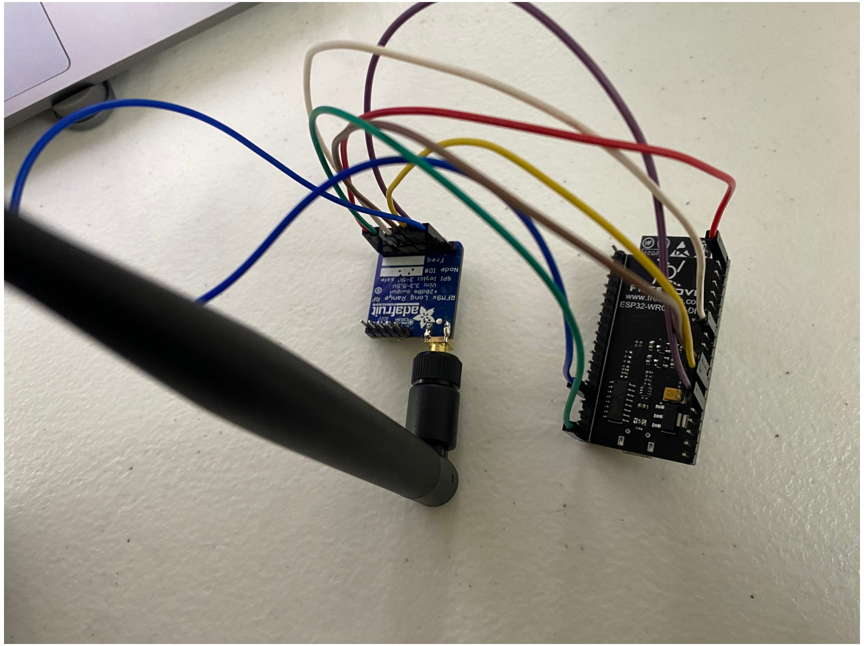
Overall structure of the hardware



Hardware implementation

Hardware implementation





Software implementation

Software platforms

Arduino IDE

- Development platform for MCU
- Compatible with a lot of open-source libraries
- Less time consuming to learn and develop with than ESP-IDF

Software implementation

Software platforms

Arduino-ESP32

- Arduino core for ESP32
- Converts Arduino features for ESP32
- Converts some essential low-level function of ESP32 for Arduino

Software implementation

Software platforms

RadioHead

RadioHead

- Packet radio library based on Arduino
- LoRa transceiver manager class
  - Easy to configure and control a transceiver
- Mesh network manager class
  - Easy to build a mesh network

Software implementation

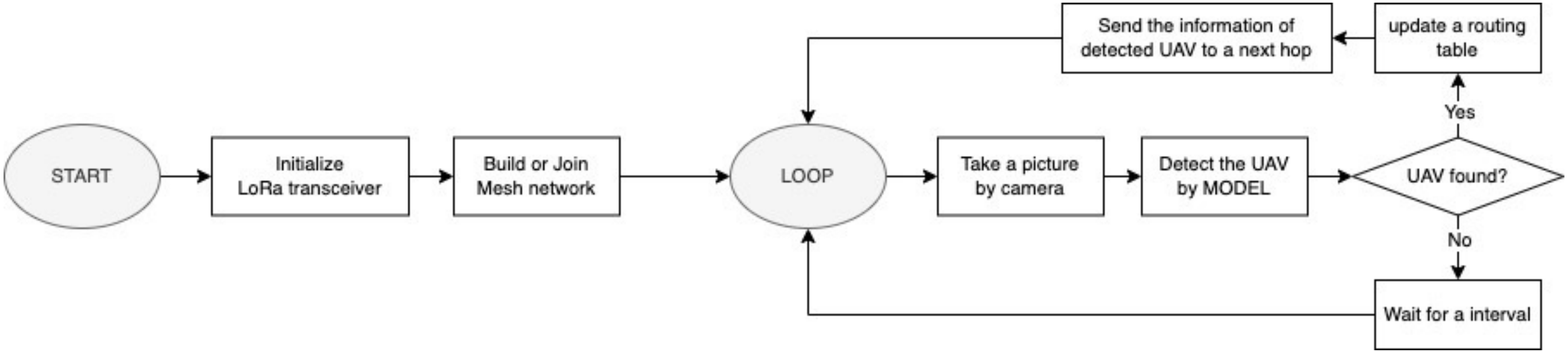
Software platforms

TensorFlow Lite for Microcontrollers

- Open-source framework for running machine learning models on MCU or small memory device
- Supports ESP32 architecture and C++ library

