50.046 Cloud Computing and IoT

Project briefing

2023 Term 7

Dr. Jiang Wenchao

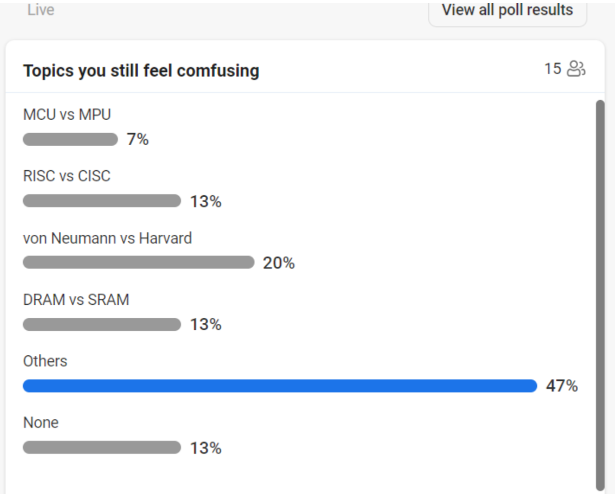
SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Outline

- Review and Q&A
- Project briefing
- Lab 1 briefing (by TA)

Review of previous classes

- Inside an embedded system
  - Processor, memory, I/O
- MCU vs. MPU
  - MCU: integrated processor, memory, I/O
  - MPU: separated
- Inside the processor
  - Instruction set architecture (ISA): RISC vs. CISC
  - Instruction and data
    - Share memory and buses (von Neumann)
    - Separated memory and buses (Harvard)



Topics you still feel comfusing 15

MCU vs MPU 7%

RISC vs CISC 13%

von Neumann vs Harvard 20%

DRAM vs SRAM 13%

Others 47%

None 13%

Review of previous classes

- Memory
  - RAM vs ROM
  - RAM: SRAM vs. DRAM vs. PSRAM vs. NVRAM
  - ROM: traditionally considered “read-only” but now with EEPROM and Flash
- I/O peripheral
  - A glance at UART, I2C, SPI
  - More to go next Monday

Q1 von Neumann vs Harvard

- Memory use in ESP32
- ROM (448KB):
  - First-stage boot loader, ROM libraries
- ESP32 is a Harvard architecture
  - Instruction RAM (by default, SRAM0, 192 KB)
    - The first 64KB can be used for cache
  - Data RAM (by default, SRAM1 + SRAM 2, 328 KB)
    - 8KB for ROM function
    - BT controller (when enabled) can take up 54KB
    - RAM1 and SRAM2 are usually used as a contiguous Data RAM address space

