E210 Engineering Cyber-Physical Systems (Spring 2021)

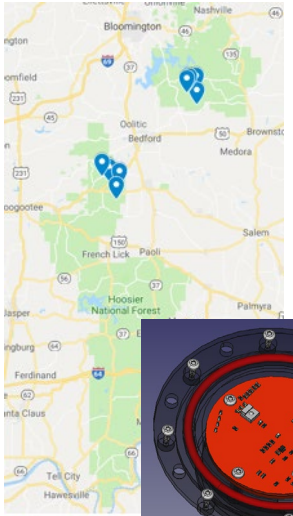
CPS Introduction

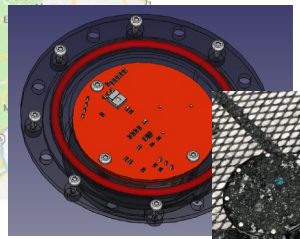
Bryce Himebaugh

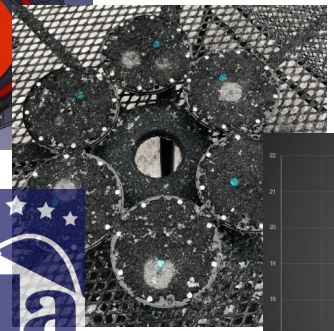
Introductions

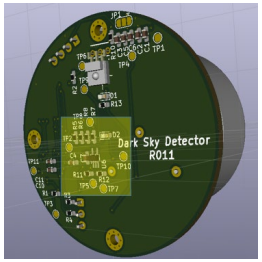
Bryce Himebaugh

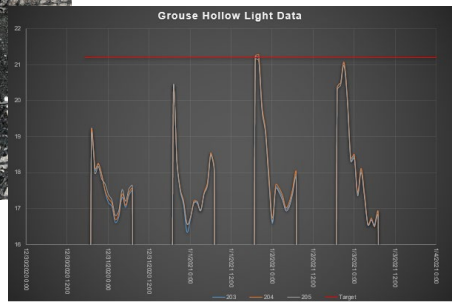
1. Clinical Professor
2. Current Classes
   - Fall E101, E321
   - Spring E210, E314, E491











Andrew Lukefahr

- Classes:
  - E210, E315
- Research
  - FPGA bitstream tamper prevention
  - FPGA piracy detection
  - Security of non-volatile FPGA architectures
  - https://sailin.luddy.indiana.edu/

Syllabus

Syllabus Review

1. https://engr210.github.io/syllabus.html

Schedule Review

1. https://engr210.github.io/

Weekly Focus	Reading	Monday	Wed	Lab
CPS Intro/UART		**1/10:** CPS Introduction	**1/12:** Pi Intro/UART Bus	**Project 0 Raspberry PI Setup**
GPIO/Scope		**1/17:** MLK Day	**1/19:** Electronics/GPIO/LED	**Project 1 UART Controlled LED**
SPI Bus		**1/24:** I2C Bus Overview	**1/26:** Pressure Sensor	**Project 2 I2C Pressure/Temperature Sensor**
I2C Bus		**1/31:** SPI Bus Overview	**2/2:** SPI Flash Memory	**Project 3 Project 2 SPI Flash Memory**
Networking		**2/7:** Networking Overview	**2/9:** Flask	**Project 4 Flask Web Server**
Web Server		**2/14:** Redis/matplotlib	**2/16:** CPS Wrapup	**P5 Demultiplexer**
Evaluation		**2/21:** Exam 1	**2/23:** CE Intro/ Logic	**P6 ALU**
CPE Intro/Logic		**2/28:** Truth Tables	**3/2:** Verilog Basics	**P6 ALU**
Combinational Logic		**3/7:** Testbenches	**3/9:** Math I	**P7 Saturating Counter**
		3/14: Spring Break	**3/16:** Spring Break	
Sequential Logic		**3/14:** Math II	**3/16:** Latches	**P7 Saturating Counter**
FSMs		**3/21:** Flops	**3/23:** Sequential Logic	**P8 Elevator Controller**
FSMs		**3/28:** FSMs I	**3/31:** FSMs II	**P8 Elevator Controller**
FSM/Timing		**4/4:** FSMs III	**4/6:** Timing	**P9 SPI Interface**
SPI Implementation		**4/11:** SPI I	**4/13:** SPI II	**P9 SPI Elev. Controller**
Memory Interfaces		**4/18:** MMIO I	**4/20:** MMIO II	**P10 Flask/SPI Elev. Controller**
Review		**4/25:** Overflow	**4/27:** Review	
		5/06: Final Exam @ 12.40-2:40pm		

What is CPS ?

National Science Foundation

Cyber-physical systems (CPS) are engineered systems that are built from, and depend upon, the seamless integration of computation and physical components.

