Electronics for IoT

Introduction

Bernhard E. Boser
University of California, Berkeley
boser@eecs.berkeley.edu

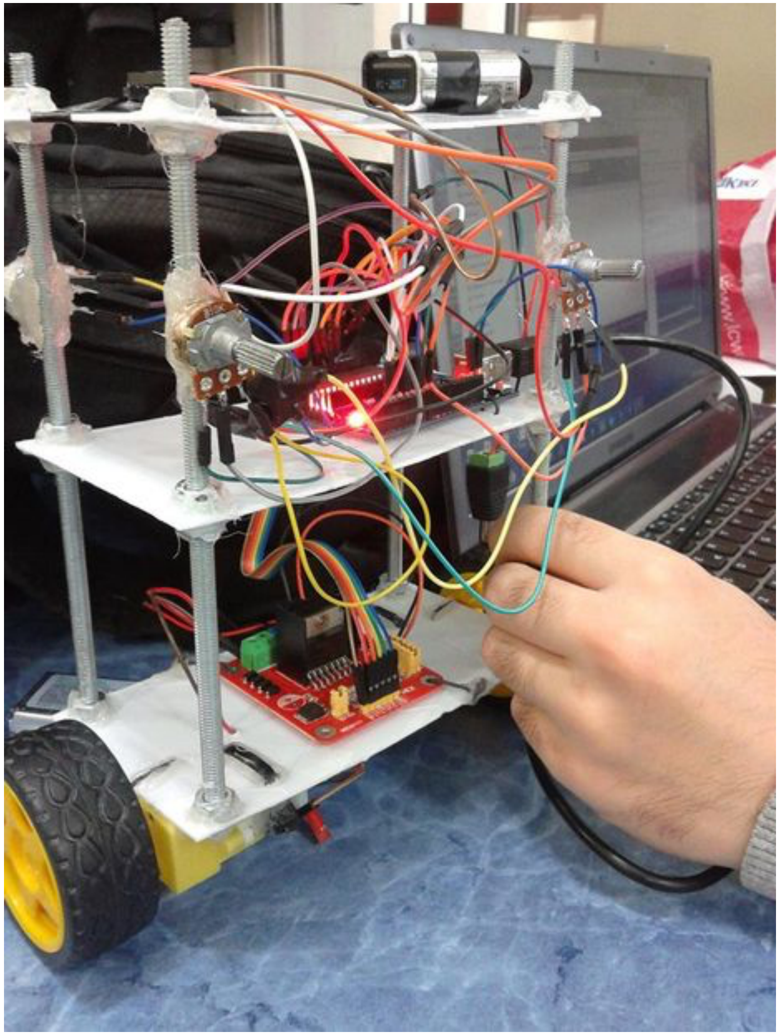
Welcome!

- IoT – Internet of Things
- Introductions & course organization
- What’s an MCU?

Electronics for the Internet of Things

- Sounds good, but what the heck is it?

