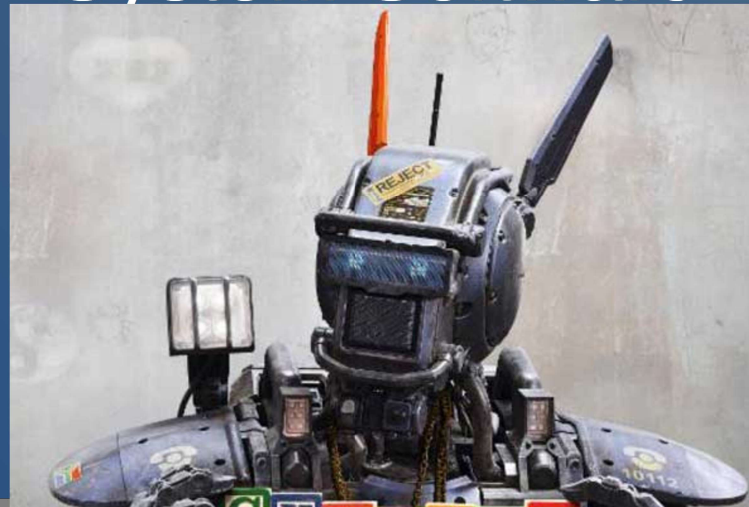


# Welcome to ENSC 351 Embedded and Real Time System Software



25-09-07

Slides #1

© Dr. B. Fraser, M. Stewart

1

# Thank you Dr. Brian Fraser

- 1) Course is based on CMPT 433 developed by Dr. Brian Fraser
- 2) Similar to the offering of ENSC 351 offered in 2022 & 2023
- 3) Best parts are probably him, mistakes are mostly mine.

# Topics

- 1) Introductions
- 2) What's an embedded system?
- 3) Course overview
- 4) BeagleBone preview
- 5) Project introduction

# About Us all (students)



Results of our survey! (2022)



# What do I notice?

- **Worried & Excited**: Top most common words!
- **Many different backgrounds**
  - We won't assume much about **Linux** use
  - We won't assume much about **programming** experience (especially not **multithreaded..**)
  - **We'll cover everything!**

# Guide to Slides

- **Slide Colour Guide (often...):**
  - **Green:** headings.
  - **Yellow:** Highlighted text.
    - This course has **no exams** but some **quizzes**.
  - **Blue:** Term being defined.
    - **Hour:** 60 minutes.
  - **Sweep-in Text:** Blanked out text.
- **Joke:**
  - There are 10 **types of people** in the world...

# Course Expectation

- Only one thing
  - Use a positive tone for all communication (asking questions, on Piazza forums, with TAs)
  - Anon trolling hurts someone. It won't be tolerated
  - Students have wide range of backgrounds; respect it
- If sending a message
  - Give a little context (class, your name, topic, ...)
  - Email: If you are sending more than 2 a week, may be too many.

# Getting Help

- **Big Class**
  - 228 students.
  - I would love to spend time with each of you but that will be difficult.
- **Use the TAs**
  - We have 8 of them!!
  - Many of them have previous experience with the course.

# What is an embedded system?



# What is an Embedded System?

- **Real-time Embedded System:**

- Embedded systems:

A Computer system designed to do...

(wikipedia).

- Real-time systems:

Must respond to events..