(https://www.nsf.gov/funding/pgm_summ.jsp?pims_id=503286)

“Cyber-physical systems are physical, biological, and engineered systems whose operations are **integrated**, **monitored**, and/or controlled by a **computational core**. Components are **networked** at every scale. Computing is **“deeply embedded”** into every physical component, possibly even into materials. The computational core is an embedded system, usually demands real-time response, and is most often distributed. The behavior of a cyber-physical system is a fully-integrated hybridization of computational (logical) and physical action.” (Gill, 2008)

Helen Gill: Program Director for the NSF Directorate for Computer and Information Science and Engineering (CISE)

1. Cyber – computation, communication, and control that are discrete, logical, and switched

2. Physical – natural and human-made systems governed by the laws of physics and operating in continuous time

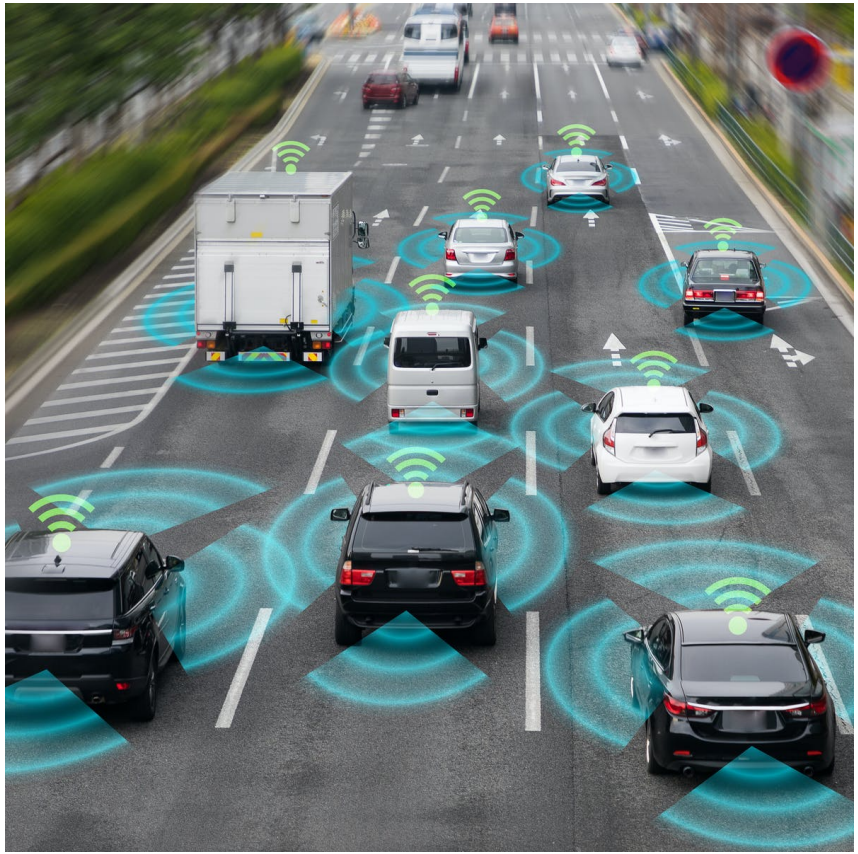
3. Cyber-Physical Systems – systems in which the cyber and physical systems are tightly integrated at all scales and levels

https://labs.ece.uw.edu/nsl/aar-cps/Gill_HCSS_Transportation_Cyber-Physical_Systems_2008.pdf

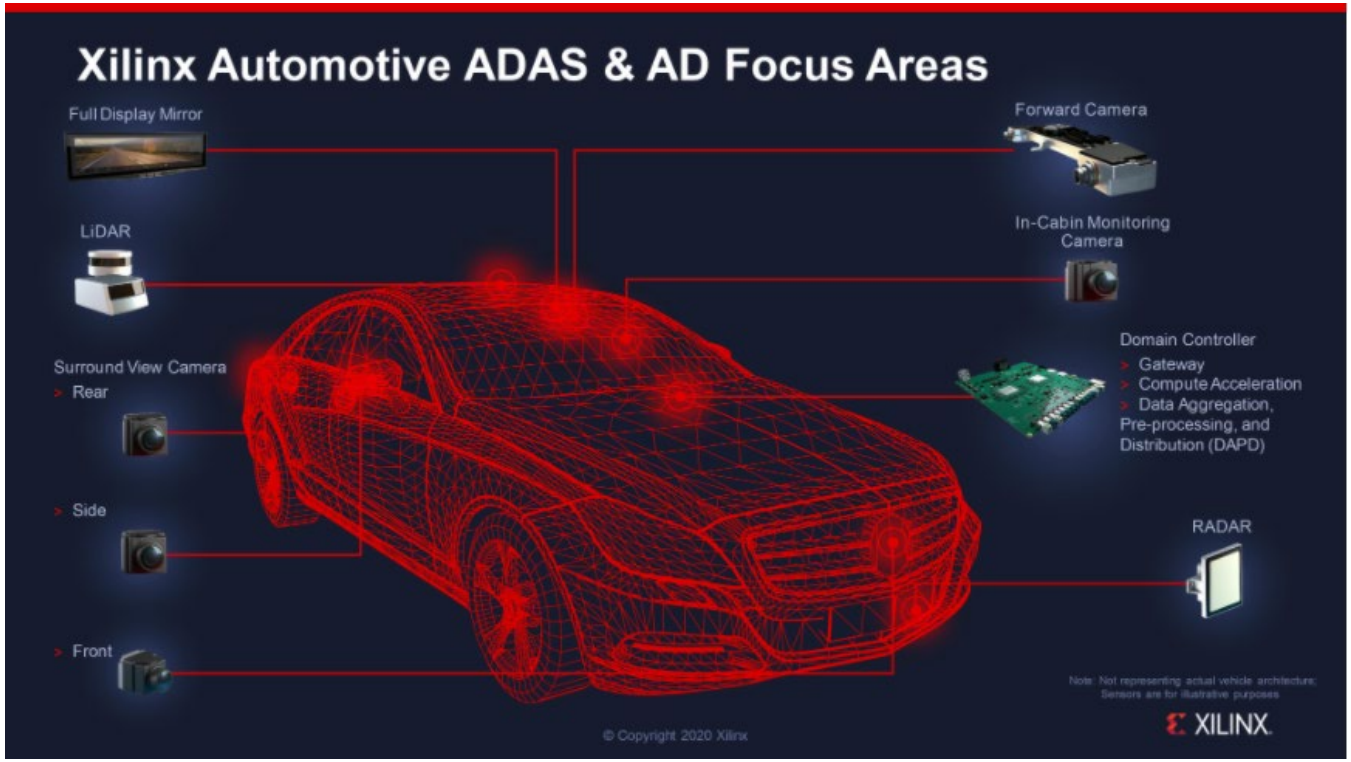
In CPS, embedded computers and networks monitor and control the physical processes, usually with feedback loops where physical processes affect computations and vice versa. The design of such systems, therefore, requires understanding the joint dynamics of computers, software, networks, and physical processes. It is this study of joint dynamics that sets this discipline apart. (Lee, Seshia 2017)