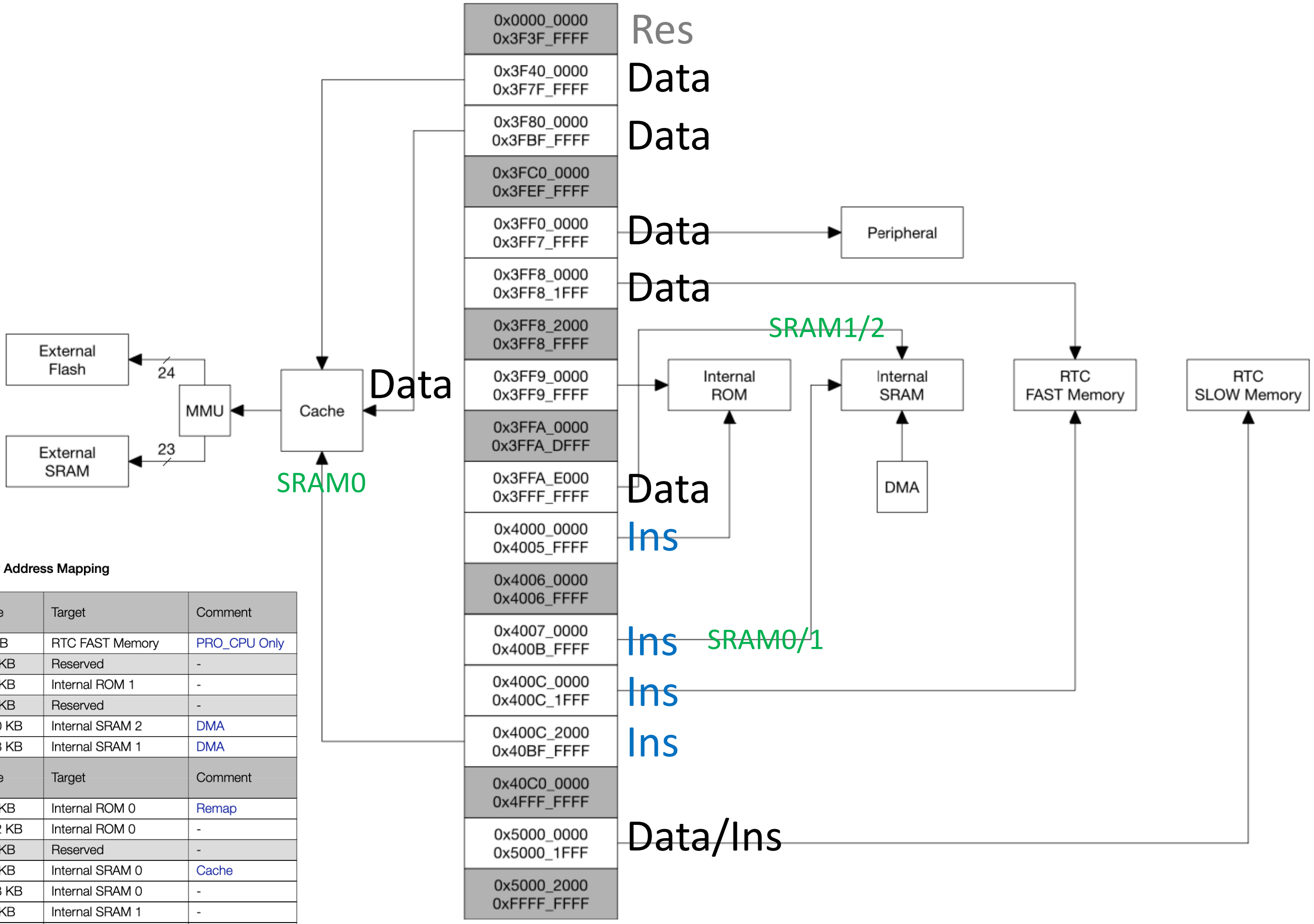
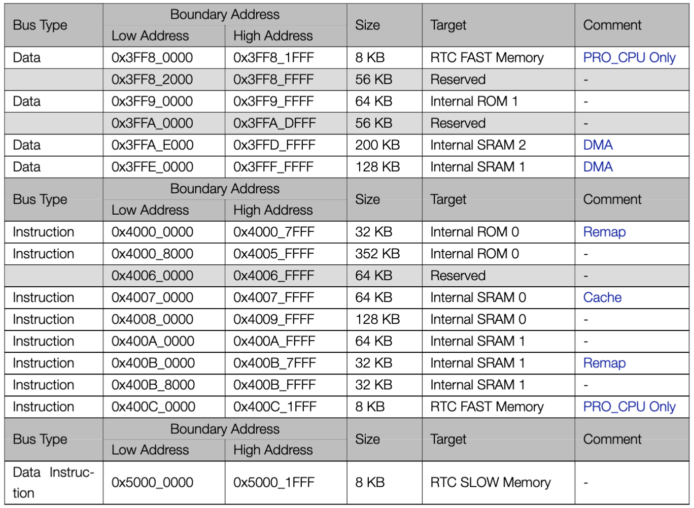