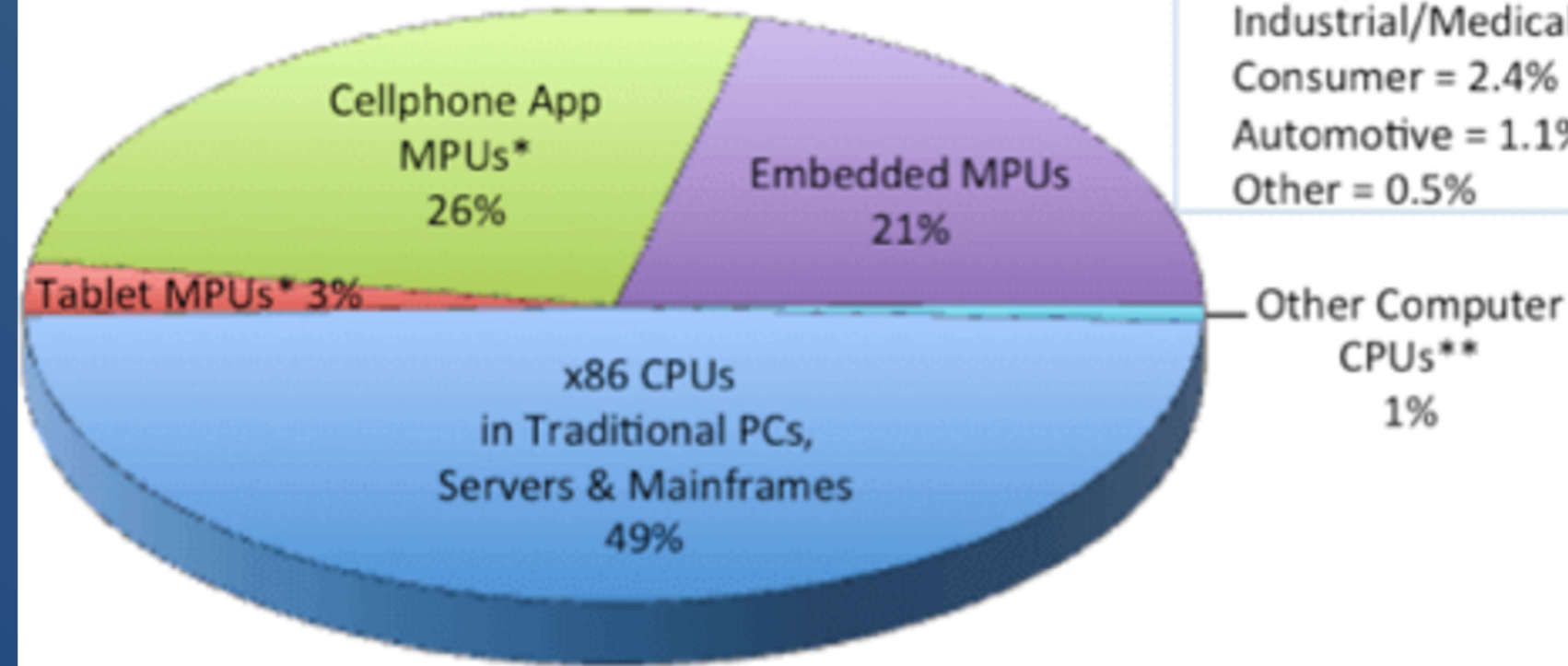
within a predictable specific time constraint.

- **Spectrum of Examples:**

- Controller in an **AA-battery recharger**.
- Controller in a **laser printer**.
- Air-quality controller on **international space station**.
- Control software in an **autonomous vehicle**.

# Inspirational Statistics - \$ CPU Sales

## 2020 MPU Sales by Application (Fcst, \$79.3B)

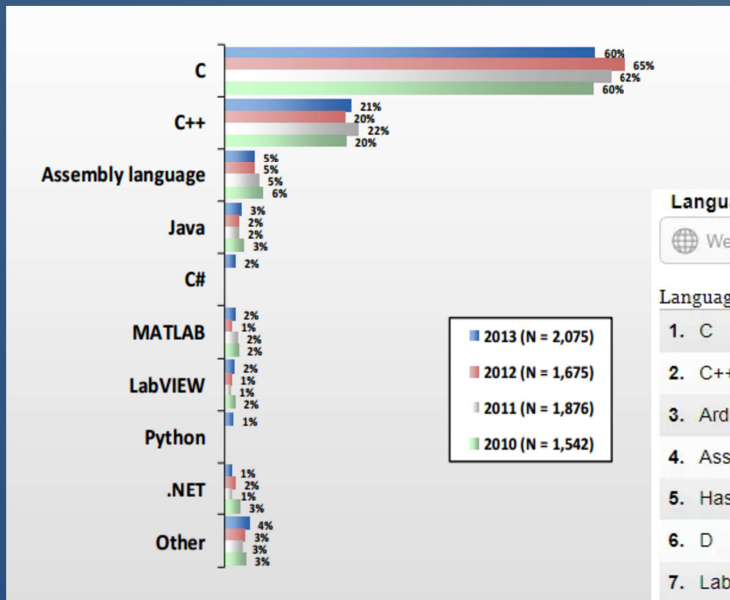


*Embedded Microprocessors = 21%*  
Communications = 9.4%  
Computers & Peripherals = 1.9%  
Industrial/Medical = 5.8%  
Consumer = 2.4%  
Automotive = 1.1%  
Other = 0.5%

\*Includes ARM-based and x86 processors. \*\*Includes ARM-based and other RISC processors.