https://ptolemy.berkeley.edu/books/leeseshia/

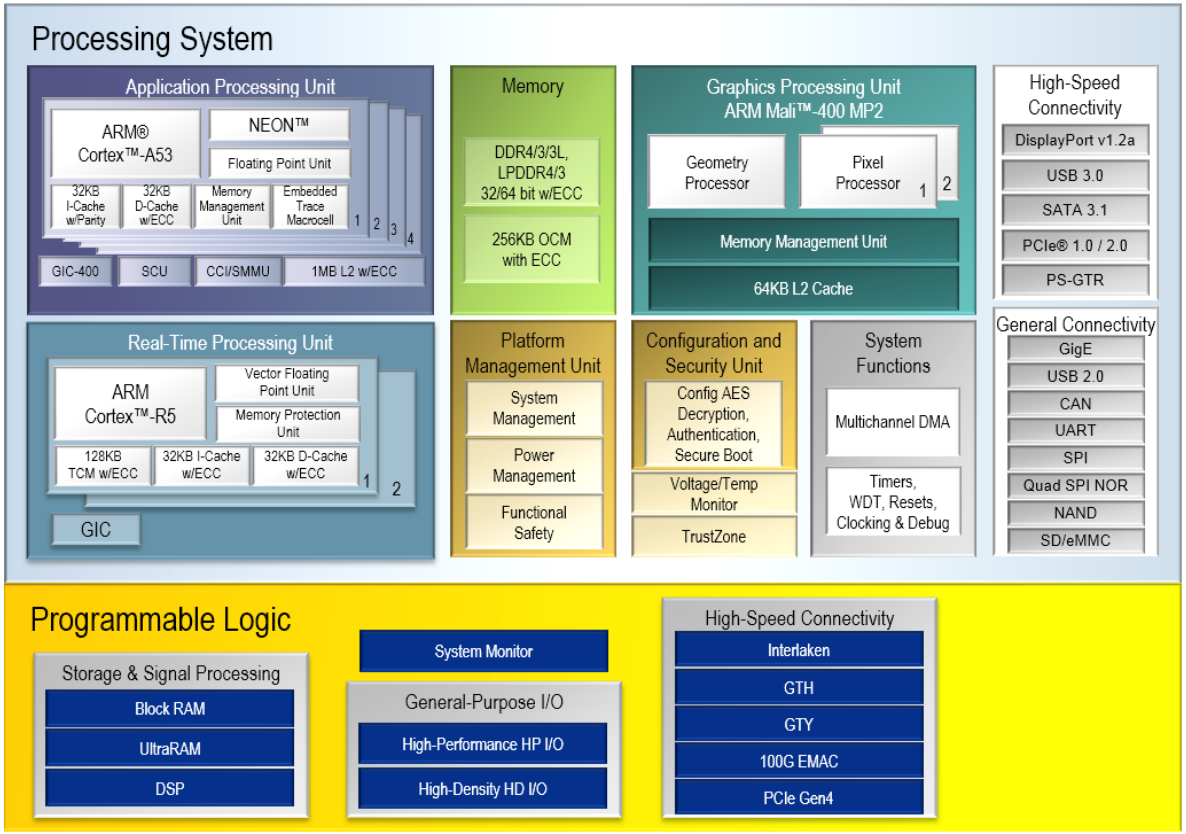
Examples



https://theconversation.com/what-if-autonomous-vehicles-actually-make-us-more-dependent-on-cars-98498

