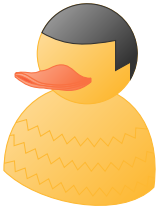
STEM, EECS & A Little Touch on Cloud

Montgomery Hackathon 2021 Keynotes

Speaker: Changxu Luo, Software Engineer @Google

github.com/camelboat

Outlines



"This is a 45-min presentation with 15-min Q&A"

A Little About Me

- BS in Electrical and Computer Engineering @Shanghai Jiaotong University
- MS in Electrical Engineering @Columbia University
- Worked as Software/Electrical Engineer @GE Appliances, Bosch, and Rover Diagnostics
- Software Engineer @Google - Google Cloud Networking Control Qualification
- Research Assistant on distributed storage, instructed by Prof. Asaf Cidon.

A Little About Me, Cont.

Figure: Project Expo at UMich

A Little About Me, Cont.

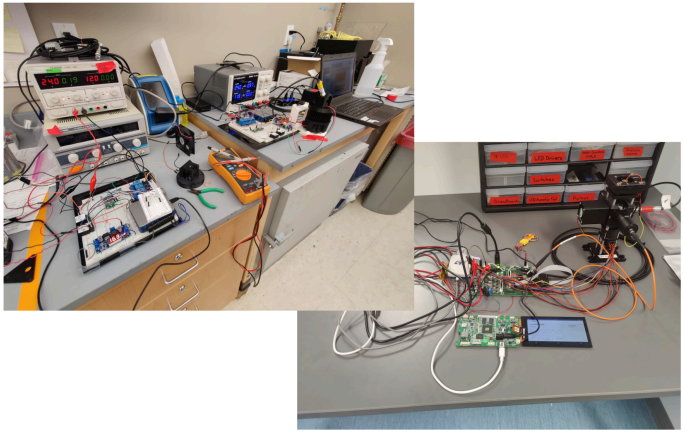


Figure: Prototypes developed at Rover Diagnostics

A Little About Me, Cont.

At Google, I've worked for two teams:

1. Assistant and Search(internship, Google Assistant Feature Development for Google Home)
2. Google Cloud Platform(current position, GCP Networking Release Qualification Framework)

No worries, you don't have to understand my work now ☺

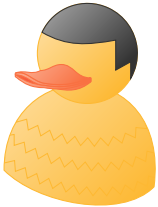
(and I am not able to show you any details ☹, but wait for the second session for the Cloud ice-breaker!)

STEM?

"What is STEM?"

STEM?

"What is STEM?"



"I am glad you asked!"

STEM? Cont.

STEM

Science, technology, engineering, and mathematics(STEM), a term to group together these academic disciplines[1].

And if we name some of those disciplines(with the narrow definition[2])

- Science: Physics, Chemistry, **Computer Science**, etc.
- Engineering: **Electrical Engineering**, Computer Engineering, Mechanical Engineering, etc.
- Mathematics: Mathematics

[1] https://fas.org/sgp/crs/misc/R42642.pdf

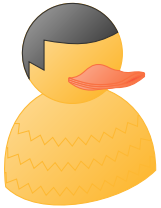
[2] In a general definition, all these disciplines can be a form of science, such as Formal Science(Math), Natural science(Physics, Chemistry), Applied science(Engineering)

STEM? EE? CS?

Relations of Science, Technology, Engineering, and Math(with the narrow definition)

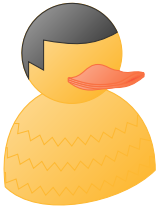
1. Math is the language for describing science(study of the natural world).
2. Engineering is the systematic and iterative approach to the human applications of science.
3. Technologies are those applications.

STEM? Cont.



"Let's see an example technology, say Wi-Fi"

STEM? Cont.

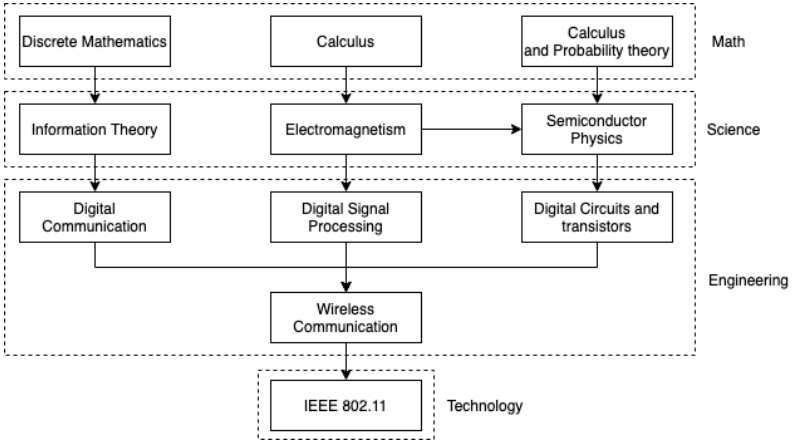


"What's Wi-Fi?"