Source: IC Insights

# Inspirational – Language Choice for Embedded



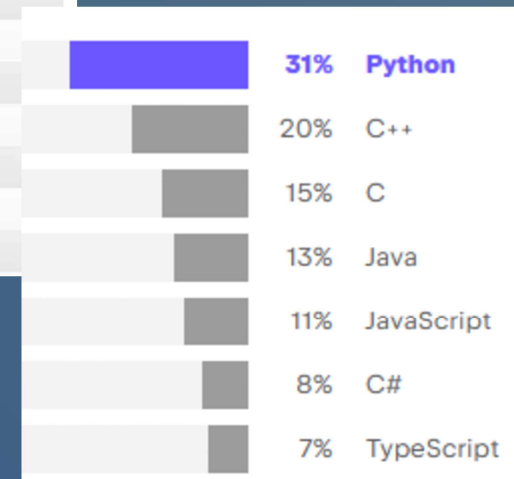
From ~2013

Language Types (click to hide)

Web Mobile Enterprise Embedded

Language Rank	Types	Spectrum Ranking
1. C	Web, Mobile, Enterprise, Embedded	100.0
2. C++	Web, Mobile, Enterprise, Embedded	95.9
3. Arduino	Embedded	69.9
4. Assembly	Embedded	68.6
5. Haskell	Enterprise, Embedded	44.1
6. D	Web, Embedded	38.9
7. LabView	Enterprise, Embedded	35.7
8. VHDL	Embedded	35.4
9. Ladder Logic	Embedded	28.1
10. Erlang	Enterprise, Embedded	28.0
11. Verilog	Embedded	26.7

IEEE ~2016



Jet BRAINS (2023)

[http://images.content.ubmtechelectronics.com/Web/UBMTechElectronics/%7Ba7a91f0e-87c0-4a6d-b861-d4147707f831%7D\\_2013EmbeddedMarketStudyb.pdf](http://images.content.ubmtechelectronics.com/Web/UBMTechElectronics/%7Ba7a91f0e-87c0-4a6d-b861-d4147707f831%7D_2013EmbeddedMarketStudyb.pdf)  
<https://deepbluembedded.com/programming-languages-for-embedded-systems/>  
<https://www.jetbrains.com/lp/devecosystem-2023/embedded/>

# Embedded System Development

- **Cross-compiling:**

- Development done on the PC using powerful tools: editor, compiler, debugger, etc.
- Compiled code...



Compiled Code



# Discussion

In groups of 3 to 4 people:

- Exchange **email address**;
- Answer the following:

1. What are **5** different **embedded systems** in your rooms right now? Which is **most interesting**?
2. What one **computer failure** could be **most life critical**? Is it an embedded system?
3. What is the **best** or **worst thing** an embedded system could be used for?



# Course Overview

LINUX: A TRUE STORY:

WEEK ONE

HEY, IT'S YOUR COUSIN  
I GOT A NEW COMPUTER  
BUT DON'T WANT WINDOWS.  
CAN YOU HELP ME  
INSTALL "LINUX"?

SURE.



WEEK TWO

IT SAYS MY XORG  
IS BROKEN. WHAT'S  
AN "XORG"? WHERE  
CAN I LOOK THAT UP



HMM,  
LEMME  
SHOW YOU  
MAN PAGES.

WEEK SIX

DUE TO AUTO-  
CONFIG ISSUES, I'M  
LEAVING UBUNTU  
FOR DEBIAN.



UH  
OR  
GENTOO.  
UHOH.

WEEK TWELVE

YOU HAVEN'T ANSWERED  
YOUR PHONE IN DAYS.

CAN'T SLEEP.  
MUST COMPILE  
KERNEL.



I'M  
TOO  
LATE.



PARENTS: TALK TO YOUR  
KIDS ABOUT LINUX..  
BEFORE SOMEBODY ELSE DOES.

# Course Overview

- **Goal**
  - Qualified for junior embedded software developer.
  - Course mostly...
    - *May* spend hours **solving build issues**, and **downloading code to device**.
- **Course Components**