Figure 2: System Address Mapping

Table 4: Embedded Memory Address Mapping



Bus Type	Boundary Address		Size	Target	Comment
	Low Address	High Address			
Data	0x3FF8_0000	0x3FF8_1FFF	8 KB	RTC FAST Memory	PRO_CPU Only
	0x3FF8_2000	0x3FF8_FFFF	56 KB	Reserved	-
Data	0x3FF9_0000	0x3FF9_FFFF	64 KB	Internal ROM 1	-
	0x3FFA_0000	0x3FFA_DFFF	56 KB	Reserved	-
Data	0x3FFA_E000	0x3FFD_FFFF	200 KB	Internal SRAM 2	DMA
Data	0x3FFE_0000	0x3FFF_FFFF	128 KB	Internal SRAM 1	DMA
Bus Type	Boundary Address		Size	Target	Comment
	Low Address	High Address			
Instruction	0x4000_0000	0x4000_7FFF	32 KB	Internal ROM 0	Remap
Instruction	0x4000_8000	0x4005_FFFF	352 KB	Internal ROM 0	-
	0x4006_0000	0x4006_FFFF	64 KB	Reserved	-
Instruction	0x4007_0000	0x4007_FFFF	64 KB	Internal SRAM 0	Cache
Instruction	0x4008_0000	0x4009_FFFF	128 KB	Internal SRAM 0	-
Instruction	0x400A_0000	0x400A_FFFF	64 KB	Internal SRAM 1	-
Instruction	0x400B_0000	0x400B_7FFF	32 KB	Internal SRAM 1	Remap
Instruction	0x400B_8000	0x400B_FFFF	32 KB	Internal SRAM 1	-
Instruction	0x400C_0000	0x400C_1FFF	8 KB	RTC FAST Memory	PRO_CPU Only
Bus Type	Boundary Address		Size	Target	Comment
	Low Address	High Address			
Data Instruction	0x5000_0000	0x5000_1FFF	8 KB	RTC SLOW Memory	-

Source: ESP32 Technical Reference Manual

Q: Why Cache?

- ESP32:
  - Internal SRAM (520KB) is 32bit @ 240MHz max, so **960MByte/second**
  - Quad SPI connected PSRAM and Flash memory (they share the same QSPI) is 4-bit @ 80MHz, so **40MByte/second**

Question:

How to address speed difference?

Code executed from Flash

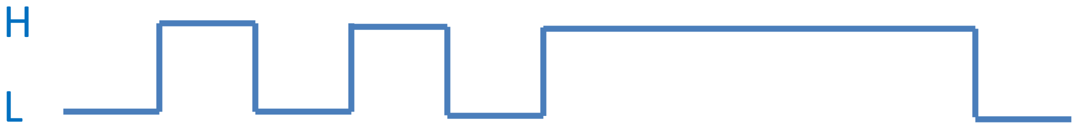
IROM (code executed from Flash)

- If a function is not explicitly placed into IRAM or RTC memory, it is placed into flash. The mechanism by which Flash MMU is used to allow code execution from flash is described in the Technical Reference Manual. ESP-IDF places the code which should be executed from flash starting from the beginning of 0x400D0000 — 0x40400000 region. Upon startup, second stage bootloader initializes Flash MMU to map the location in flash where code is located into the beginning of this region. Access to this region is transparently cached using two 32kB blocks in 0x40070000 — 0x40080000 range.
- *Comment: So the idea is internal memory may not be large enough to hold the code (say a large program). Then the flash will be used as 'external memory'. In order to make this process transparent to the CPU, i.e., not feeling a speed drop. The code will be cached in the 32kB blocks of SRAM0 (see address mapping in user mannual).*

General Notes About ESP-IDF Programming - ESP32 - — ESP-IDF Programming Guide release-v4.2 documentation (espressif.com)

Q: Why synchronization?

- One use case is the sync between Tx and Rx to **decode signal.**
- For example, suppose High is ‘1’ and Low is ‘0’, enough to decode the signal below?



- Solutions:
  - Agree on the speed
  - Use a clock line
  - Make signal self-clocking

Project briefing

Course project overview

- A group project that:
  - **Identify / formulate** a {societal | business} {problem | opportunity} that can be addressed by IoT
  - **Design** an end-to-end IoT-cloud solution (requirements, constraints, design space exploration, tech stack, design choice justification)
  - **Develop** a proof-of-concept prototype that demonstrates ***selected*** features

Form your group

- Each group consists of ~~3-4 students~~
  - Form your own group: we will adjust if a group is too big or too small
  - As more students enrolled the class, now we can have group of **5 students**.

