E210 Engineering Cyber-Physical Systems (Spring 2021)

CPS Introduction

Bryce Himebaugh

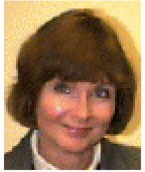
What is CPS ?

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

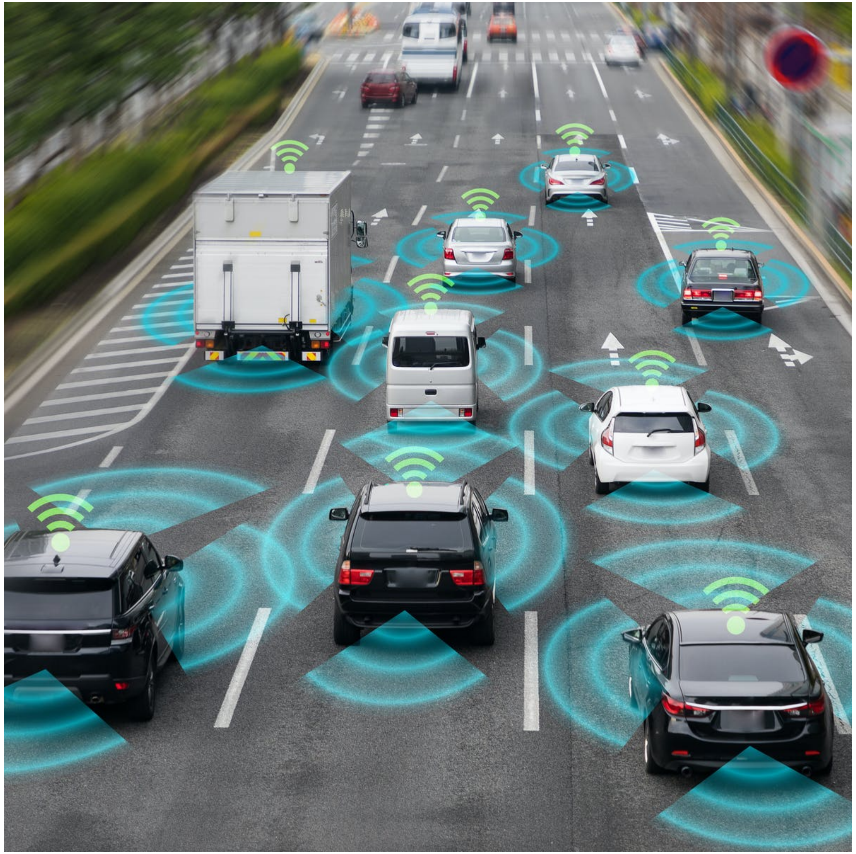
3. Cyber-Physical Systems – systems in which the cyber and physical systems are tightly integrated at all scales and levels

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

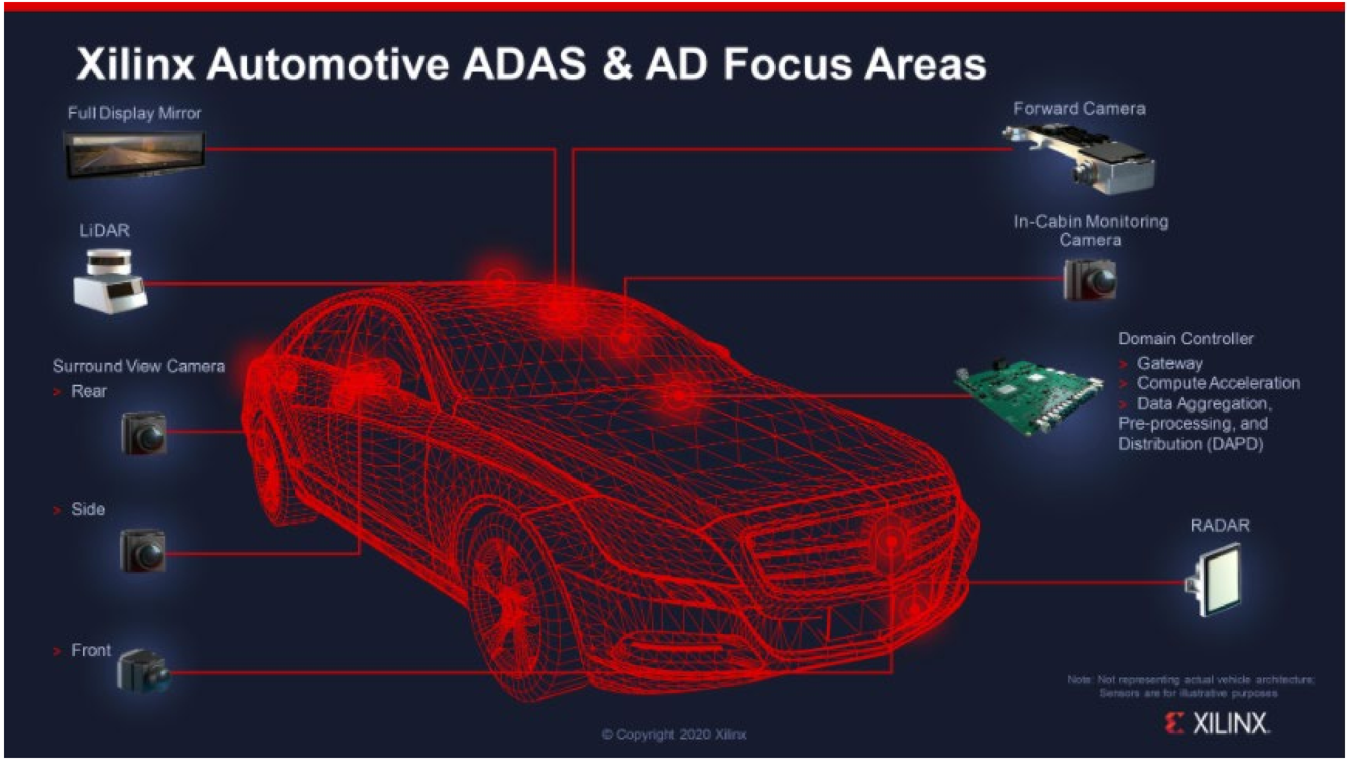
In CPS, embedded computers and networks monitor and control the physical processes, usually with feedback loops where physical processes affect computations and vice versa. The design of such systems, therefore, requires understanding the joint dynamics of computers, software, networks, and physical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

