Name: Rohith Suresh

Roll number: EP20B029

General Purpose Swarm Application

Introduction | Q1 | More about me

About me

Hi there! My name is Rohith, and I'm passionate about all things related to computers. I find immense joy in delving into the intricate world of software and hardware, constantly exploring and pushing the boundaries of what technology can achieve.

I have dedicated a significant portion of my time to tinkering with both software and hardware components. It's a thrilling experience for me to create and experiment with different aspects of computing.I have a drem of building an entire computing stack from scratch and have progressed somewhat on that journey through some of my projects: A single-cycle CPU (here), A chip8 emulator(here) and I am currently engrossed in the development of a compiler/interpreter (here). I am an ardent enthusiast of systems programming and loves looking "under the hood". Apart from my hands-on approach to technology, I have a deep love for writing. I maintain a blog at rohith.page, where I share my insights, experiences, and discoveries in the world of computing. It's my way of contributing to the vast knowledge-sharing community and engaging in meaningful discussions. You can see a comprehensive list of my projects in my github repo (here)

When I'm not immersed in my own projects, you can often find me actively participating in online communities. I frequently visit platforms like Hacker News, where I absorb the latest news, trends, and thought-provoking discussions from the tech world. I also enjoy sharing my thoughts and insights on Twitter.

Motivation and Takeaways

Some of the reasons why I feel like the drone swarm project would be an exciting and fullfilling endeavour for me are:

1. Multi disciplinary nature and practical application

   The drone swarm project encompasses various modules, including electrical, robotics, and communication. This multidisciplinary nature aligns perfectly with my diverse interests,and integrating these different components would be both challenging as well as rewarding for me.The project offers an opportunity to see my work in action and contribute to some extent on a somewhat nascent niche of technology.

2. Embedded Hardware Development

   Working with physical components and getting hands-on experience with circuitry, sensors, actuators, and other embedded systems and low level programming would be fun. It would be a great opportunity for me to deepen his knowledge and skills in hardware design,implementation and low level software.

3. Low-Level Networking and Communication

   As someone who enjoys twiddling with software and hardware, delving into the intricacies of low-level networking protocols, communication protocols, and network optimization would be intellectually stimulating for me and would help me implement such systems for my future endeavours(like when i build an OS :))

Past Work

Hardware

1. Single Cycle CPU

   A single cycle RISC-V CPU, written in verilog and can be run on a PYNQ Z1 FPGA. Tested for performance and correctness for the RISC-V RV32I instruction set. Can clock upto 150MHz.

2. Floating Point Adder

   A hardware multi-cycle floating point adder thats capable of adding floating point numbers of 32 bits. The floating point numbers are represented in the IEEE 754 standard. The adder is written in verilog and can be run on a PYNQ Z1 FPGA.

Systems software

1. Virtual Machine For Chip8

   A chip8 virtual machine implemented in typescript and runs in the terminal. Tested with the complete chip8-test suite and with canonical chip8 games like pong, tetris, space invaders etc.

2. Interpreter for Monkey lang

   An interpreter for a javascript-esque progamming language written from scratch with no dependencies in C.

3. Pratt Parser

   A parser implementing the pratt parsing algorithm in golang. Supports operator precedence for infix,prefix and postfix operations.

Popular Swarm Projects + Swarm Research Paper

Communication and Swarming | Q1 | Microcontroller Communication

Communication between mirocontrollers

Microcontrollers can communicate with each other in a variety of different ways. Some of them are USB, Parallel port, I²C, SPI, UART, Ethernet, WiFi, Bluetooth SPP, BLE, ISM, Cellular etc. Some of these are wired, for example USB,parallel port,I²C etc and thus pose the obvious disadvantage during a swarm flight, since the physical constraint set by the wires would decrease their degrees of freedom.

Wifi

Wifi communication modules like ESP2866 wifi module provides a serial transreciever module based on the ESP8266 system on chip. They implement the whole TCP/IP stack and has some computational power overboard(which may or may not be used, thus providing the flexibility of being an independent peripheral as well as a offloading processor that can run worker threads from the main processor).

Advantages

- Wifi modules are cheap and easy to use.
- They have a good range and can be used in a variety of applications.
- Easy to configure and ready availaibility of libraries enables easy iteration for software

Disadvantages

- Wifi modules are relatively power hungry and this can be a disadvantage on a battery operated drone.
- TCP/IP modules in software would require some significant compute and that puts a high lower cap on the microcontroller chip requirements.
- Wifi modules are not as fast as other communication modules like BLE and ISM.