- Register your group's info here: link

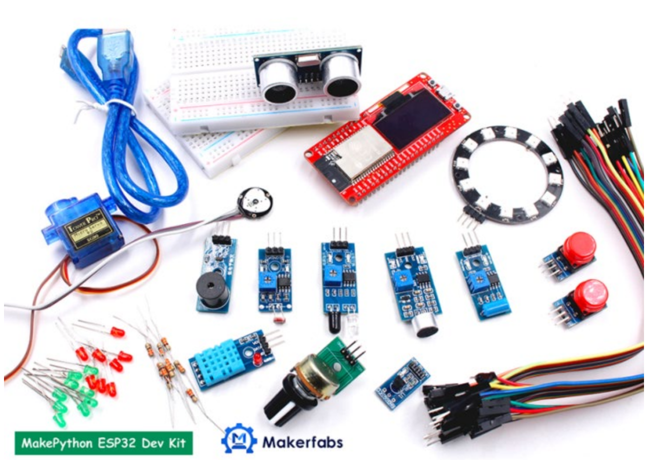
Approach us by next Thursday (21 Sept) if you cannot form a group of the right size

Project: logistics

- Each group will need to appoint one representative to collect IoT gadgets
- Keep good of the gadgets and return by
  - 11th Dec 2023 (Monday Week 14)

Devices we will provide

- ESP32
  - MakePython ESP32 Dev Kit
  - ESP32 CAM Arduino Kits



- Raspberry Pi
  - with touch screen

Some links to the gadget

- https://www.makerfabs.com/makepython-esp32-starter-kit.html

- https://www.wish.com/product/esp32-cam-arduino-kits-monitor-snapshot-face-detection-recognition-wifi-bluetooth-camera-module-with-128m-sd-card-usb-to-serial-cable-hc-sr501-sound-sensor-compatible-for-arduino-idetutorial-5e1c341229e786439a698b05

If you need additional gadgets

- We may have some additional devices (with limited stock) that you can loan from us
  - Talk to us
  - We will try our best to support, but you can simulate some component in your project, in case the actual devices are hard to get