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

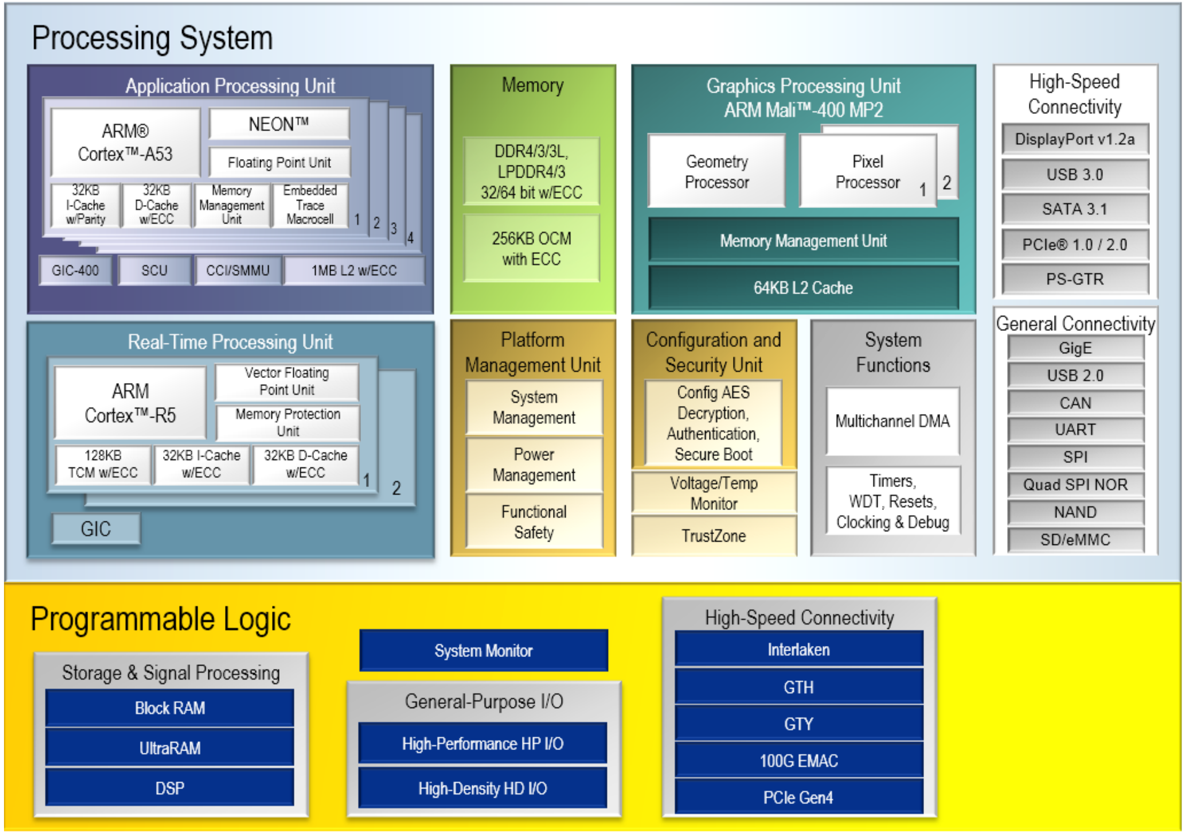
Examples



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