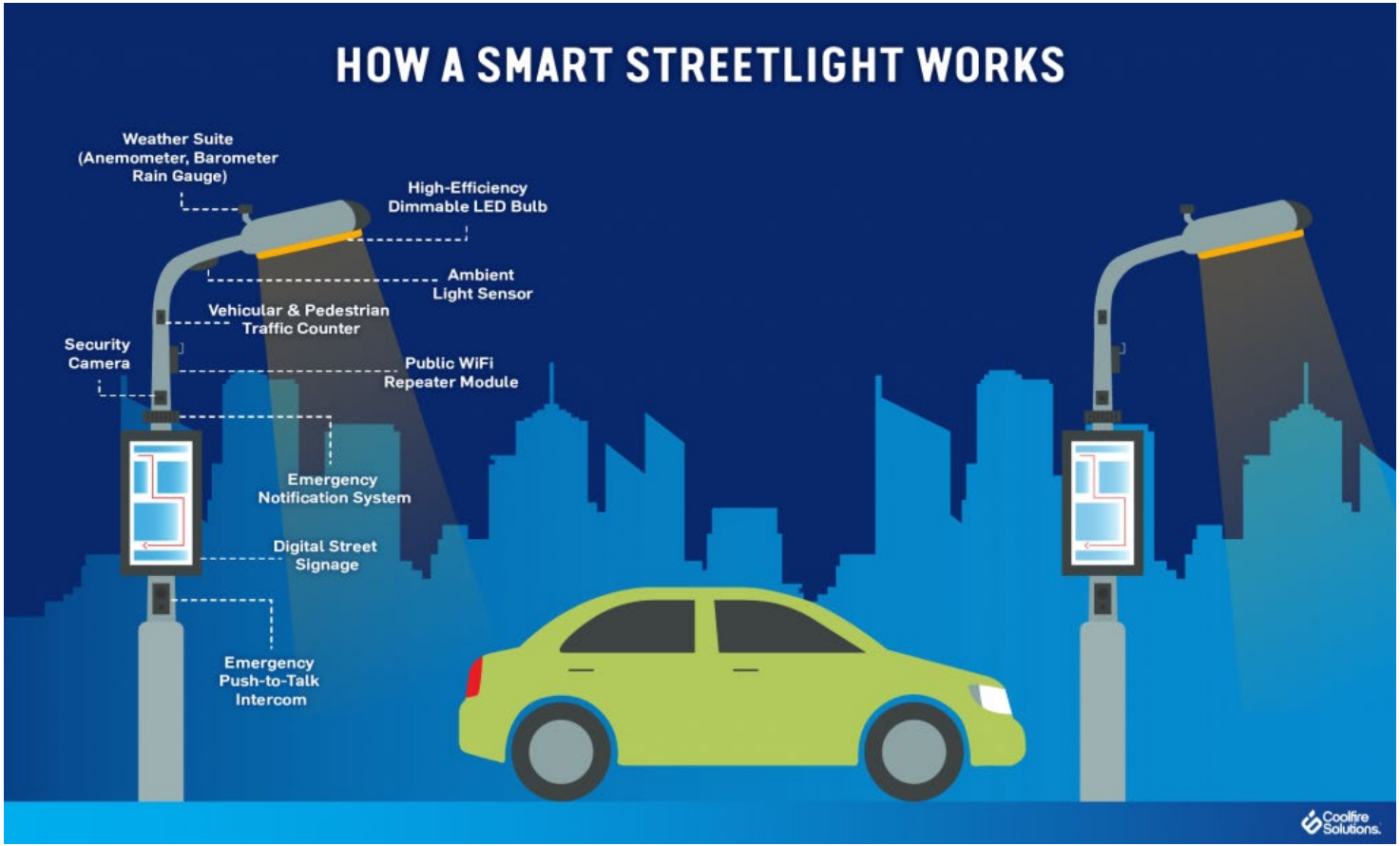


https://www.forbes.com/sites/marcochiappetta/2020/08/19/subaru-taps-xilinx-for-its-new-eyesight-vision-based-advanced-driver-assistance-system/?sh=28889b7471f5



https://www.rs-online.com/designspark/accelerate-opencl-applications-with-digilent-genesys-zu-3eg-zynq-ultrascale-mpsoc-platform

https://medium.com/@SunflowerLab/how-can-iot-make-cities-smart-e17290fb6838



https://www.coolfiresolutions.com/blog/smart-street-lights/

https://orthofeed.com/2018/05/29/the-operating-room-of-the-future/

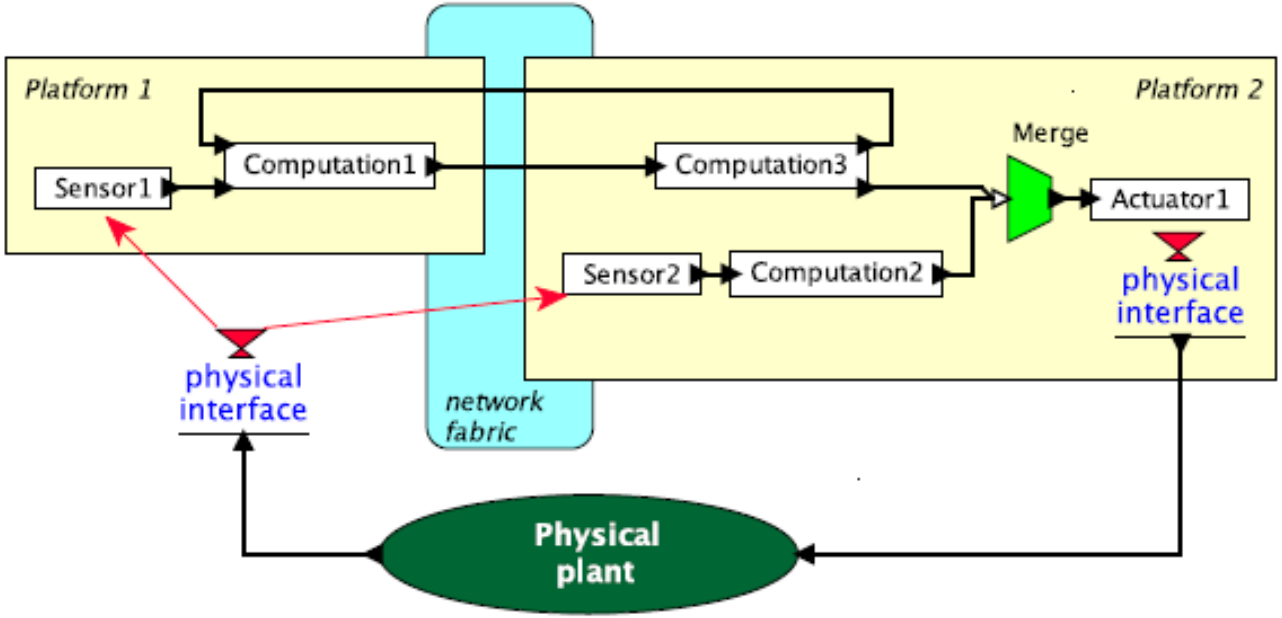


Figure 1.1: Example structure of a cyber-physical system.