Software implementation

Flowchart of the system



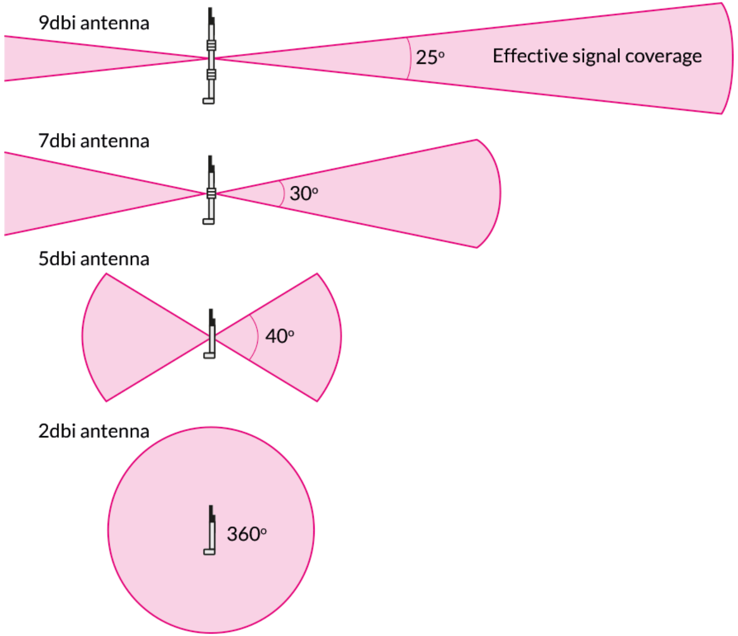
PROBLEMS

CHALLENGES

- Radiation patterns of Omni-directional antenna
- Detection model compression

Problems & Challenges

Radiation patterns of Omni-directional antenna

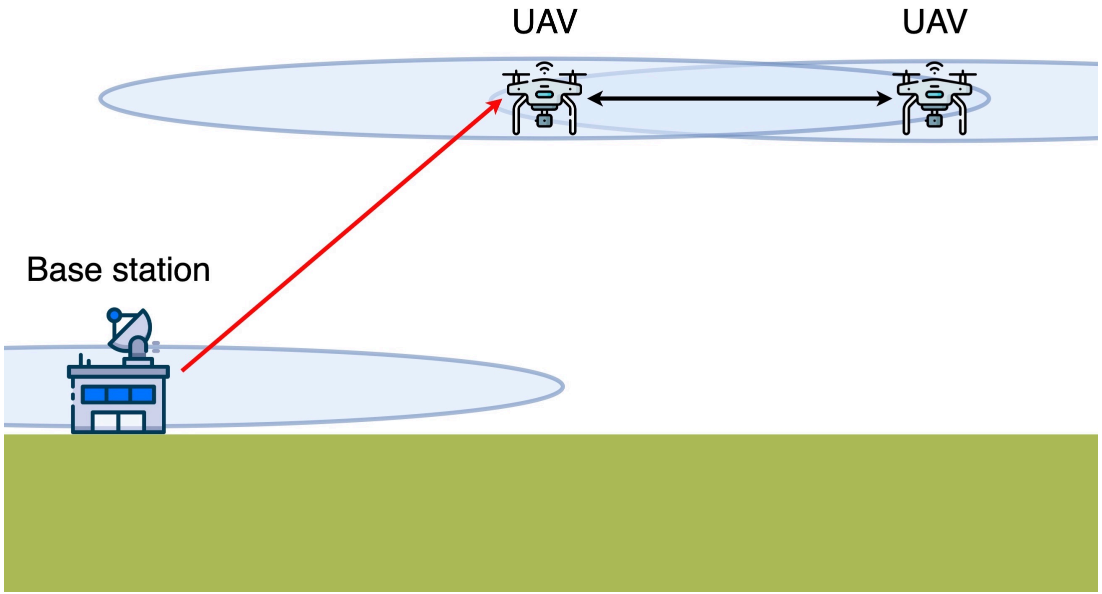


Trade-off

"Angle VS Distance"