Bluetooth(BLE 4.0)

Bluetooth is one of the most used wireless protocols in IOT, thanks to the recent introduction of Bluetooth low energy extension(BLE). BLE is intended for applications where the demand is for ultra-low power rather than high throughput. Microcontrollers like the Arduino Nano 33 BLE board features an nRF52840 microcontroller with 1MB CPU Flash Memory and the possibility for BLE connectivity.

Advantages

- BLE is a low power protocol and thus can be used in battery operated drones.

Disadvantages

- Depending on the message size the low throughput of the BLE protocol can be a disadvantage.
- Limited range of communication as compared to TCP/IP.
- BLE is not as widely used as TCP/IP and thus the software support is limited.(as compared to TCP/IP)

Cellular

Cellular communication is a type of communication that uses cellular networks to transmit data. Cellular communication is a good option for long range communication.

Advantages

- Cellular communication is a good option for long range communication.

Disadvantages

- The installation of antennas for cellular network can be cumbersome

Zigbee

Zigbee is a low-power, low data rate, and close proximity (i.e., personal area) wireless ad hoc network. The technology defined by the Zigbee specification is intended to be simpler and less expensive than other wireless personal area networks (WPANs), such as Bluetooth or more general wireless networking such as Wi-Fi. Applications include wireless light switches, home energy monitors, traffic management systems, and other consumer and industrial equipment that requires short-range low-rate wireless data transfer.

Advantages

- Zigbee is a low power protocol and thus can be used in battery operated drones.

Disadvantages

- Low throughput of the Zigbee protocol can be a disadvantage.
- Limited range of communication as compared to TCP/IP.

Communication and Swarming | Question 2 | TCP/IP

Network Communication Protocols

A network communication protocol is a set of rules and conventions that govern how data is transmitted, received, and processed over a computer network. It defines the format, timing, sequencing, and error control of data exchange between devices or systems within a network.

Protocols enable different devices or software applications to communicate effectively, ensuring that data is transmitted accurately and reliably. They establish a common language or set of rules that all participating entities understand and follow, allowing them to exchange information seamlessly.

TCP/IP Stack

Lets imagine an ethernet cable(or even simpler a cable of copper,or maybe you have an optical fibre: which has a cladding that has a different refractive index than the strand and that causes TIR and will propogate to the reciever). How can computers communicate throught the internet using this cable?

Well we can give a voltage to its ends and vary it. Computers communicate in binary so we need a ref high and low voltage and the signal is sent as a series of high and low voltages(Light on/off can be the signal for optical fibres). What about wifi routers?They send signals as radio waves of a certain frequency(say a 2.4GHz sine wave).A device picks up these radio waves(using a reciever).By shifting the phase of the signal sent we can communicate the serial bits. These constitute the **physical layer** of a network connection.

We know that CPUs execute instructions based on the tick of a clock. Whilst communicating data, we need to ensure that both the sender and reciever are in sync. This is done by sending a clock signal along with the data.Now we have sent the serial bits and the clock signal.Say you connect an oscilloscope to the ethernet cable and you plot the bits that are sent. You will see that the bits are not sent as a continous stream of bits but rather in packets. These packets(payload) are sent with a header and a footer. The header contains the source and destination MAC address(unique for each computer, you can check yours by doing `ifconfig -a | grep ether` on a linux machine) and the footer contains a frame check sequence(This is used by reciever to check corruption).This is the **data link layer** of a network connection.

What if there is no physical connection between the computers as in our case(say in the case of wifi there is a computer not connected to a wifi, or a computer not connected to your network switch in case of ethernet). Basically how do we communicate **inter** networks? Well we use an **Internet Packet(IP)** embedded in an ethernet payload.

Each router constituting a network has a forwarding table(A table containing a list of **IP addresses** right? Well not quite, it can contain a subnet address, something that looks like `172.17/16`, meaning the first 16 bytes of the ip is 172.17: there can be multiple such matches and it does a **longest prefix match**) and the router forwards this connection to the matched router/ gateway.