Lee, Seshia 2011

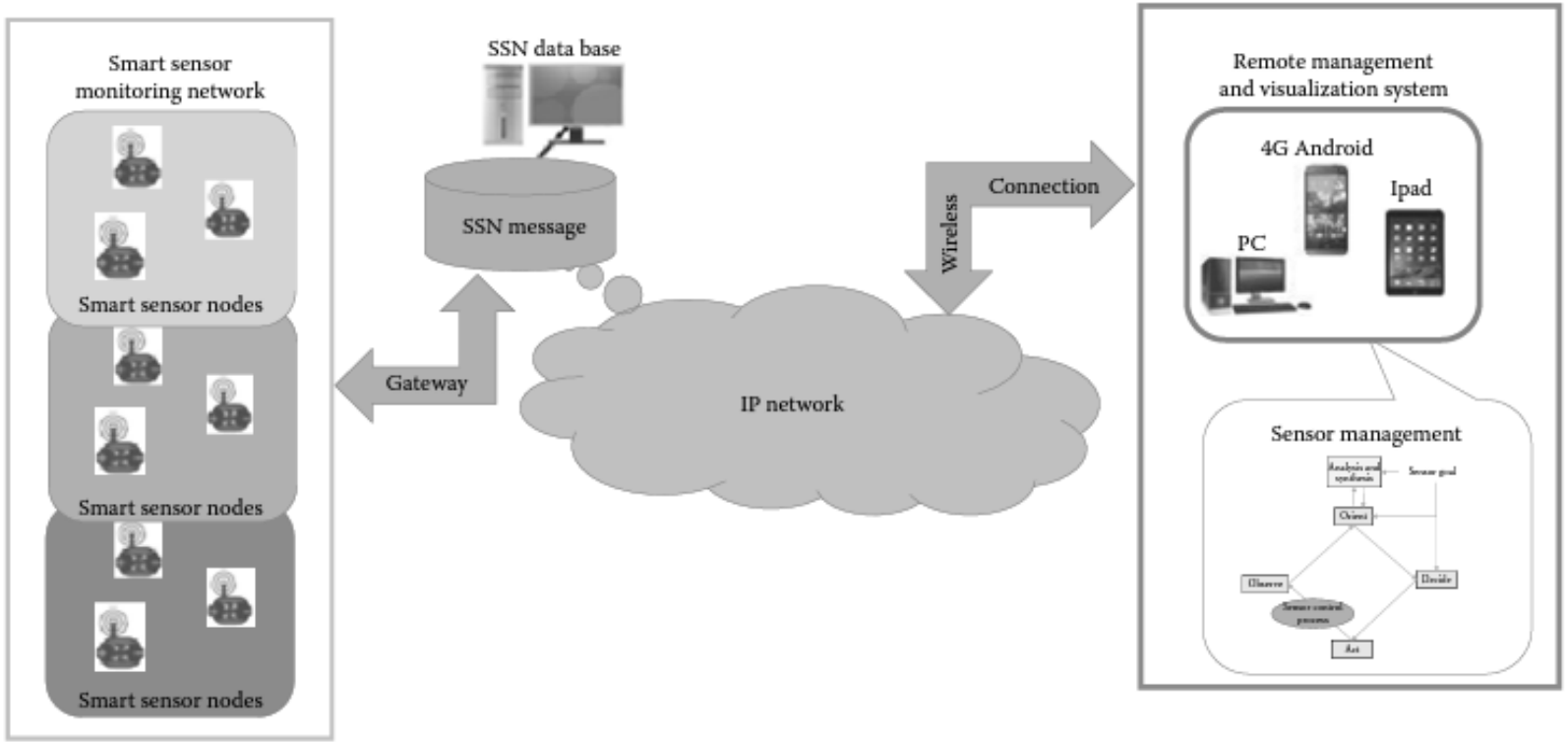
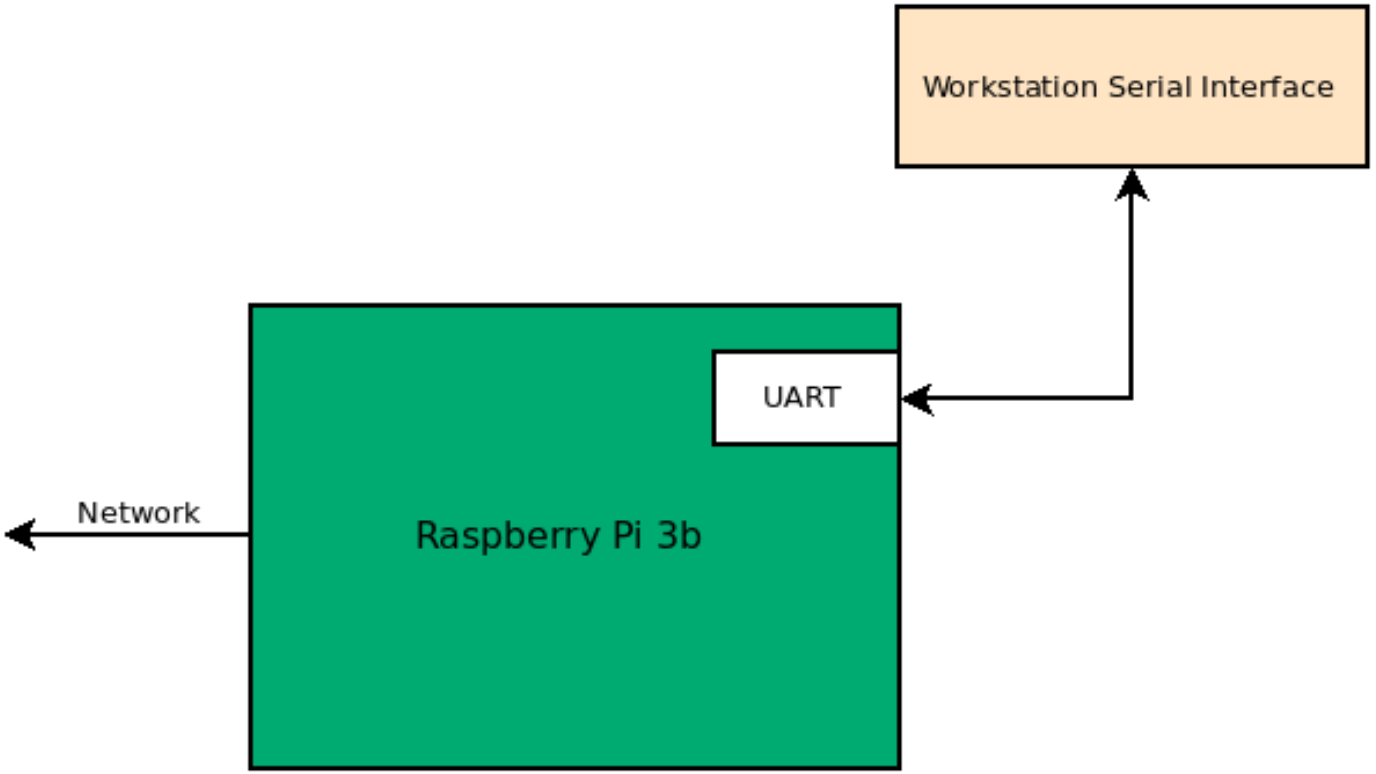