Problems & Challenges

UAV-to-Ground (U2G) Communication



The altitude difference of U2G communication

Problems & Challenges

UAV-to-Ground (U2G) Communication

The role of
Base station

The role of
UAV

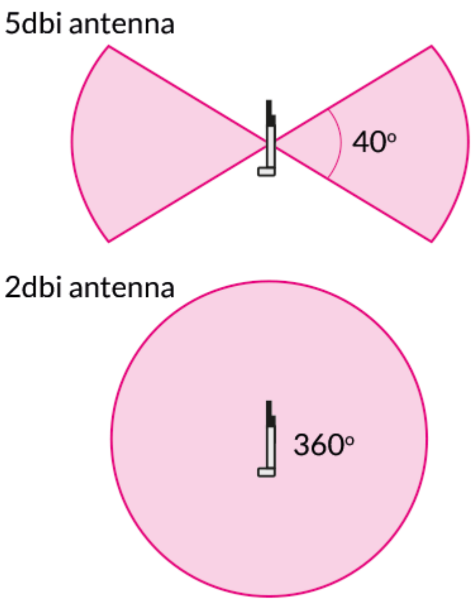
Problems & Challenges

UAV-to-Ground (U2G) Communication

Ground ->	Node	for 2m30s		KSW ->	Parking building	
Height	0m	5m	8m	11m	14m	17m
Distnace	30m	30m	30m	30m	30m	30m
Where	Ground	2nd	3th	4th	5th	roof
AVG	-78.64865	-74.3043	-70.4681	-70.9744	-78.4545	-79.8542
WORSE	-88	-85	-96	-73	-84	-84
BEST	-73	-69	-67	-69	-76	-75

Problems & Challenges

Choosing Omni-directional antenna



Problems & Challenges

Detection model compression

ESP32-WROVER: small memory

- 4 MB Flash memory, 8 MB PSRAM (technically 4 MB)
- 520 KB on-chip SRAM for data and Instruction

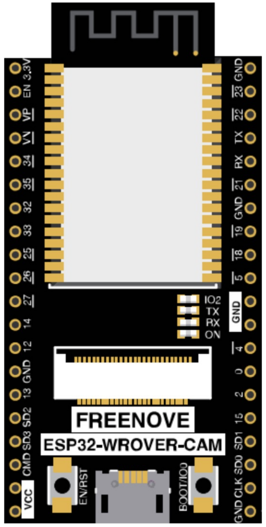
Detection model in TensorFlow Lite is
usually stored as global,
it takes large memory.

Small detection model

: resize input data,

design small model,

quantization



ESP32-WROVER
MCU

Problems & Challenges

Detection model compression

Input data: Visual data

- Supports limited image types

Image resolution	Sharpness	Image resolution	Sharpness
FRAMESIZE_96X96	96x96	FRAMESIZE_HVGA	480x320
FRAMESIZE_QQVGA	160x120	FRAMESIZE_VGA	640x480
FRAMESIZE_QCIF	176x144	FRAMESIZE_SVGA	800x600
FRAMESIZE_HQVGA	240x176	FRAMESIZE_XGA	1024x768
FRAMESIZE_240X240	240x240	FRAMESIZE_HD	1280x720
FRAMESIZE_QVGA	320x240	FRAMESIZE_SXGA	1280x1024
FRAMESIZE_CIF	400x296	FRAMESIZE_UXGA	1600x1200

- Resizes image resolution and channel

(width, height, channel): (98 x 98 x 3) $\xrightarrow{\text{Downsize: 12x}}$ (48 x 48 x 1)