Let's take a solid example. Say you make a request to `https://google.com`. Now there's aweb server running on some machine inside google's warehouse in it's corresponding subnet. How can we construct our ethernet packet? We only know the IP (We know the IP because the domain resolves to the IP address thanks to DNS). First I send my ethernet packet to all the networks(**broadcasting**) in our subnet(This can be done by setting destination mac as `ffffff..`). The payload here is going to be an **ARP(Address resolution protocol)** packet. The router that has the corresponding ip in its forwarding table responds with the MAC address(with a different opcode:[1]) of the reciever(which is the mac of the machine containing the webserver). Practically you can try sending an IP(not really, its ICMP but similar in essence) packet to google's server using `ping google.com`. You can inspect the mac address using a packet watcher like `wireshark`. This constitutes the **network layer** of a network connection.

Now we move to the core of the question. What exaclty is TCP/IP? Well TCP/IP is a set of protocols that are used to communicate between computers. TCP/IP is a **protocol suite** that contains a set of protocols that are used to communicate between computers. **TCP** stands for **Transmission Control Protocol** and **UDP** stands for **User Datagraph Protocol**.

Links in the network can be congested(due to lesser baud rates of the network and or multiple link ins and outs). TCP provides a **byte stream service**: Basically a connection is established between the sender and reciever via a set of request responses. First a syn is sent to the reciever,reciever sents back syn-ack and the sender sends the ack signals. Once this has taken place the stream of bytes can seamlessly be sent(You can think of TCP as the phone call and the IP packet as the conversation, the phone call has to connect and the reciever should take the call for the conversation to happen).

Also the TCP connection exists between **ports** (A port maps the webserver process/network service in the OS). TCP also has a checksum footer for checking for corrupt data/data loss. In case the reciever communicates that a packet has not been recieved succesfully, TCP can resend the data.

UDP is kind of similar to TCP in the sense that its is also used to send and recieve data online. The fist stark difference is that UDP is connectionless. Next **UDP** does'nt guarentee 0 packet loss.(It's like a megaphone making an announcement as compared to a phone call in case of TCP). **UDP** is faster than TCP because it does'nt have to establish a connection and it does'nt have to resend data in case of packet loss. UDP is used in cases where packet loss is acceptable like in video streaming, online gaming etc.

The differences between TCP and UDP are:

TCP	**UDP**
Connection oriented	Connectionless
Reliable	Unreliable
Slower	Faster
Used for HTTP, FTP, SMTP etc	Used for video streaming, online gaming etc
Has a header of 20 bytes	Has a header of 8 bytes
Has a checksum	No checksum
Has a sequence number	No sequence number
Has a flow control and congestion control mechanism	No flow control and congestion control mechanism
Has a 3 way handshake	No handshake

Communication and Swarming | Q3 | Microcontroller communication

The github repo with the code: here. Further details are listed in the `README.md` file

Communication and Swarming | Q4 | Wifi Server

The github repo with the code: here. Further details are listed in the `README.md` file

Communication and Swarming | Q5 | Real Time Kinematics

Real Time Kinematics

- Real Time Kinematics is a positioning technique to provide highly accurate real-time positioning information and is used in surveying, precision agriculture and autonomous vehicles
- Fixed base station + one or more rovers(rovers form the GNSS reciever)
- Base station recieves signals from GNSS(Global navigation statellite systems) satellites such as GPS(Global positioning system) or GLONASS
- The technique is primarily based on carrier phase measurement techniques.
- The carrier phase measurement is a measurement of the beat frequency between the received carrier of the satellite signal and a receiver-generated reference frequency.
- Base station computes its location by using the signal recieved from the GNSS satellites based on the carrier phase measurement techniques
- It then compares its location and the actual location to generate a correction signal
- Correction signals are passed to the rovers and the rovers use this correction signal to correct its location data recieved from the GNSS satellite to improve precision.(The rovers also use carrier phase measurement to compute their location)
- On the base station there are 3 components:
  1. Antenna: To pick up the GNSS satellite signal
  2. Radio Modulator: Convert the correction signal to a radio signal
  3. Amplifier: Amplify the radio signal thereby increasing the range of the signal.
- The computation of the correction signal can take some time and this can cause a delay in the transmission of the signal to the rover. This can lead to a delay in the correction signal and thus the correction signal can be outdated. Thus the correction signal sends a range rate correction with them.

How to we connect 100s of drones?

- Let's assume there are 100 nodes.
- There are 3 popular ways of implementing RTK communication layer:
  1. NTRIP
     - **NTRIP** (Networked Transport of RTCM via Internet Protocol) is a protocol that enables streaming of RTK correction data via the internet over common TCP/IP methods. The system consists of two parts which communicate via the internet: the server side and the rover side.
     - The server is responsible for receiving data from the base station and rebroadcasting them to the rover via TCP/IP.
  2.