Project budget

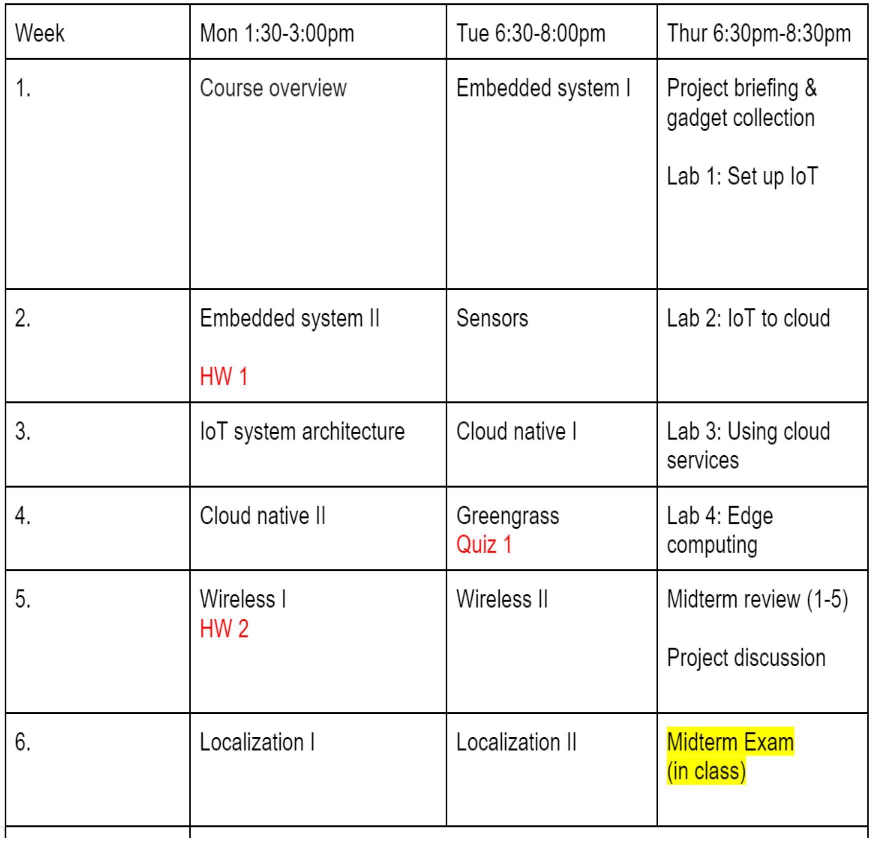
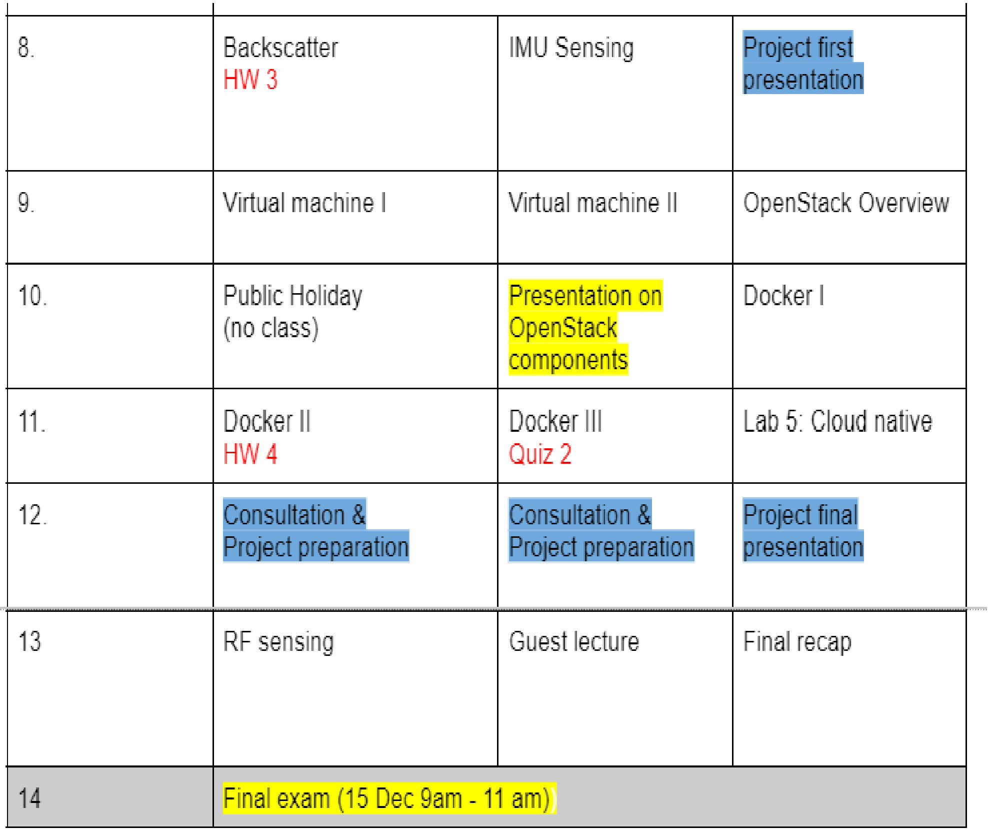
- Each group will also be given a budget of 100 USD / group for your project
  - Can be used for cloud platform charging, purchase of additional gadgets (if unavailable from school) required for the project
  - Please seek course instructors’ approval before purchase over SGD 20 (except for cloud service charge)
  - The allowance will be reimbursement-based. The official receipts / invoices are required
  - Please use the following code: **ISTD00CCI** when submitting your claims in Concur System

Project management

- Start now

- Each member expects to spend ~ 30 hours on the project (spend them in 2-3 hours / chunk)
  - So each team has ~ 100 hours to tackle some interesting problem together!
    - Work as a team. Start from a small, specific setting. Keep iterating. Keep your working journal.