Embedded  
Basics  
&  
Hardware

Linux  
Coding  
& Admin

Real Time  
system

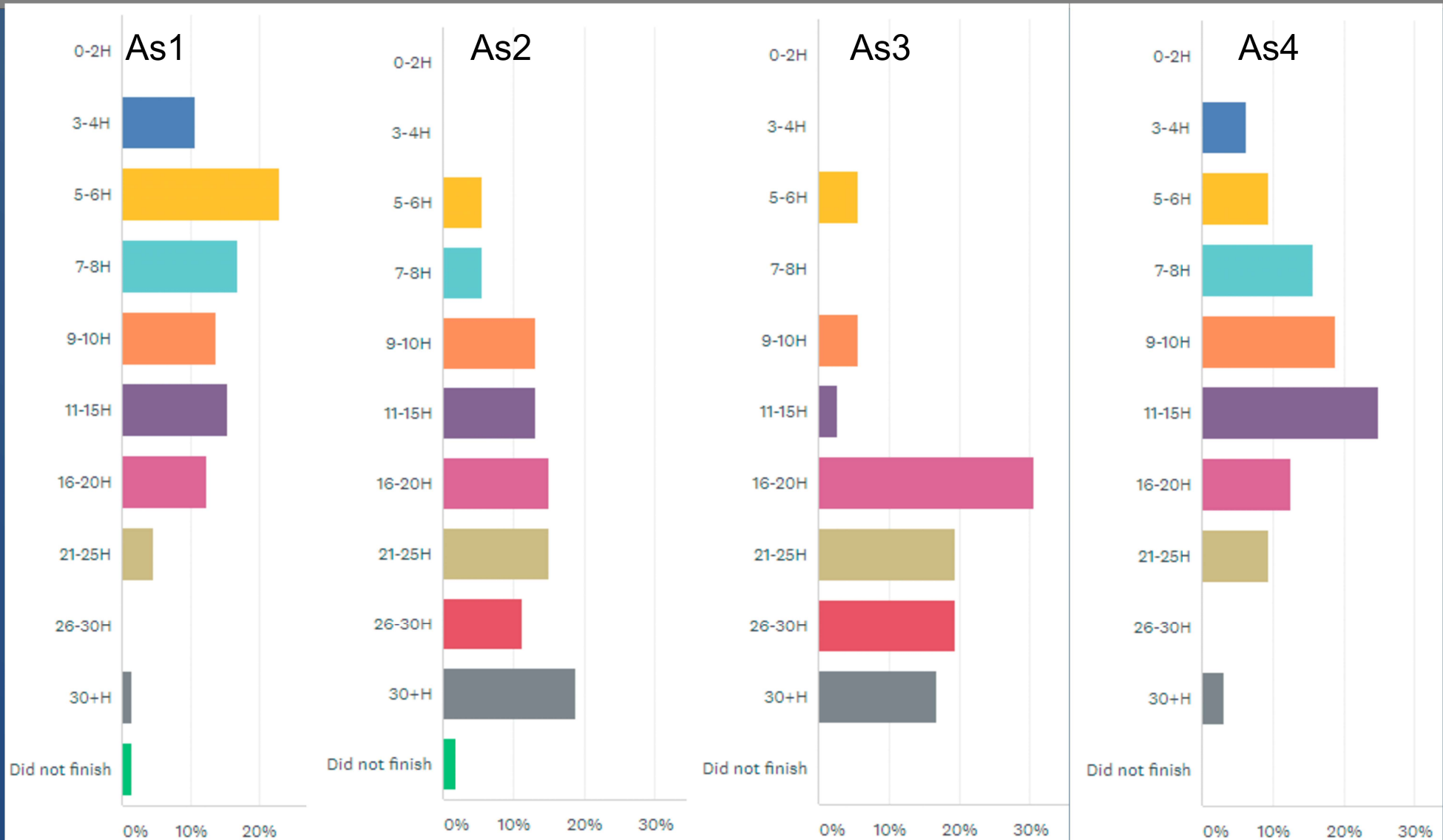
# Course History

- Based on CMPT 433: Embedded Systems
  - Here in ENSC 351 we'll spend more time reviewing and learning:
    - C programming
    - Linux programming (files, mutexes, ports, ...)
  - We are likely going to skip some CMPT 433 content:
    - Circuits
    - Kernel drivers and Bare metal
    - Web pages
  - Like 433, we'll learn all about how to develop under Linux, and take pride in our code!
    - Throughout the semester we'll ensure students of all backgrounds are having success.
- 25-09-07      – We will add time on topics as needed

# What to expect

- Previous students have found this course:
  - very rewarding to do so much hands-on, and
  - very time consuming to do so much hands-on!
- So be ready for:
  - A lot of C/Linux programming
  - Steep initial learning curve working with real HW
  - Group work
  - Spending good time on this course each week.
- Stay on top of assignments and how-to guides.
- Submitted code may anonymously be discussed in class