Downsize: 12x

Problems & Challenges

Detection model compression

Simple and small detection model

- structure
  - Convolution layer
  - Batch normalization
  - Average pooling
  - Flatten
  - Dropout
  - Dense

```
Layer (type)                 Output Shape              Param
=================================================================
conv2d_34 (Conv2D)           (None, 48, 48, 2)         52
_________________________________________________________________
conv2d_35 (Conv2D)           (None, 48, 48, 3)         297
_________________________________________________________________
batch_normalization_17 (Batc (None, 48, 48, 3)         12
_________________________________________________________________
average_pooling2d_17 (Averag (None, 8, 8, 3)           0
_________________________________________________________________
flatten_19 (Flatten)         (None, 192)               0
_________________________________________________________________
dropout_34 (Dropout)         (None, 192)               0
_________________________________________________________________
dense_38 (Dense)             (None, 1)                 193
_________________________________________________________________
dropout_35 (Dropout)         (None, 1)                 0
_________________________________________________________________
dense_39 (Dense)             (None, 4)                 8
=================================================================
Total params: 562
Trainable params: 556
Non-trainable params: 6
_________________________________________________________________
```

Problems & Challenges

Detection model compression

Quantization

- Usually data type of TensorFlow models: 32-bit floating point numbers.
- Perform model compression before converting TF Lite model to C style array

Compressed 2x

Quantization:

Converting float 32-bit types of model weight to 16-bit float

EXPERIMENTS

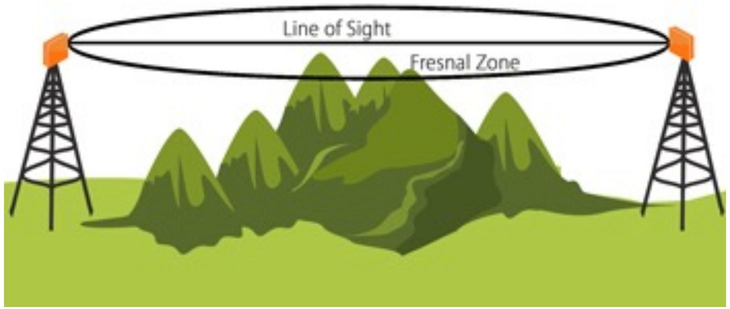
[redacted]

- Communication quality
- Drone detection test

Communication quality

Purpose: Calculating the maximum distance of communication

Long range communication technology



Secure a Fresnel Zone by drone

Make hard to measure the maximum distance

Communication quality

Calculating the maximum distance of communication

Two measures for calculating

Receiver Sensitivity		Link Budget
The minimum signal strength that a receiver can detect	$<$	A measurement of the received power, including all gains and losses