FIGURE 1.4
Typical SSN system architecture.

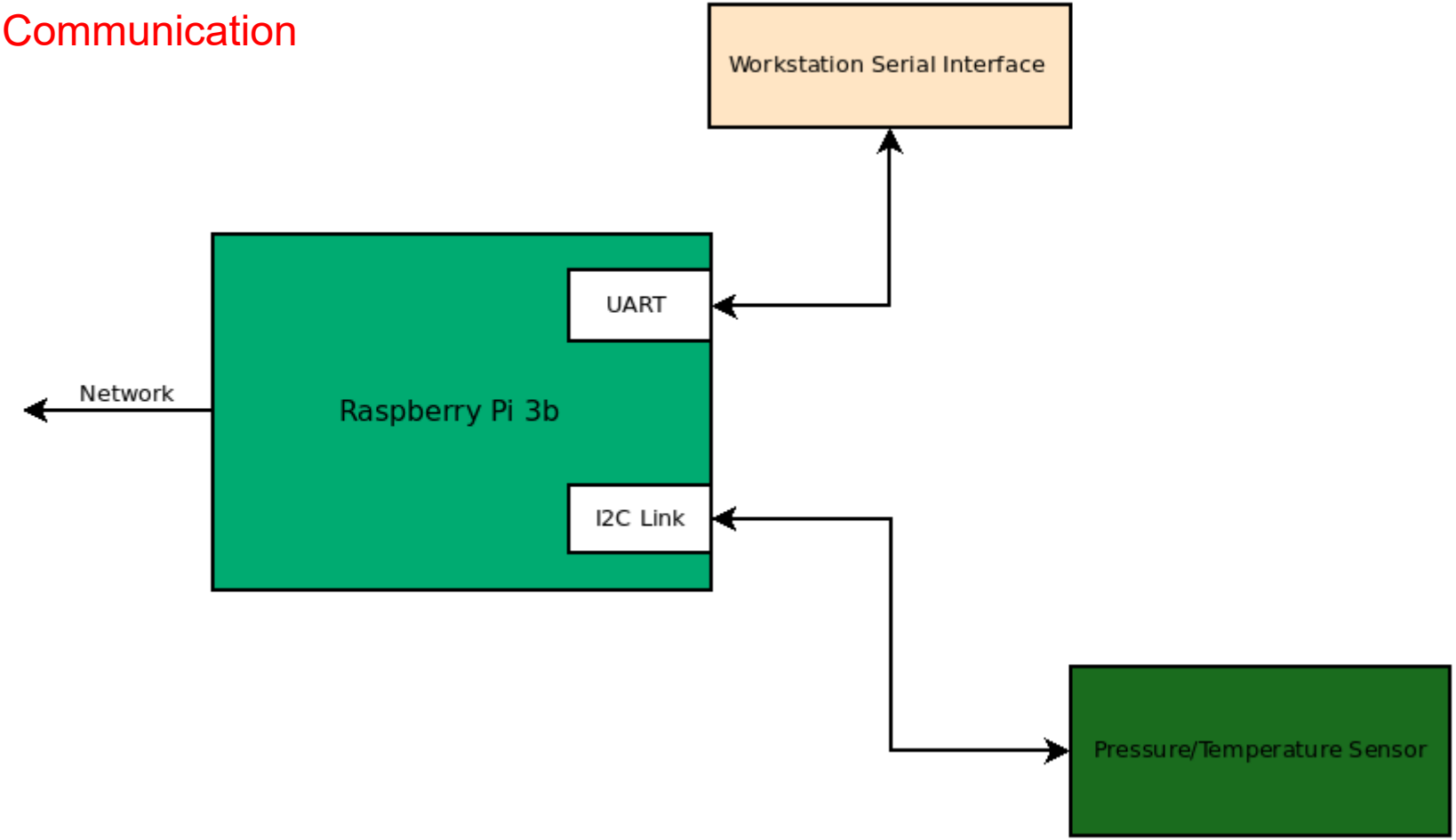
Siddesh, Deka, Srinivasa, Patnaik 2016

CPS Projects

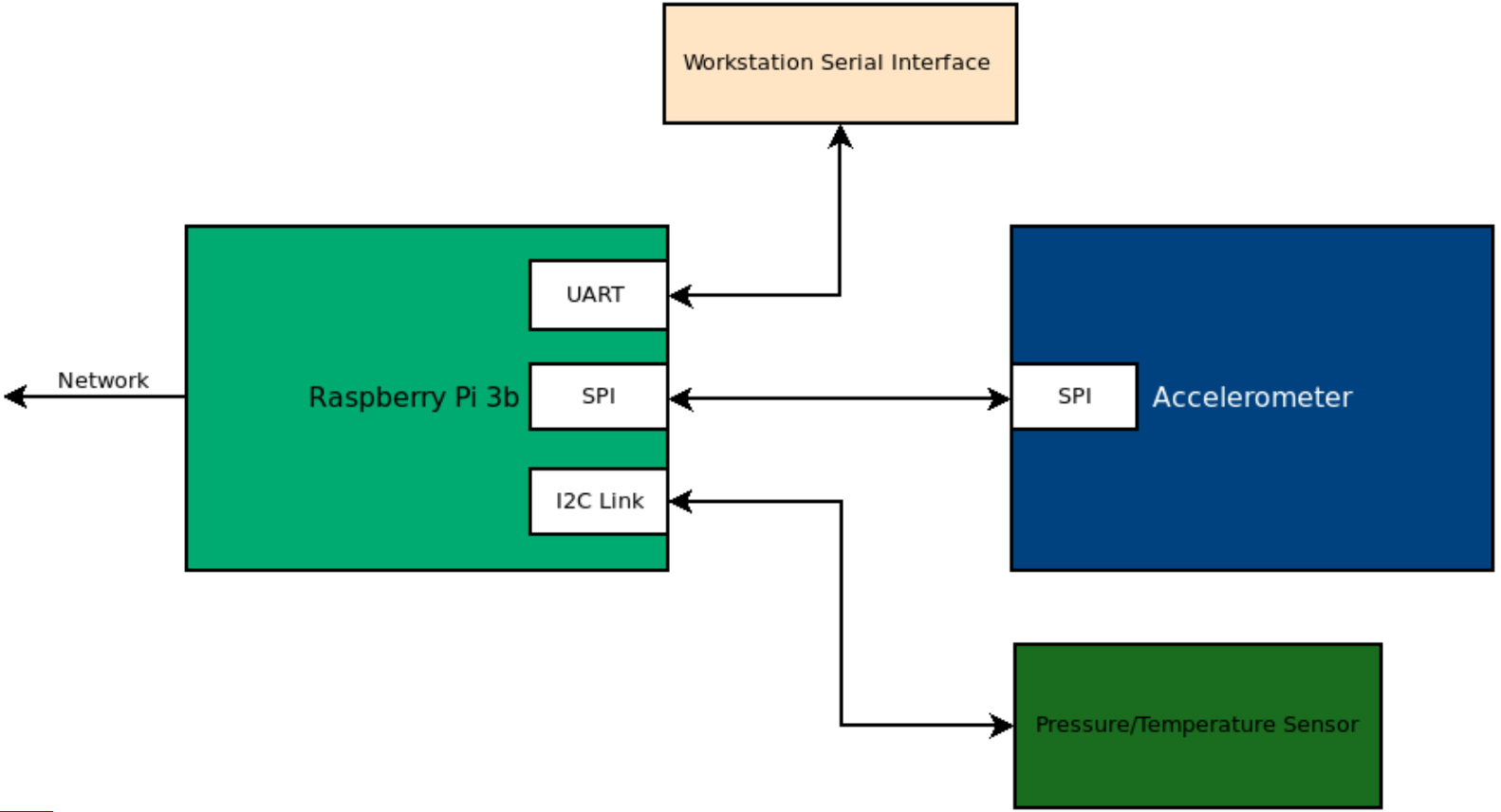
Raspberry Pi/UART



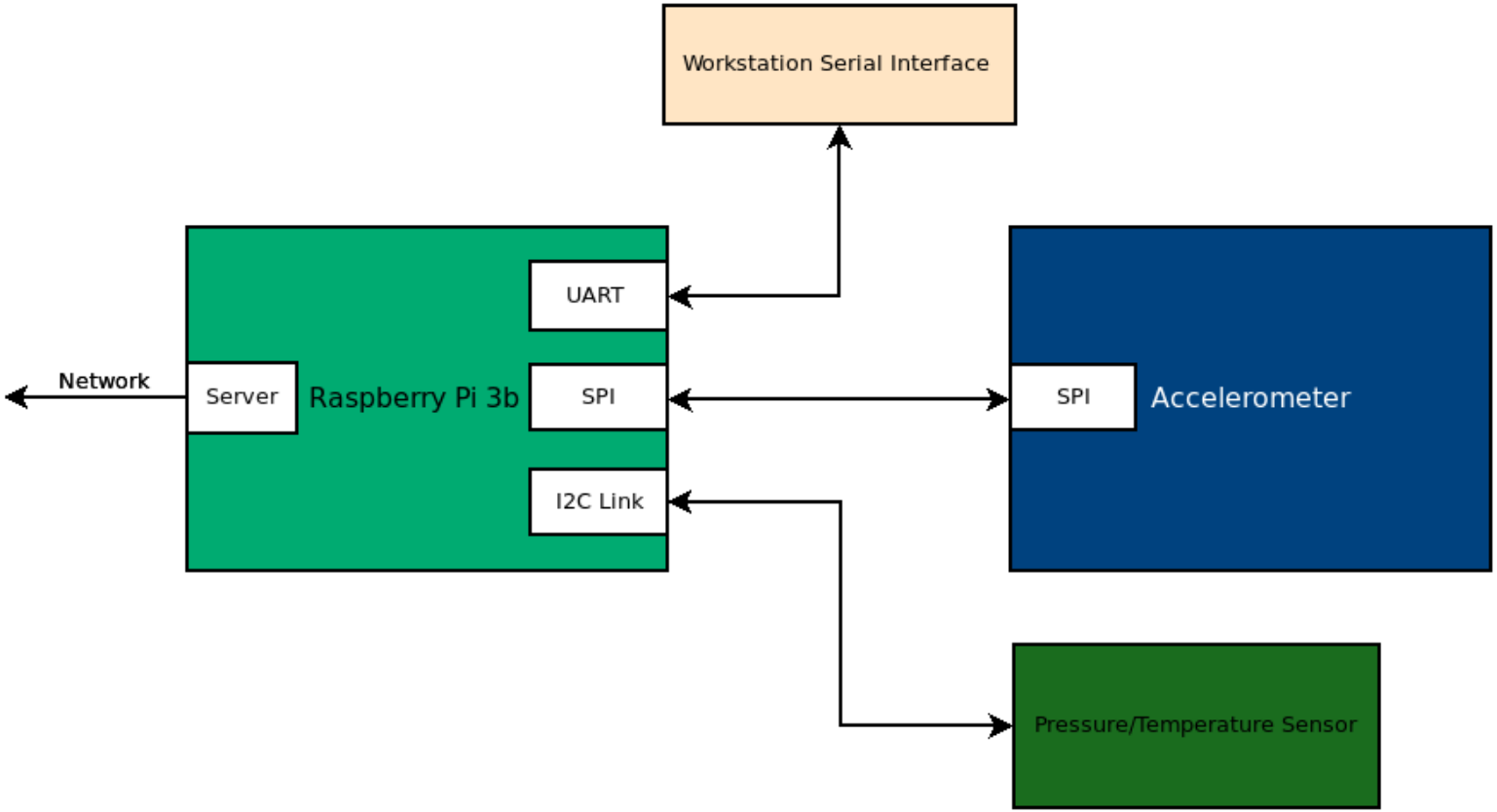
I2C Communication



System Verilog SPI Interface



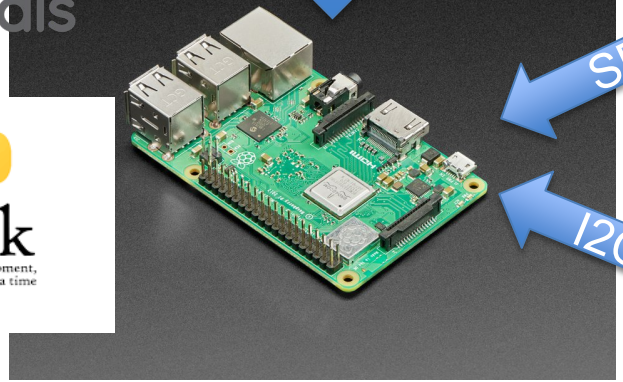
Python Network Interface



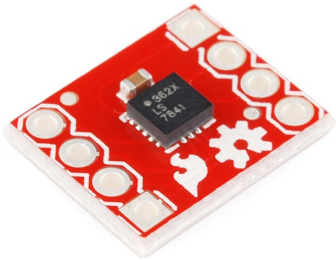
UART

redis

TCP/IP

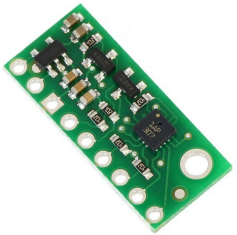


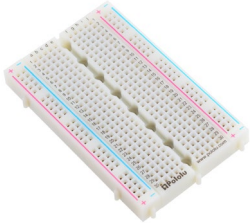
SPI



Flask
web development,
one drop at a time

I2C





Reading References

Reading References:

1. Introduction to Embedded Systems - A Cyber-Physical Systems Approach
2. Pro Git Book
3. Sparkfun I2C Tutorial
4. Python Classes
5. ST LPS331 Pressure Sensor
6. Sparkfun SPI Tutorial
7. Cyber-Physical Systems - a Computational Perspective

https://engr210.github.io/

Learning Objectives

A successful student will be able to:

- Create a networked sensor system.
  - Develop System Verilog in the context of a larger system.
  - Apply communication networks UART, I2C, and SPI.
  - Develop python programs as part of a CPS system.
  - Demonstrate the ability to utilize a version control system.
  - Perform effectively in a small design team.