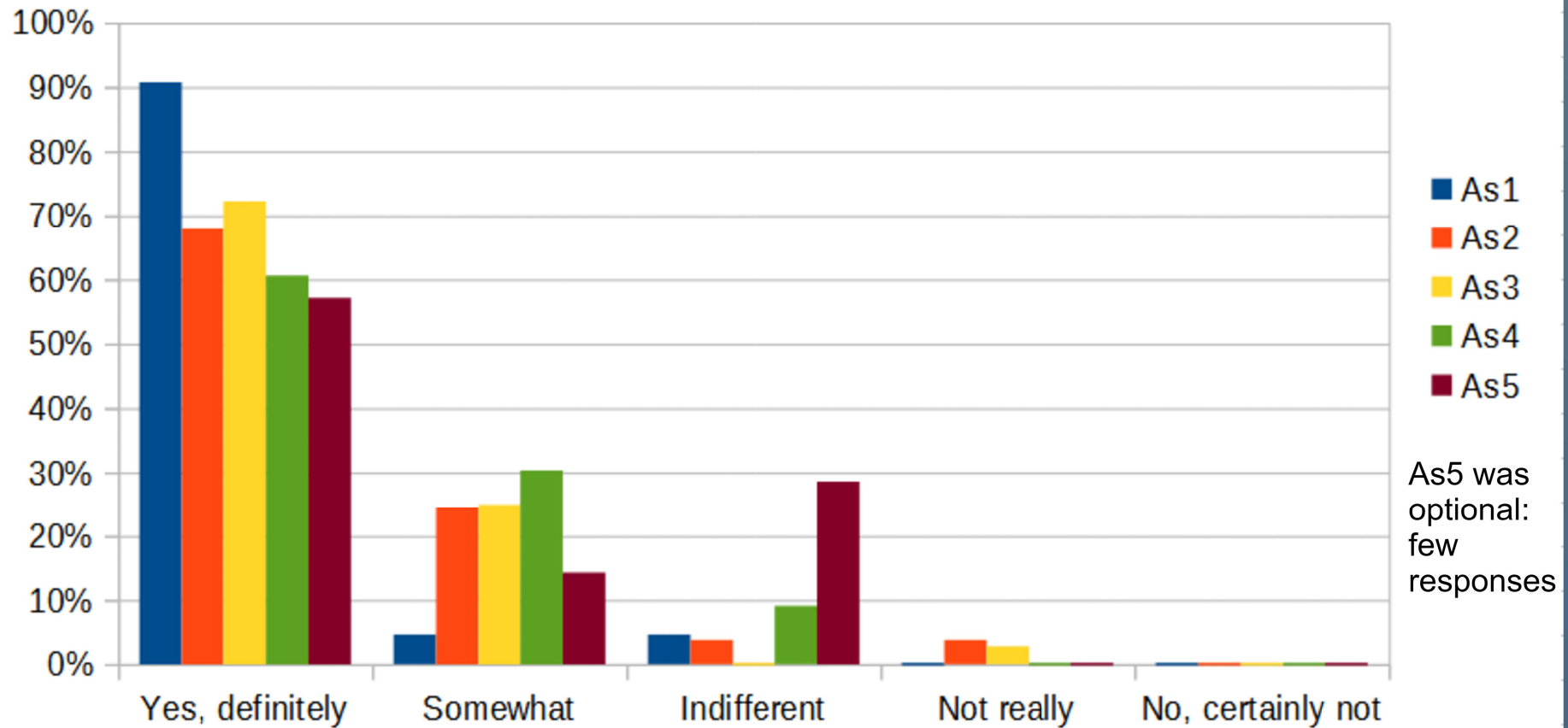
# Hours spent working on assignment (2021 Spring)





# Learning worth the time (2021-Spring)

Do you feel the time you spent on the assignments were worth what you learned from them?