Communication quality

Calculating the maximum distance of communication

Can calculate PathLoss and maximum distance

1. PathLoss in 100m = (Link budget in 200m) - (Link budget in 100m)
2. **PathLoss in 10m** = (PathLoss in 100m) / 10

➡ The (Link budget) in **N** meter

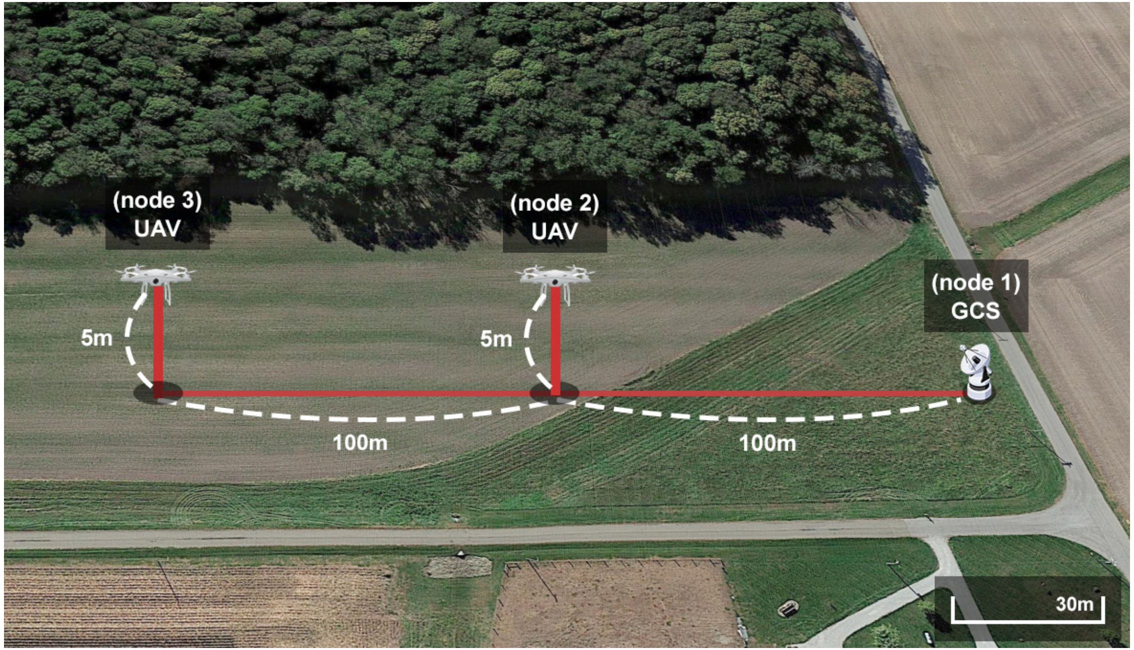
where N represents the distance that you want to calculate

➡ The maximum distance, **(M)** meter, which makes (Link budget) equal to (Receiver sensitivity)