Electronics for the Internet of Things



Electronics for the Internet of Things

IoP

Internet of People

Internet of *Things*

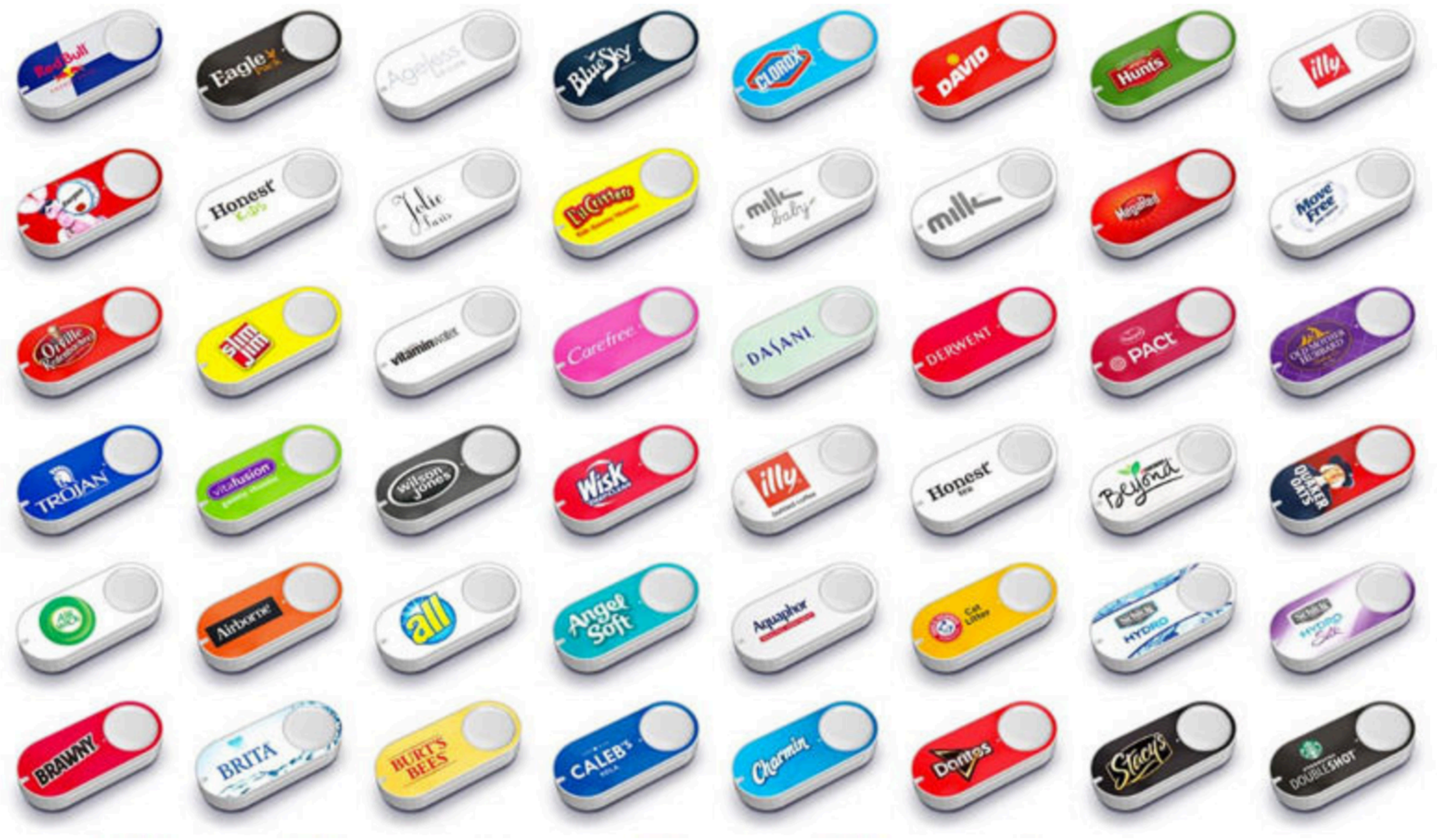
Sprinkler

“Smart” Sprinkler Controller

Other IoT Examples?

IoT Ideas?

Where are the opportunities?