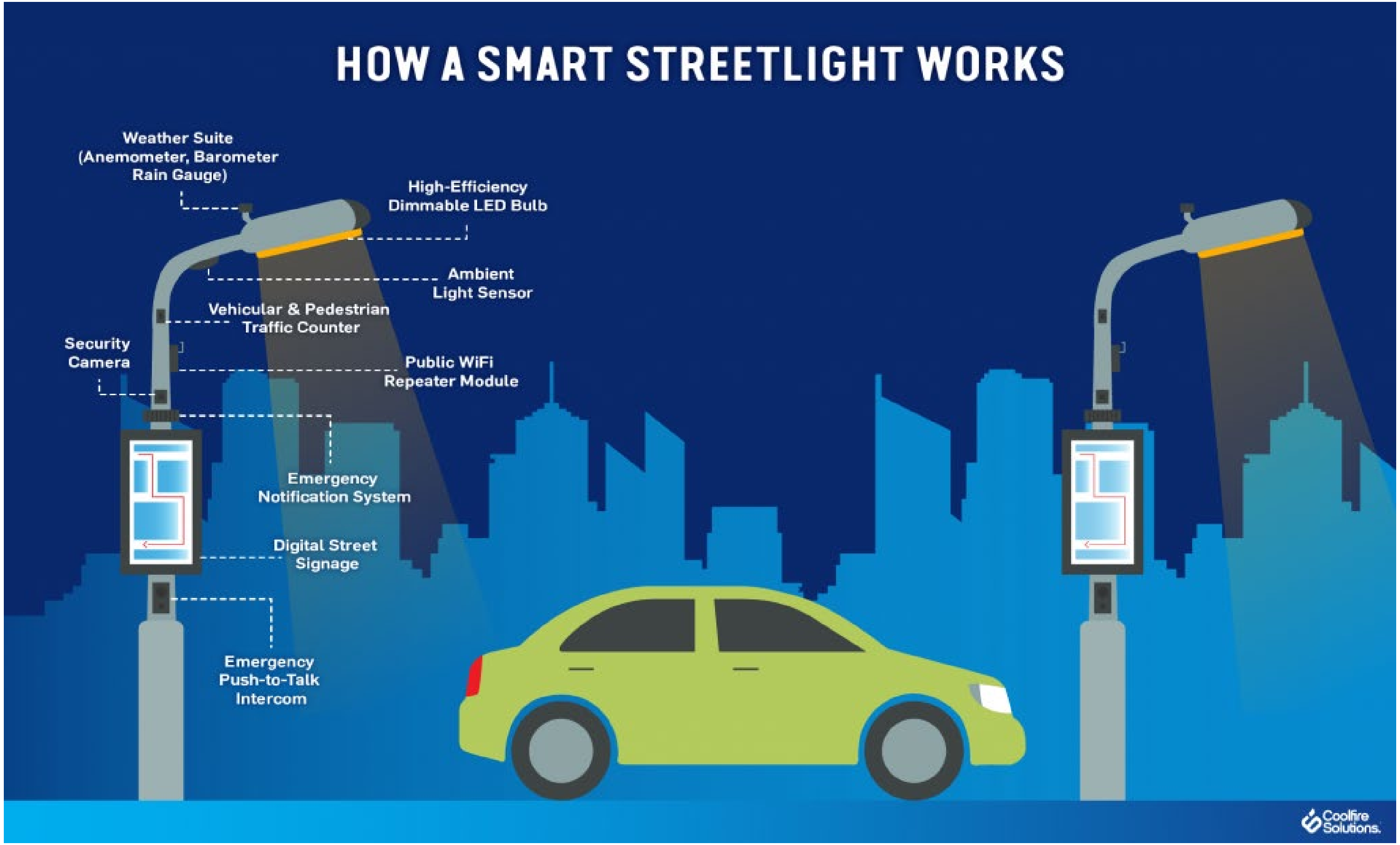


https://www.forbes.com/sites/marcochiappetta/2020/08/19/subaru-taps-xilinx-for-its-new-eyesight-vision-based-advanced-driver-assistance-system/?sh=28889b7471f5