where (M) represents the maximum distance of communication

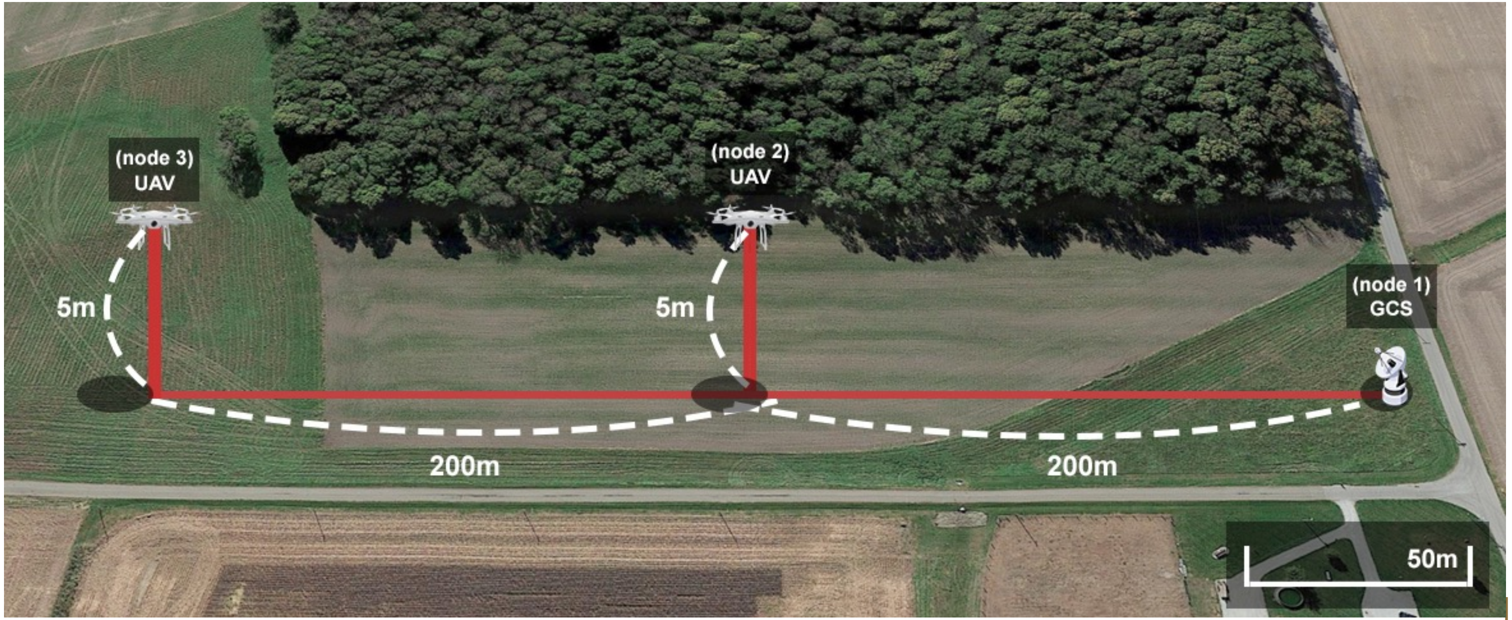
Communication quality

Environment – 100m distance



Communication quality

Environment – 200m distance



Communication quality

Experiment setting

Communication quality

Experiment result

Link Budget between two nodes

		100m communication
Destination	Ground -> Node 2	Node 2 -> Ground
AVG	-80.96296296	-86
WORSE	-84	-96
BEST	-72	-74

		100m communication
Destination	Node 2 -> Node 3	Node 3 -> Node 2
AVG	-87.09090909	-82.16666667
WORSE	-96	-87
BEST	-74	-77

		200m communication
Destination	Ground -> Node 2	Node 2 -> Ground
AVG	-97.78095238	-90.11764706
WORSE	-105	-97
BEST	-89	-82

		200m communication
Destination	Node 2 -> Node 3	Node 3 -> Node 2
AVG	-90.85526316	-90.97435897
WORSE	-98	-98
BEST	-82	-83

Communication quality

Calculating a maximum distance of communication

Calculating maximum distances

UAV-to-Ground

- **PathLoss in 10m of U2G communication** = -1.051 dBm

UAV-to-UAV

- **PathLoss in 10m of U2U communication** = -0.628 dBm

➡ Maximum distance of U2G **communication** = 676m

➡ Maximum distance of U2U **communication** = 1044m

Drone detection test

Purpose & Field experiment

- **Major purpose**

  Most important thing in this experiment is verifying that detection model is running on the system.

- **Field experiment**

  Outdoor experiment

Drone detection test

Experiment setting

- Experiment location: next to the K-SW building
- Dataset for training

  "Drone Dataset for Amateur Drone Detection and Tracking Project" in 2019

  1100 drone images

  Labeled XML file

    Information about pixel location of bounding box in drone image
- Take a photo and compute drone detection inference every 100ms

Bounding box

Detection device

[4] Drone Dataset: Amateur Unmanned Air Vehicle Detection, Nov. 25 2019, doi: 10.17632/zcsj2g2m4c.4

Drone detection test

Experiment setting

- **Optimizer: Adam**

  Improves learning rate and training stability

- **Loss function: Mean Squared Error**

  Minimizes difference between the predicted value and the real value for updating model weights

  Predicted Value & Real Value ?

  : pixel location of bounding box surrounding the drone in the image

Detection performance

Experiment result

Experiment figure

Experiment log: for 5 min

Time	Result
07:56:56	YES
07:57:02	YES
07:57:13	YES
07:58:48	YES
07:58:54	YES
07:59:11	NO
07:59:16	YES
07:59:22	YES
07:59:27	YES

Experiment result

- Could confirmed TensorFlow detection model is running on the system
- Can't guarantee detection accuracy
  - The number of trained data set is too small
  - Train data of drone images are not various