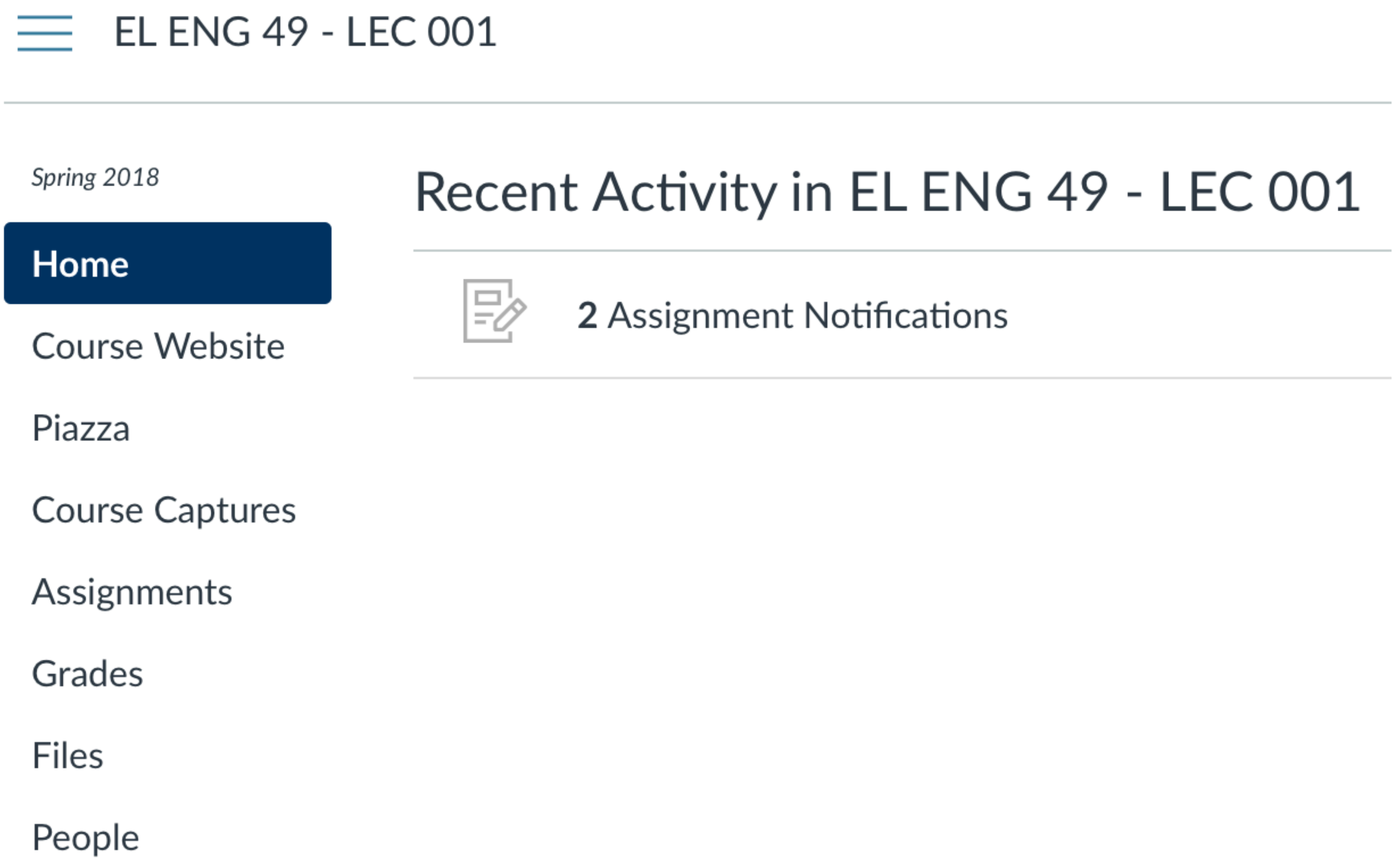


EE49 Logistics

Meet the Instructors

- Instructors:
  - Bernhard Boser
  - Kameshwar Poolla
- Teaching Assistants:
  - Mahsa Sadeghi
  - Thanh-Phong Nguyen
- Reader
  - TBD