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/

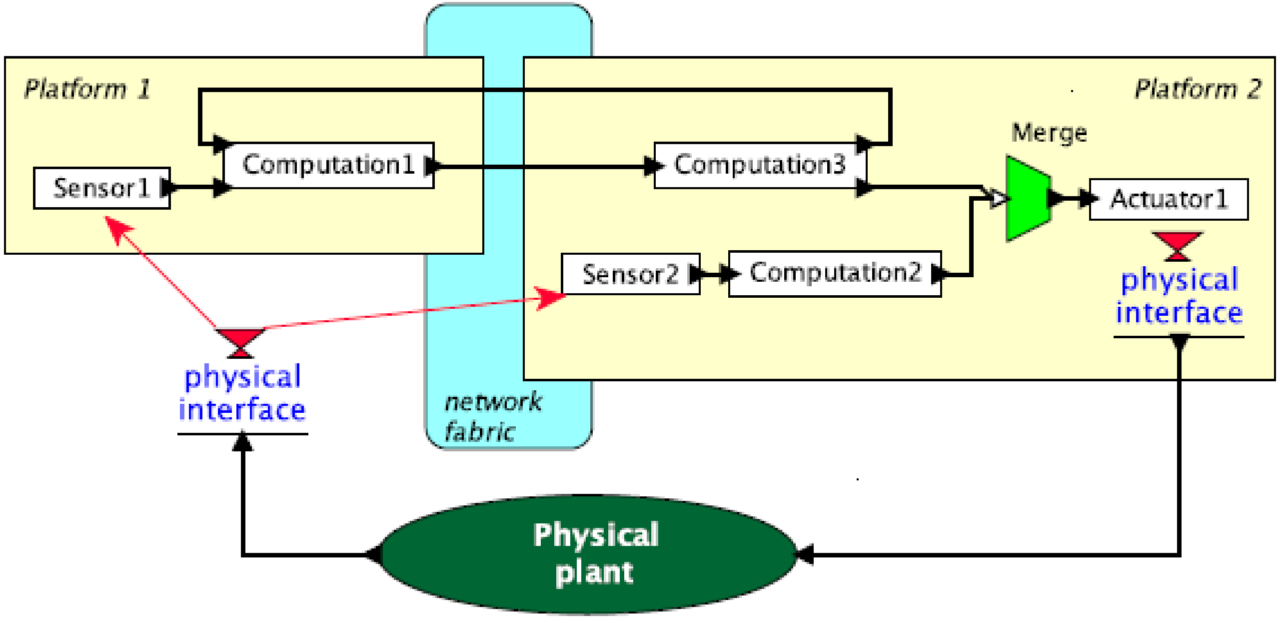


Figure 1.1: Example structure