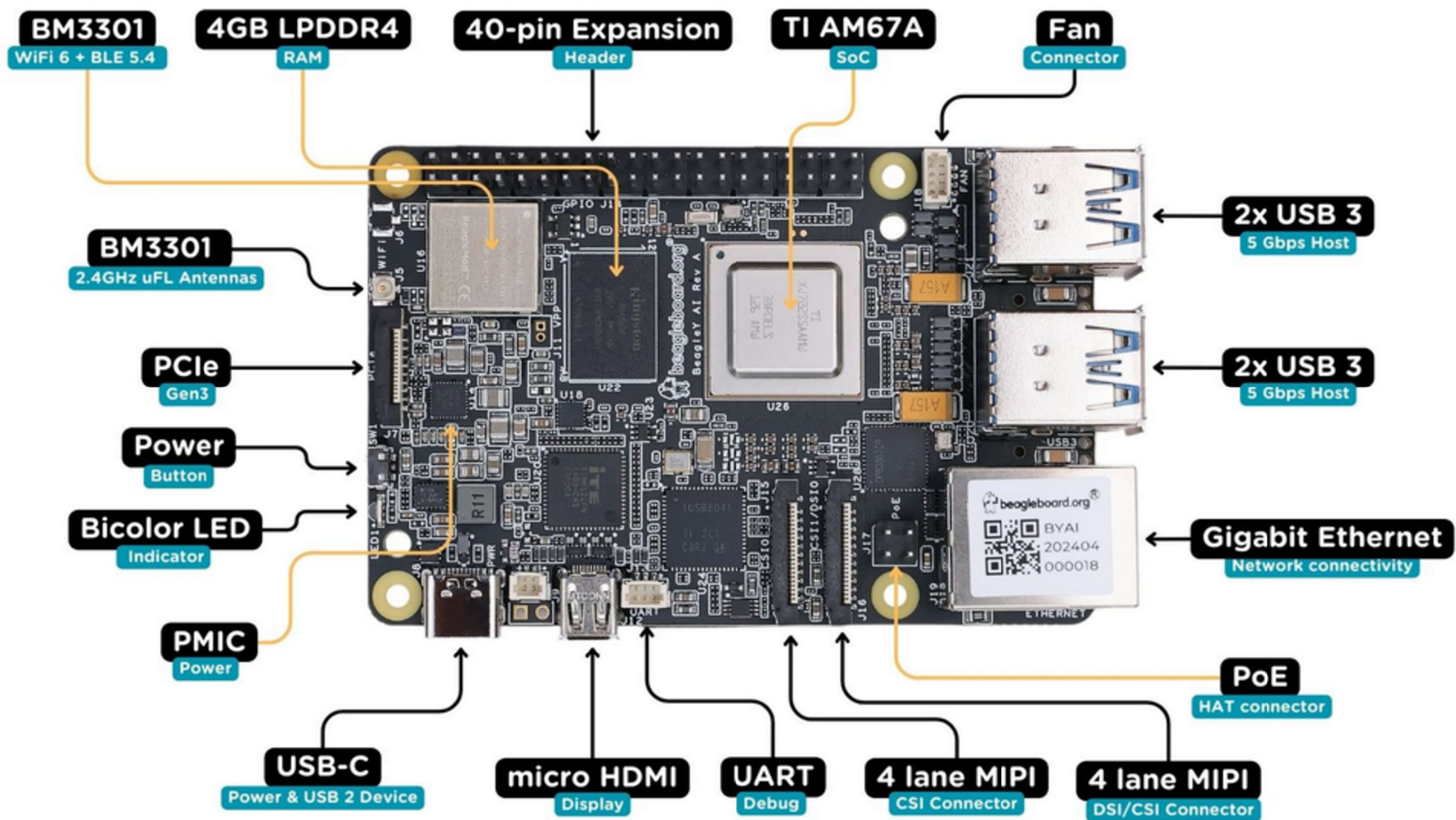


# Admin Review

- **Assessment**
  - **Quizzes** 30%
  - **Assignments** 40%: Individual/pairs to learn skills  
*Anon code reviews in class may feature your code!*
  - **Project** 30%: group (3-4) to accomplish more
  - Grade breakpoints (“% for B+?”) may be non-standard
- **Academic Honesty**
  - I am *passionate* about proving who did their own work.
  - **Corollaries:**
    - I'll give you credit for the work you do.
    - I'll catch those who don't do their own work.

# Hardware Package

# BeagleY-AI (BYAI)





# Zen Hat (What you don't get)

LED Light Strip  
Connector  
(GPIO)

LCD Display  
(via SPI)