Timeline

Week	Mon 1:30-3:00pm	Tue 6:30-8:00pm	Thur 6:30pm-8:30pm
1.	Course overview	Embedded system I	Project briefing & gadget collection Lab 1: Set up IoT
2.	Embedded system II HW 1	Sensors	Lab 2: IoT to cloud
3.	IoT system architecture	Cloud native I	Lab 3: Using cloud services
4.	Cloud native II	Greengrass Quiz 1	Lab 4: Edge computing
5.	Wireless I HW 2	Wireless II	Midterm review (1-5) Project discussion
6.	Localization I	Localization II	Midterm Exam (in class)
8.	Backscatter HW 3	IMU Sensing	Project first presentation
9.	Virtual machine I	Virtual machine II	OpenStack Overview
10.	Public Holiday (no class)	Presentation on OpenStack components	Docker I
11.	Docker II HW 4	Docker III Quiz 2	Lab 5: Cloud native
12.	Consultation & Project preparation	Consultation & Project preparation	Project final presentation
13	RF sensing	Guest lecture	Final recap
14	Final exam (15 Dec 9am - 11 am)		

Rubrics (at a scale of 100 pt, 20% of total grade)

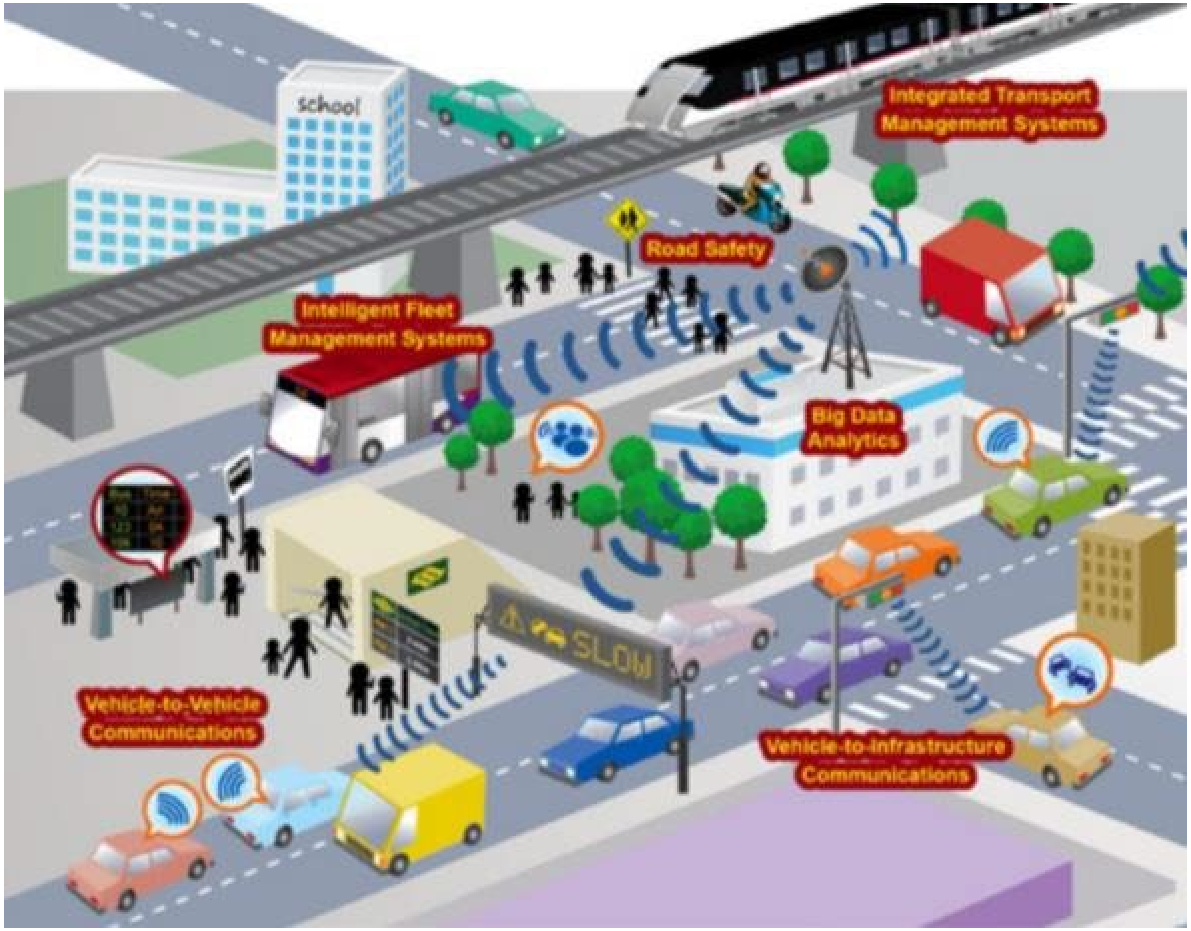
Category	Criteria	Grade	Remark
Design (25pt)	Problem statement & justification (ROI/cost benefit analysis)	5	
	Requirement, constraints, design space, state-of-the-art	10	
	System design and justification	10	
Build (40pt)	Prototype of an end-to-end demo system (some components can be simulated)	30	
	Evaluation (e.g., in terms of performance / overhead / usability / security) of key component(s) via experiment	10	
Show (35pt)	First presentation (week 8)	5	
	Final presentation + demo (week 12)	15	
	Project report / medium post	15	
Bonus	Best project (voted by your peers)	10	
Total		100 + 10	