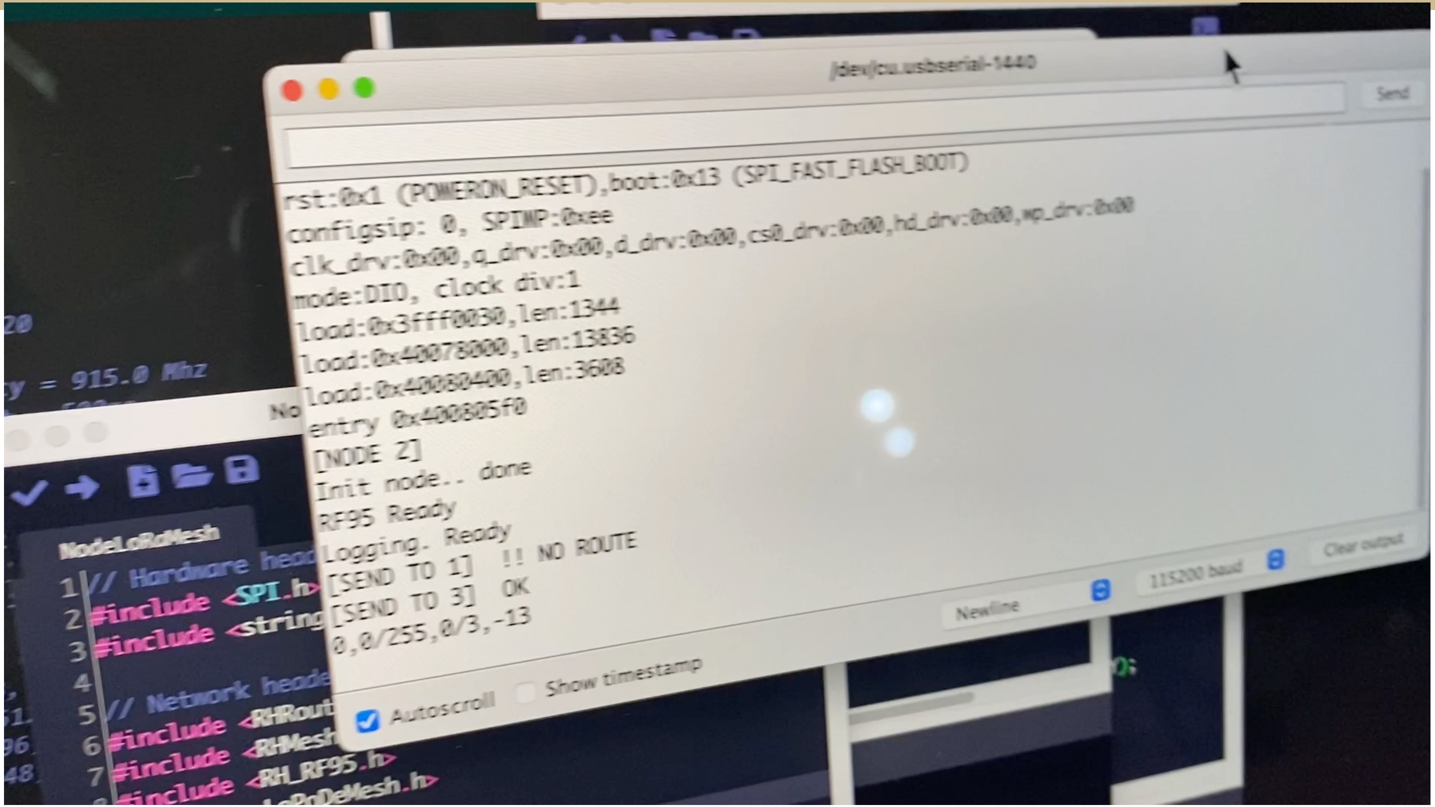
RESULT

[redacted]