Headers  
to connect  
other  
hardware

Audio OUT

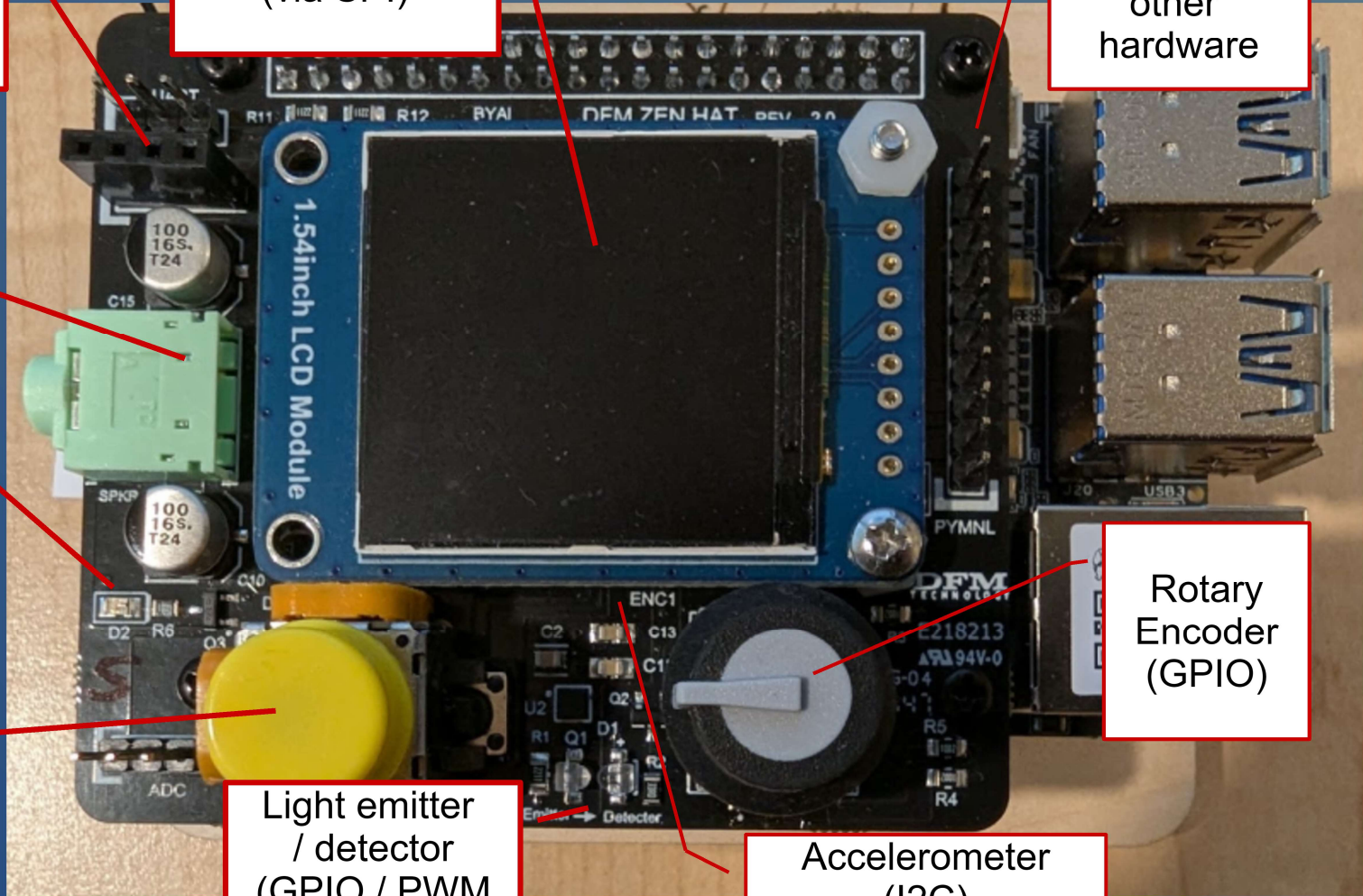
LED  
(GPIO)

Analog  
Joystick  
(ADC)

Light emitter  
/ detector  
(GPIO / PWM  
& Analog)

Rotary  
Encoder  
(GPIO)

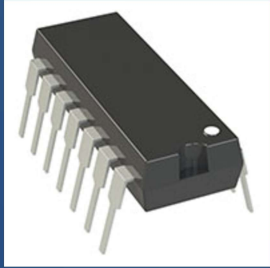
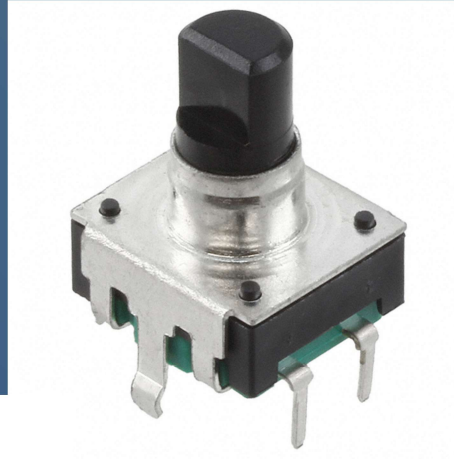
Accelerometer  
(I2C)