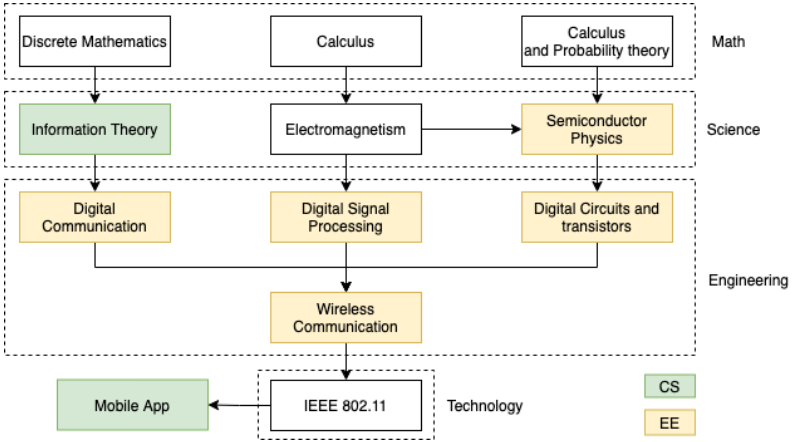
Wi-Fi

- A family of wireless network protocols, based on the IEEE 802.11 family of standards.
- Allowing nearby digital devices to exchange data by radio waves.

A VERY brief overview for Wi-Fi knowledge stack



A VERY brief overview for Wi-Fi knowledge stack, Cont.



EE, Overview by Phrases[4]

Electrical Engineering

Study, design and application of equipment, devices and systems which use electricity, electronics, and electromagnetism.

- Computer Engineering(often an individual discipline nowadays)
- Automation and Control
- Power and Energy
- Electronics
- Telecommunications
- Signal Processing
- Optics and Photonics [3]

[3] I simply copy them from Wikipedia, not a strict catalog.

[4] If you want to have a more details among which courses should you expect to take in EECS, visit the curriculum map at https://ocw.mit.edu/courses/mit-curriculum-guide/

CS, Overview by Phrases

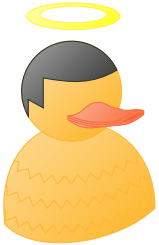
Computer Science

Study of algorithmic processes, computational machines and computation itself[5].

- Theoretical computer science(Computation Theory, Information and coding theory, Data structures and algorithms, Programming language theory and formal methods, etc.)
- Computer systems and computational processes(Artificial Intelligence, Computer architecture and organization, Concurrent, parallel and distributed computing, Computer networks, Cryptography, Databases, Graphics, etc.)
- Applied computer science(computational science, finance and engineering, software engineering, social computing and human-computer interaction, etc.)

[5] https://www.cs.york.ac.uk/undergraduate/what-is-cs/

The Magic of EECS



"Magic!"

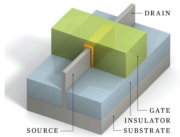
To me(as well as many others), the magic of EECS is that for the first time in human history, it frees us from following the restriction of physical laws and allows us to create and present the logics that is never seen before(just think about any video games!), and applies the logics that can be proved before(for math development).

And as a student in EECS, you can be part of the magic!

The Magic of EECS[6]


Sand

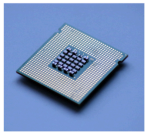

Silicon Ingot and Wafer



FinFET Transistor(a LOT)

Software Application


Operating System


CPU

Storage

[6] I found a great video made by Intel for the introduction of making a microchip: https://www.youtube.com/watch?v=_VMYPLXnd7E

Why EE? Why CS?

In EE, you can learn how to manipulate the electrics physical laws and how it grows until something that can bear pure logics(Sand→Instruction Code, Signal→Digital Data).

↓

```
def main():
  print('Hello world!')

if __name__ == '__main__':
  main()
```

↓

In CS, you can then learn to extend this logic and make system that is large-scale, complex, based on the human logic laws(system interfaces) instead of physical laws.

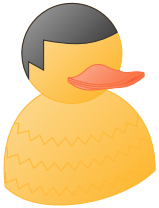
Why EE? Why CS?

Of course, this is not the whole story, since EE and CS are considered together in so many ways. Just name two:

- The signal processing requires complex algorithms and computer science theories, and before applying all the fancy artificial intelligence algorithms, it is better for you to understand what is happening in the physical world.
- Programmer needs sophisticated understanding on hardware in order to get the best software performance. For beginners, you may feel like the resources on a system is unlimited, unfortunately, they are limited[7].

[7] This is a hard job which is pursued by a lot of advanced software community like operating system, scientific computing, and programming languages communities.

Why EE? Why CS?



"This is a good time!"

Thanks to the development of computation hardware and open-source communities, we have prototyping hardwares and software tools that are highly accessible.

Resources to Get Started!

Prototype Hardware

- Arduino
- Raspberry Pi

Workstation

- Any personal device with a browser[8]!(thanks to the cloud services)

What Language to Learn

- TL;DR, Python, or follow your course instructors

[8] Maybe also a USB port.

Recourses to Get Started! Cont.

Documents

- Go to your target's official website and follow the tutorial.
- Search engine is powerful.

Sites you want to know as a beginner

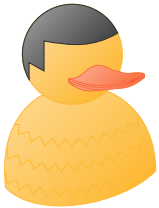
- `github.com`
- `colab.research.google.com`, run Python notebook on cloud freely
- `codepen.io`, front-end dev playground

Recourses to Get Started! Cont.

"And feel free to reach out me!"

changxu.luo@columbia.edu

What is Cloud?



"Anyone heared of Cloud Gaming?"