Some questions for you to consider

- What will be the benefit of my project? Can quantify in $ or social impact?
- What features / use cases to support?
- What will be the cost? Quantify in $ (CAPEX, OPEX, as a function of the system size, as a result of my design)
- Thing: What to sense? What (not) to sense? How powerful should be the MCU? What should / should not be included in my device?
- Connectivity: What / how / when to send? What protocol stack to use?
- Data: Any external data sources besides IoT data? How to design your data processing / storage pipeline?
- Additional key considerations? For example, real-time requirement / security / privacy / mobility / reliability / harsh environment? How my design address them?
- What kind of analytics? Where to put it? How to optimize it?
- How to deploy / bootstrap your system? How to maintain / upgrade your system? How to retire your system?
- How to interface with other services? What APIs you may want to provide?

Example I

Example II


school
Integrated Transport Management Systems
Road Safety
Intelligent Fleet Management Systems
Big Data Analytics
SLOW
Vehicle-to-Vehicle Communications
Vehicle-to-Infrastructure Communications

Presentations (assuming 13 teams)

- First presentation
  - 2 Nov (Thursday, Week 8)
  - 7 mins presentation + 2 mins Q&A
  - Focus on design, justification, and your plan
  - Mandatory: a story on https://medium.com/

- Final presentation (with demo)
  - 30 Nov(Thursday, Week 12)
  - 3 mins presentation + 6 mins demo
  - Focus on what you have built & key features
  - Mandatory: an updated story on https://medium.com/

Final submission

- Deadline: 8 Dec (Friday, week 13) 23:59
- Group submission:
  - A demo system
    - Your code, with a README / quick start guide
    - Keep your setup in both the IoT devices (when you return the devices to us) and the cloud
  - A technical report (you can reuse the content from your medium post, but add necessary technical details)
  - The link to your medium post
- Individual submission:
  - a 2-3 page individual report
- Return of gadget: 11 Dec (Monday, week 14)

Individual report – Part I (example)

- My main contributions to the project include:
  - Propose the design of …
  - Built the …. component, together with Alice
  - ……
- My working logs

Index	Activities	Week	# of hours	Notes
1	Group brainstorming	4	2	Presented to the team on the idea of ….
2	Setup … tool	5	3	Solved the issues ….
	…			
n	Write up report Section 3	10	4	Together with ….
	Total hours		30	

Individual report – Part II (example)

- My reflection ---- what we did well, what we could do better
- My peer-review

Teammate	My rating for him/her (0-5)	Notes
Alice	5	Lead the team in implementing the prototype
Bob	3	Absent in most of our group meetings
…		

We will use the peer reviews to adjust the individual member's grade (if needed)

Demo day video from last year

Time for you!

- Do you already have some potential areas / topics that you want to work on?
  - As of 14 Sept morning, 13 complete groups have formed
- Does your capstone project have a cloud computing & IoT element?
- Are you looking for peers to join your group?
- Are you seeking a group to join?

Any questions?