# Components

- Kit will include hardware, such as the following:



# Demo

- **BeagleY-AI Demo**
  - Boot & show in terminal
  - Linux commands: ls, cd, echo
  - Blink LEDs
  - Ethernet ping

# HIDDEN: First day demo script

First day demo:

- Connect via Serial & Reboot
- Log-in, check IP address, ping host  
    <OR> ssh root@192.168.7.2
- Twiddle LEDs
  - cd /sys/class/leds/beaglebone:green:usr0
  - cat trigger
  - echo none > trigger
  - echo 1 > brightness
  - echo 0 > brightness
  
  - echo timer > trigger
  - echo 250 | sudo tee delay\_off
  - echo 500 | sudo tee delay\_on

Other ideas:

- Mount NFS
- Cross-compile on host, run on target.
- Copy executable to target.
  
- Edit .profile
- Create mount NFS script via echo
- Create issue on host

# Project

- Groups of 3 or 4
- Like the course, can have a hardware component, but should have a software focus
- Must have significant amount of code
- Should have some interface with an outside system

# Project ideas

- Following pages provide some concepts for project ideas. Taken from Dr. Fraser's past offerings.
- Note: that some of these idea may need to be scaled up or down to be a reasonable size

# Interactive Educational System

Create an interactive system to help the user learn something. Think of a display at a museum and how it can engage people and help them explore some topic at the same time. Pick an area you care about (history of sports, local history, science, art) and then think of engaging ways for the user to interact with the content. What hardware can you use for physical input? How can you make the system respond to the user's input? Make it more than just a button, bring in image processing (camera), motion or distance sensors, joystick, knobs, ....



# UN Sustainable Development Goals

Think about one of the UN sustainable development goals.

- How could you make a product to help somewhere?
- Clean water (rain water capture)?
- Clean energy (smart house power, power monitor)?
- Reduce inequality?
- There are 17 goals! Make a difference!

# Reconciliation or Explore Cultures

Create a device that helps people learn about another culture. Imagine you are hired by a museum to create an interactive exhibit which highlights your culture. What neat ways could someone interact to explore? Maybe a cube they turn over to show different points in history? Maybe a map with buttons to explore places? Maybe a wheel that lets them select different dates in history?

# Sustainability SFU

- Create a device that helps people learn about another culture. Imagine you are hired by a museum to create an interactive exhibit which highlights your culture. What neat ways could someone interact to explore? Maybe a cube they turn over to show different points in history? Maybe a map with buttons to explore places? Maybe a wheel that lets them select different dates in history?

# Or some less heavy projects

- You don't all need to save the world.
- Some ideas for some lighthearted but not necessarily easier projects

# Transportation

- Use the BYAI to enhance your transportation: Car, Bike, Skateboard, Wagon! Add GPS, add remote locating, add music, add light-show connected to the motion/vibration/acceleration of your transportation. Add an LED light panel to display something, add some buttons for control. Add a locking mechanism? Locate via GPS?

# Game

Create a physically interactive game. Either make it like a standard gaming controller with a joystick and buttons, or make it more physically interactive with the user having to move around on the floor. Make it competitive or cooperative with a multi-player feature. Make it real-time with timed action and timed game-play. Add a light-panel for graphical output. Add some interesting inputs like a gyroscope, accelerometer, or video processing.



# Alarm Clock

Make a skill-testing alarm clock which ensures you are awake before turning off. Connect it to sensors to read if you lay back down in bed! Connect it to the web to pull up your Google calendar to figure out what time your alarm should be set for. Send an email to your boss if you are still in bed when you should be starting your shift. Make it bullet-proof and crash-proof so a baseball bat cannot turn it off (well, maybe not). ;)

# Lego Mindstorm

Connect the target to Lego Mindstorm robots and sensors to map a room, scan an object, fold a paper airplane, or anything else you can build!

# Logistics

- **Buying a Board Package**
  - \$?? Package includes:  
BeagleBone, USB audio,  
anti-static bag, electronics / components, box.
  - Hopeful we'll be able to sell kits next week
  - Afterwards able to buy from ENSC office in Burnaby
- **Academic Honesty**
  - Each student must have own board:  
sharing encourages *too much* cooperation.

# Summary

- Course is hands on:
  - Learning **skills**, not so much **theory**.
  - Expect to spend quite a bit of time figuring things out
- Have fun!