of a cyber-physical system.

Lee, Seshia 2011

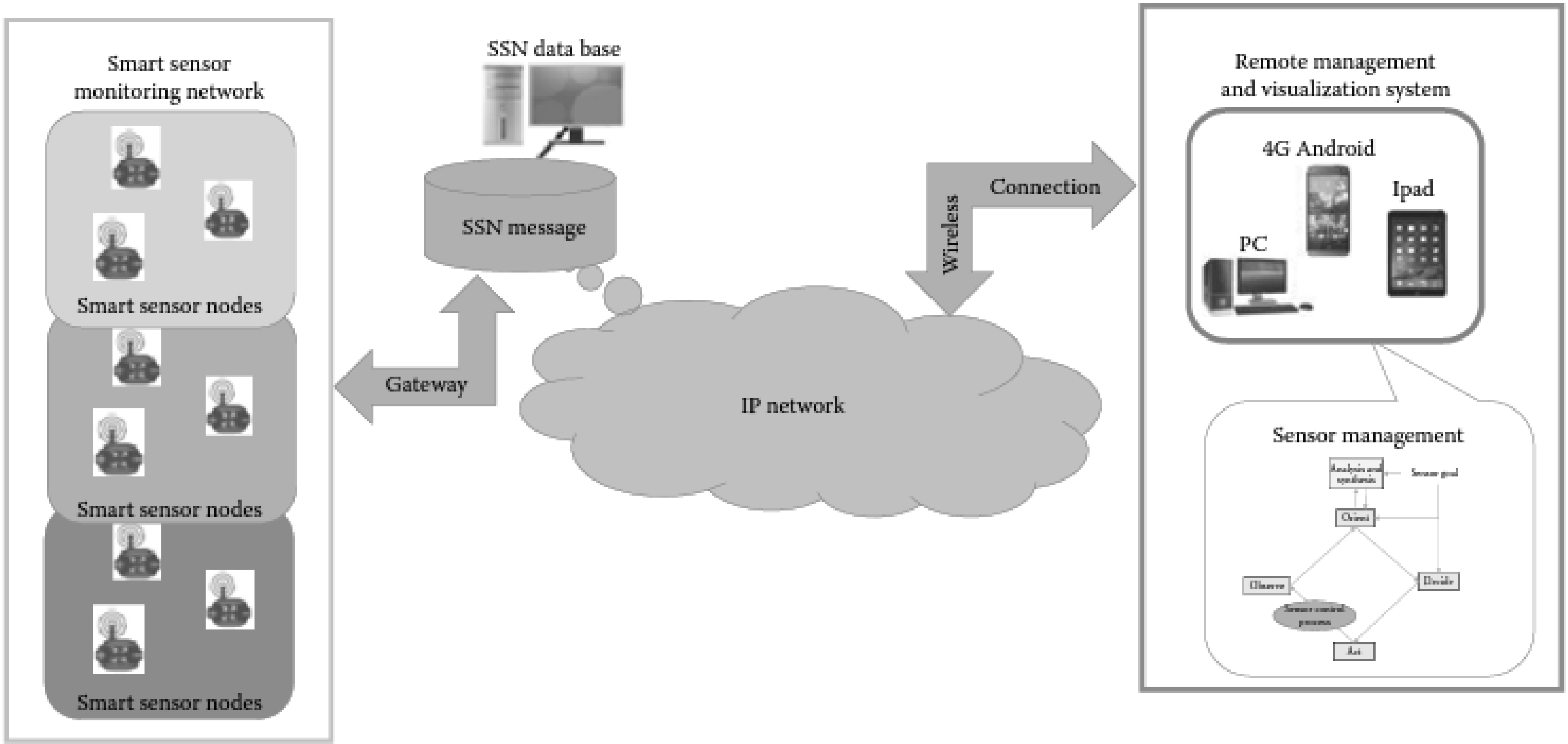
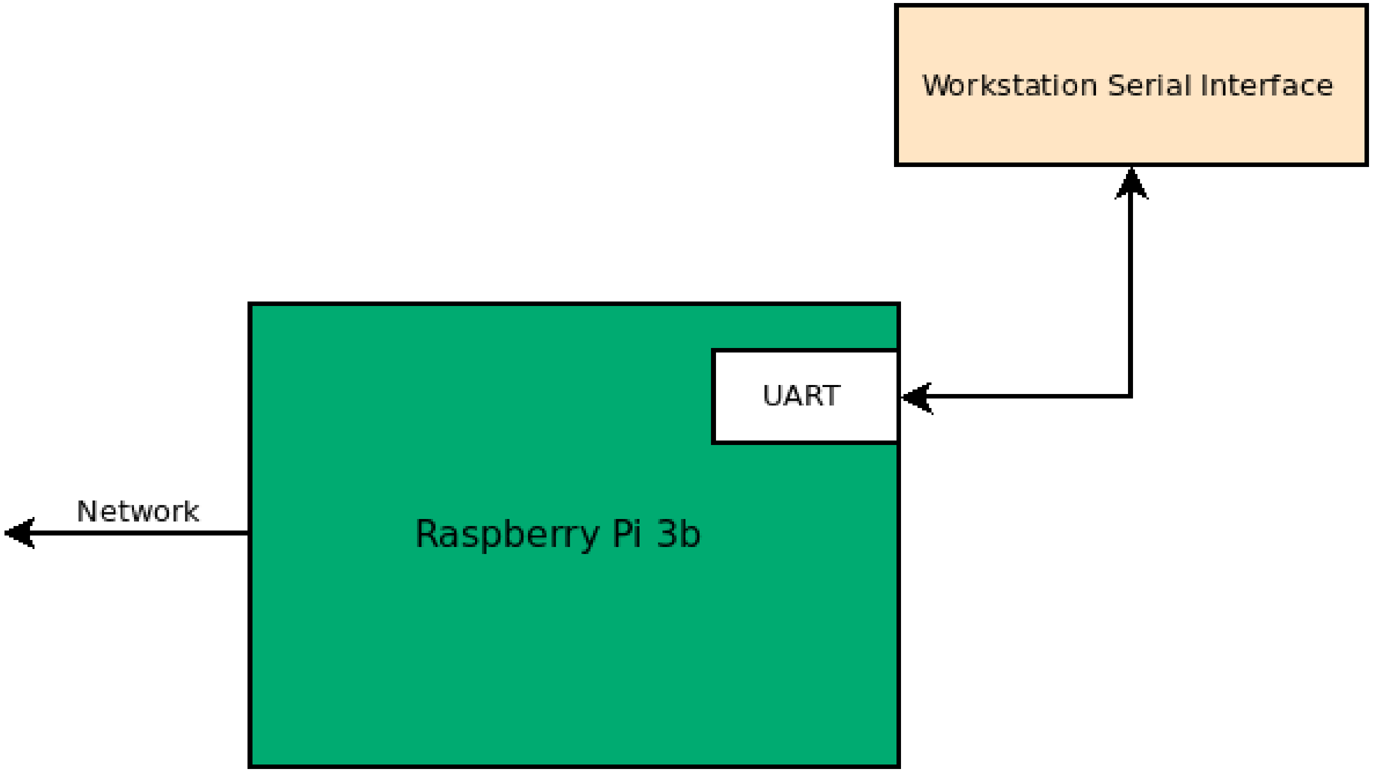