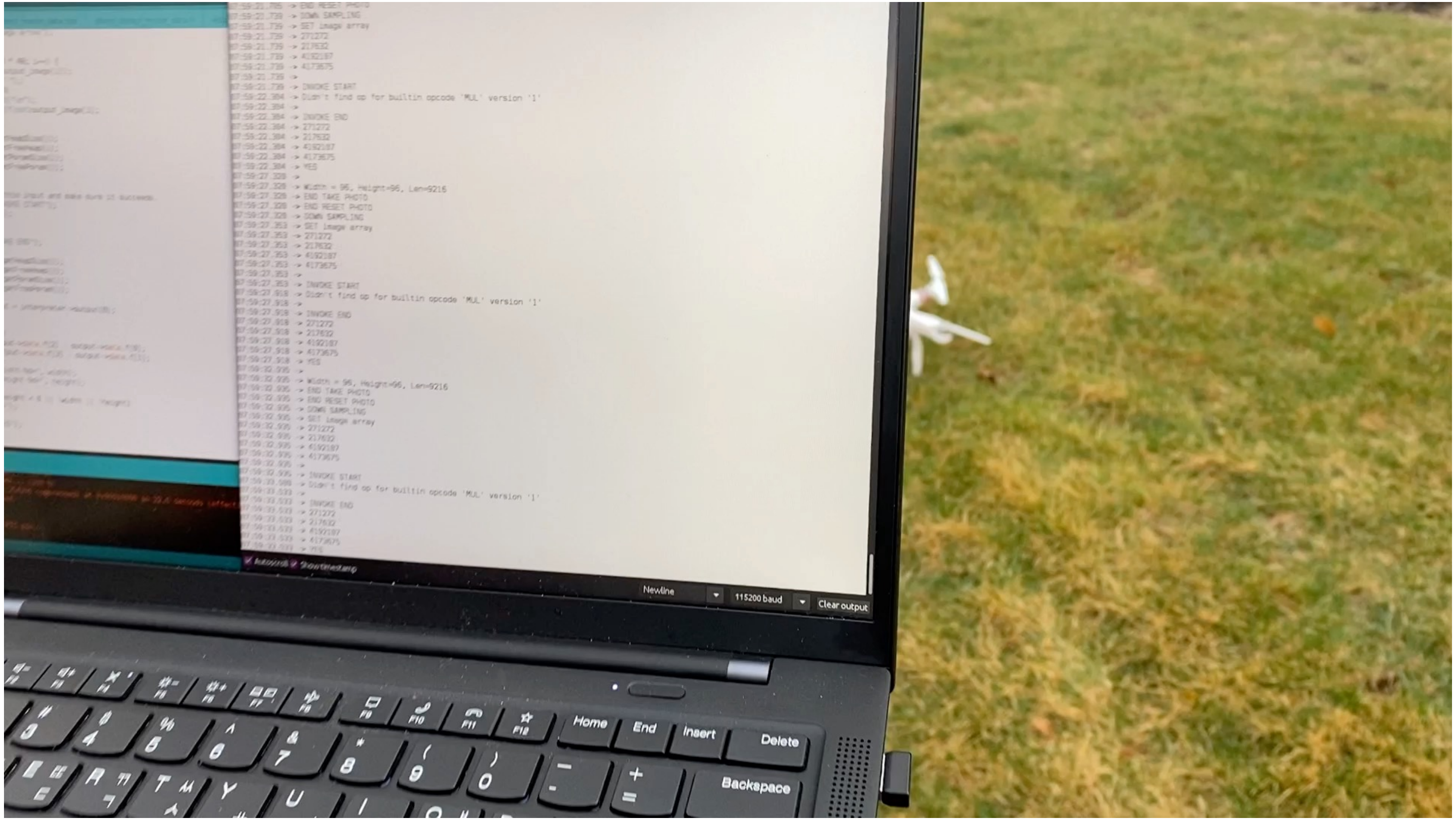
- Demo
- Conclusion

DEMO

DEMO


/dev/cu.usbserial-1440
Send
rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0030,len:1344
load:0x40078000,len:13836
load:0x40080400,len:3608
entry 0x400805f0
[NODE 2]
Init node.. done
RF95 Ready
Logging. Ready
[SEND TO 1] !! NO ROUTE
[SEND TO 3] OK
0,0/255,0/3,-13
Autoscroll
Show timestamp
Newline
115200 baud
Clear output
NodeLoRaMesh
// Hardware head
#include <SPI.h>
#include <string
// Network heade
#include <RHRout
#include <RHMesh
#include <RH_RF95.h>
915.0 Mhz

DEMO



Didn't find op for builtin opcode 'MUL' version '1'
Newline
115200 baud
Clear output
Home
End
Insert
Delete
Backspace

Result

Conclusion

- TensorFlow Lite Model can be run successfully on ESP32
- LoRa Mesh network is implemented by the system
- Failed to integrate all these components for running at the same time

Q&A

[redacted]

Thank You

Team. BTT

Soonchan Kwon, Gihwan Kim, Nawon Kim, Nahyeong Kim, Karteikay Dhuper, Prakshi Chander

polytechnic.purdue.edu

/ TechPurdue