Communications and Computational System for an Intelligent Radiation Awareness Drone (iRAD)

Ryan A. Kim, Marlee E. Trager, Andrew J.E. Kent, Jordan D. Noey
Radiological Health Engineering Lab, University of MIchigan
Kimberlee J. Kearfott, kearfott@umich.edu
Consortium for Monitoring, Technology, and Verification (MTV)

Introduction and Motivation

* Detecting and evaluating widespread radiological contamination currently requires inefficient methods of survey such as scanning over an entire area
* Creating a drone capable of performing an intelligent survey would greatly improve monitoring thoroughness, speed, and accuracy
* Such a drone requires sensors and additional computers that a typical consumer drone would not
* This work focuses on the underlying design of the communication software for such a drone

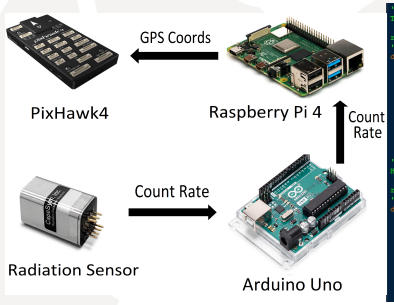
Mission Relevance

* Verification and monitoring of radiological sources is critical for material security and non-proliferation.
* This work seeks to create a faster and more efficient means of widespread monitoring to address those concerns.
* With such a tool as iRAD, it becomes possible to better identify and monitor both routine and anomalous widespread radioactive emissions

iRAD

Technical Approach

* The drone contains a PixHawk 4 flight controller, Raspberry Pi 4, and Arduino Uno
* PX4, an open-source firmware for drones has code added to interact with the various onboard computers and sensors
* Modifications to PX4 and sensor code for the Arduino will be written in C++, while driver code for the Raspberry Pi will be written in Python



▲ High-level system overview

```
'''
Take data from hasnav algorithm to be encoded into packet

returns byte string of packet
'''
def encode_packet(lat, lon, alt):
    packet = b""
    packet += generate_header(0, lat+lon+alt)
    header += lat.to_bytes(4, "big")
    header += lon.to_bytes(4, "big")
    header += alt.to_bytes(4, "big")

    packet = cobs_encode(packet)

    return packet

'''
Helper function to generate packet header

returns byte string of packet header
'''
def generate_header(dev_id, checksum):
    header = b""

    time_stamp = int(time.time_ns/1000)

    header += time_stamp.to_bytes(8, "big")
    header += dev_id.to_bytes(4, "big")
    header += checksum.to_bytes(4, "big")

    return header
```

▲ Packet encoding code on Raspberry Pi

Results

* Documentation has been written for PX4 sensor driver development for the PixHawk4
* A packet protocol has been designed for transferring data between the computers
* Code for the flight controller to receive packets and code for the Raspberry Pi to send packets has been written

Expected Impact

This work will provide a safe, fast, and efficient method of detecting radiological sources across a wide area

MTV Impact

* Working in my MTV-sponsored undergraduate team has allowed me to be involved in research with peers, graduate students, and staff
* My team has provided me with guidance on preparing and giving presentations
* My research has allowed me to explore the NNSA mission and the importance of the computer science field to it
* The ultimate iRAD would be of great use for both routine and incident-responsive radiation surveys for monitoring, decommissioning, cleanup, and surveillance for nuclear power, weapons, and existing contaminated sites

Conclusion

* The necessity of multiple computers onboard a single drone has required careful development of communications protocols
* The addition of multiple computers reduces the workload and thus increases the performance of each individual computer
* Thus far, the Raspberry Pi4 appears to be an adequate choice for running a source-seeking algorithm in terms of its performance and its input/output connections

Next Steps

Work will be continued on the communications systems and computers to create a working drone

This work was funded in-part by the Consortium for Monitoring, Technology, and Verification under Department of Energy National Nuclear Security Administration award number DE-NA0003920

National Nuclear Security Administration