FIGURE 1.4
Typical SSN system architecture.

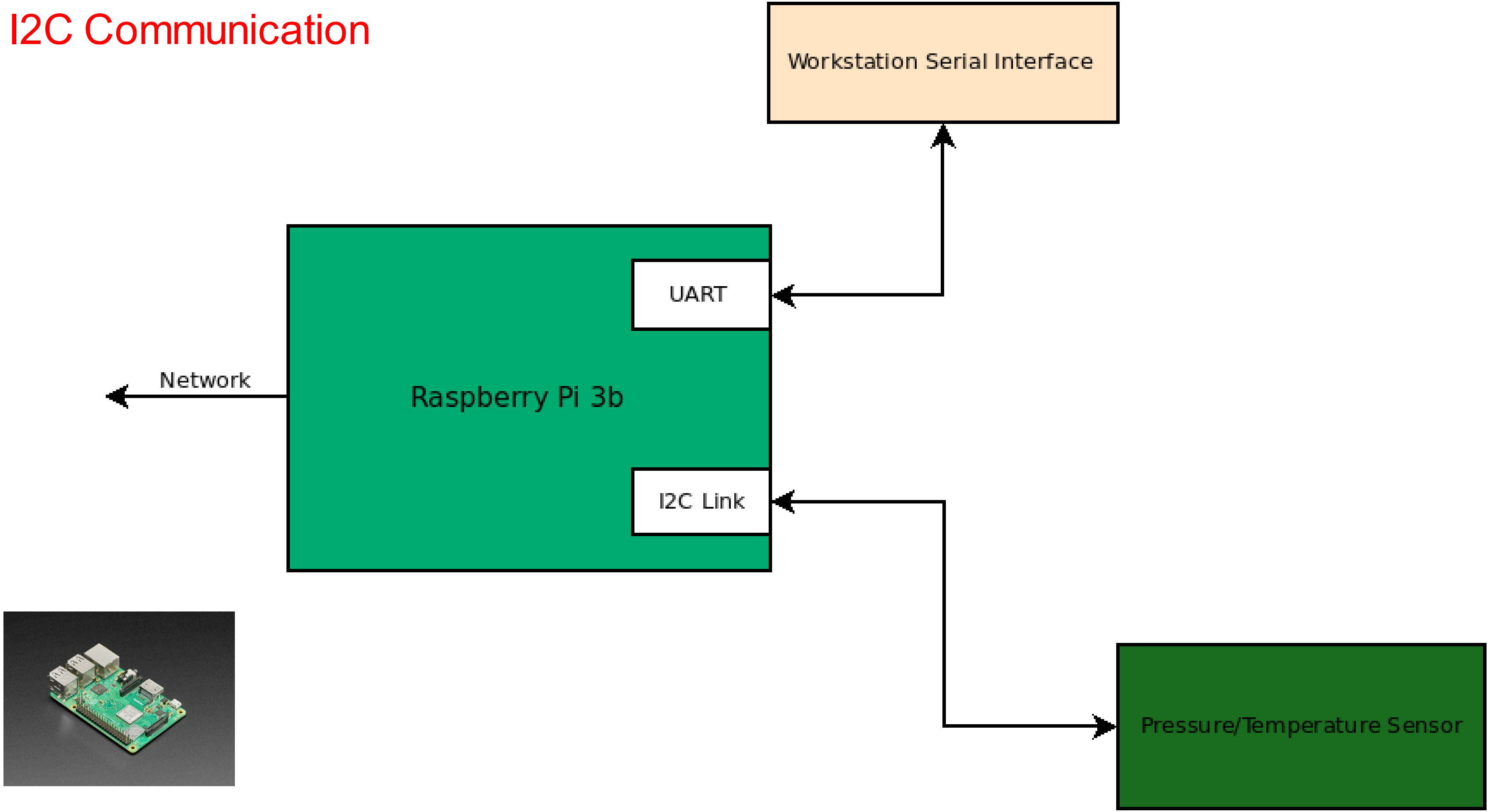
Siddesh, Deka, Srinivasa, Patnaik 2016

CPS Projects

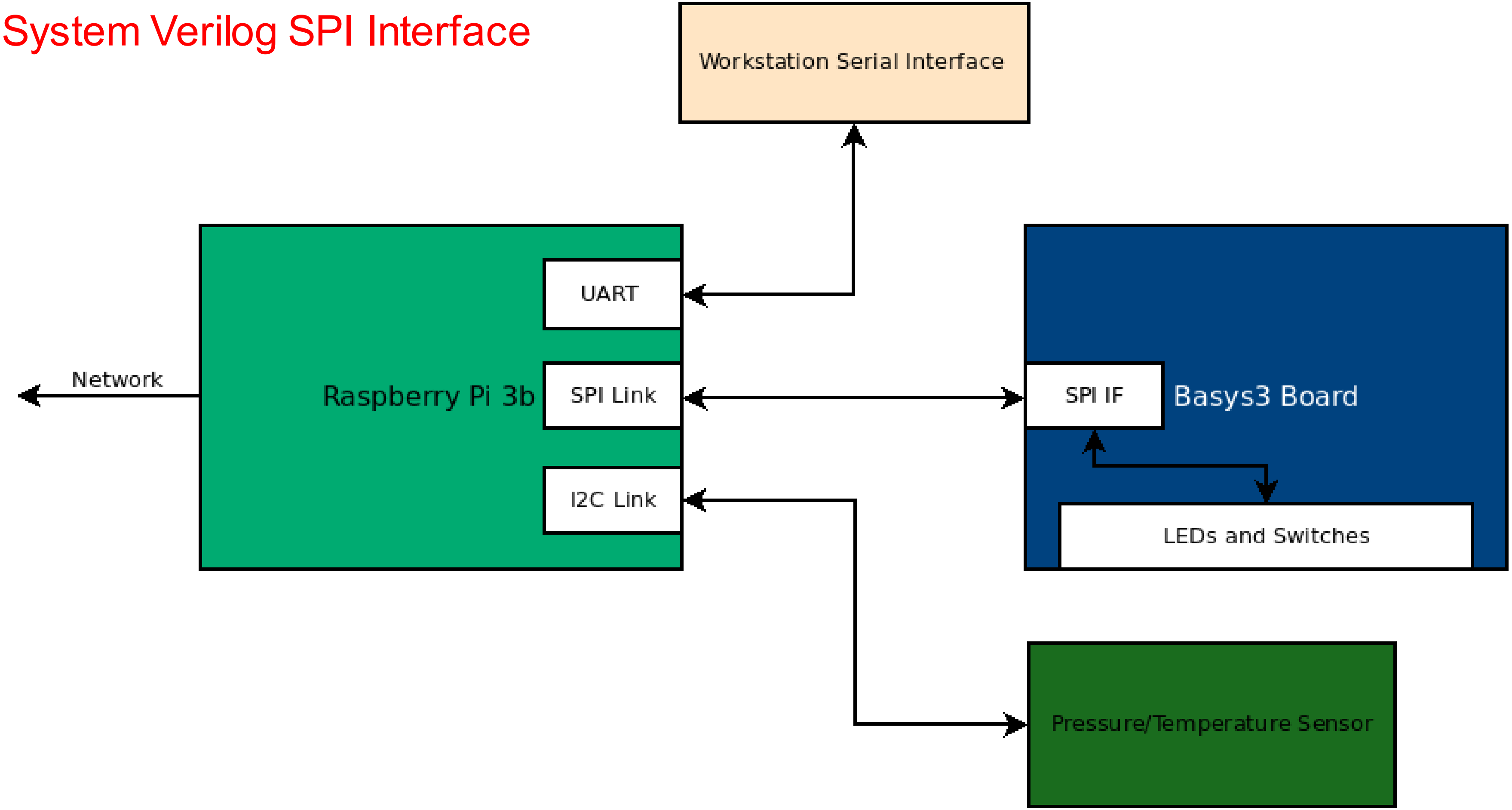
Raspberry Pi/UART



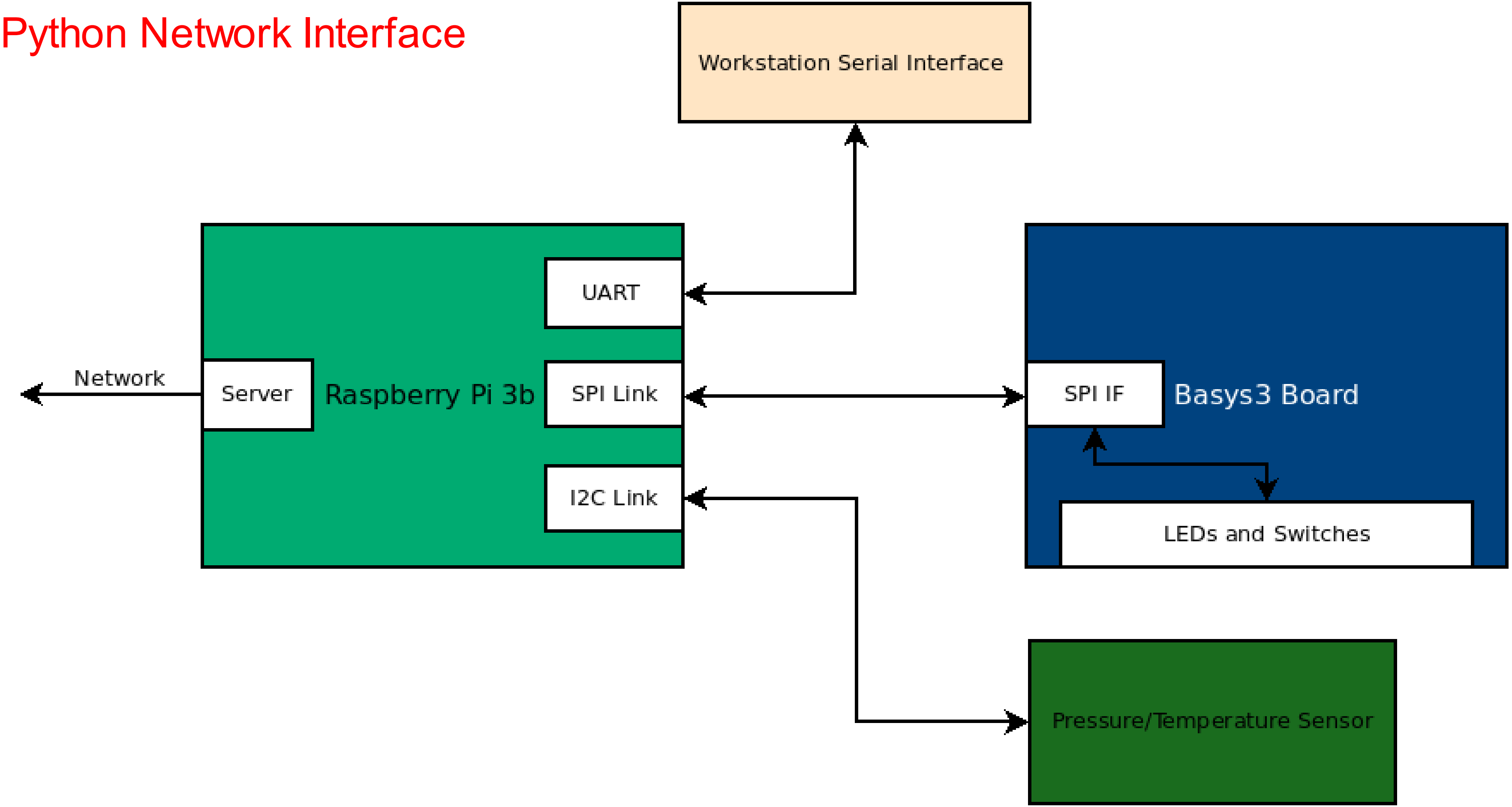
I2C Communication

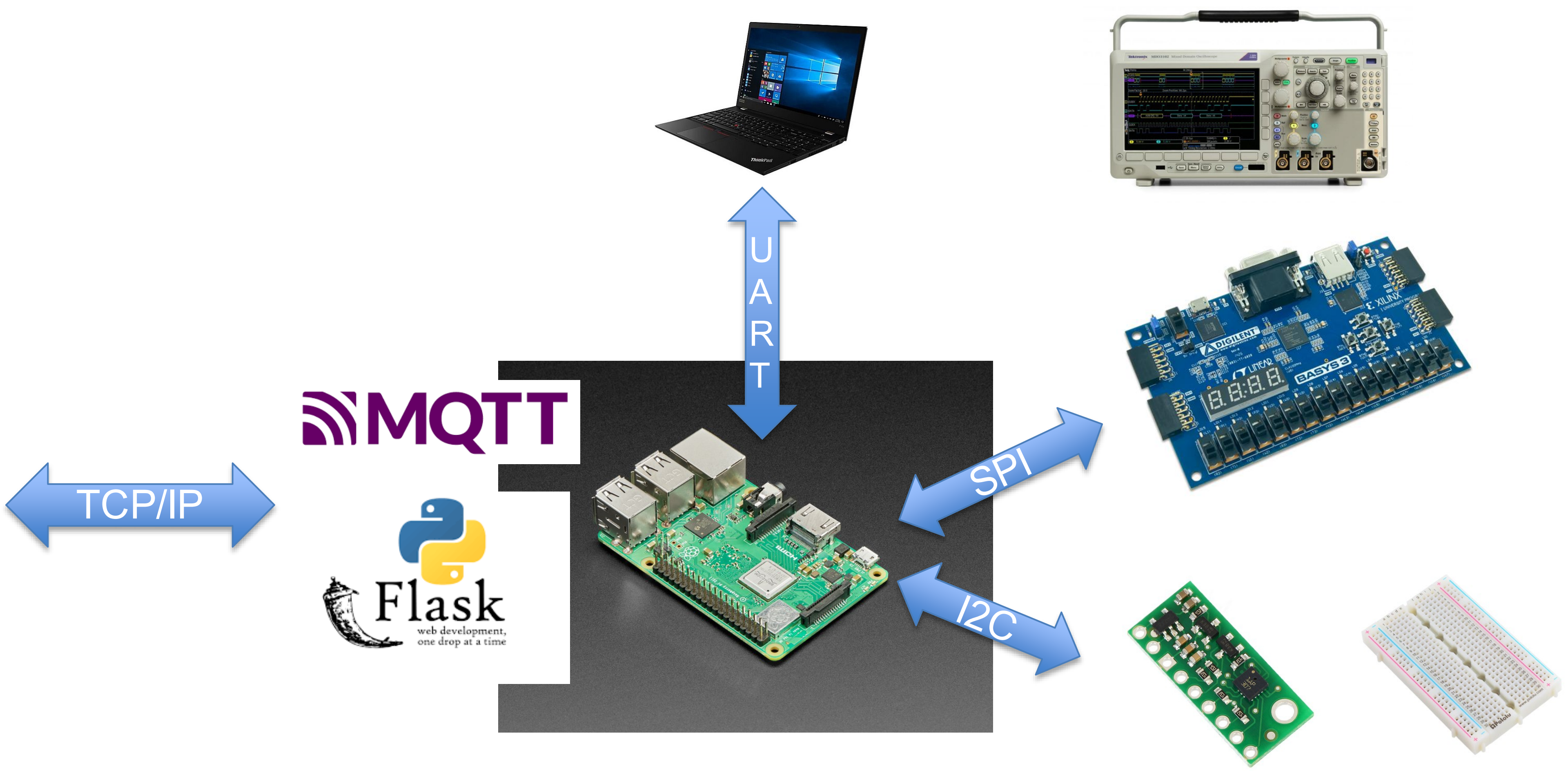


System Verilog SPI Interface



Python Network Interface




UART
MQTT
TCP/IP
SPI
I2C
Flask
web development,
one drop at a time

Schedule

Weekly Focus	Reading	Monday	Wed	Lab
Exam/CPS Introduction	Ref 1 Chapter 1	**3/8:** Exam 1	**3/10:** CPS Introduction	**Project 5 Raspberry PI Setup**
Raspberry Pi	Ref 2 Chapter 1-3	**3/15:** Pi Intro/UART Bus	**3/17:** Git/Github	
I2C Bus	Ref 3	**3/22:** I2C Bus	**3/24:** Wellness Day	**Project 6 I2C Pressure Sensor**
Python/Sensor	Ref 4, Ref 5	**3/29:** Classes/Modules	**3/31:** Pressure Sensor	
SPI	Ref 6	**4/5:** SPI Bus Overview	**4/7:** SPI HDL Design	**Project 7 SPI Connected I/O**
SPI	Ref 7 Chapter 1	**4/12:** SPI HDL Design	**4/14:** Sensor Memory	
Network Interface	Ref 7 Chapter 2	**4/19:** Ethernet Interface	**4/21:** MQTT	**Project 8 Network Interface**
MQTT/Flask	Ref 7 Chapter 14	**4/26:** Flask	**4/29:** Open Topic	

Final Exam Tues 5/4 10:10-12:10

https://engr210.github.io/

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

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.