EE49 on the Internet …



EL ENG 49 - LEC 001

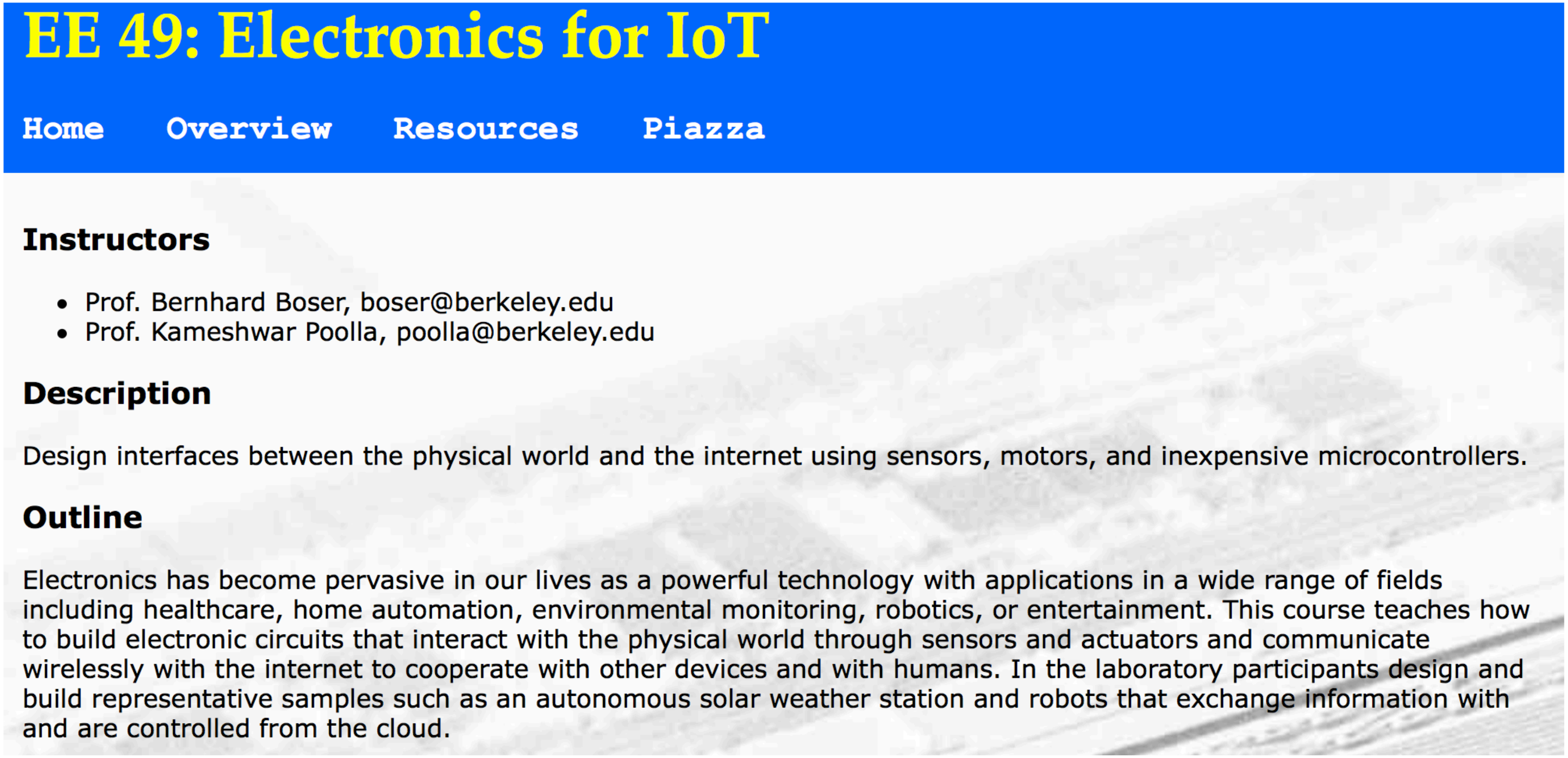
Spring 2018

- **Home**
- Course Website
- Piazza
- Course Captures
- Assignments
- Grades
- Files
- People

Recent Activity in EL ENG 49 - LEC 001

2 Assignment Notifications

Course Website



EE 49: Electronics for IoT

Instructors

- Prof. Bernhard Boser, boser@berkeley.edu
- Prof. Kameshwar Poolla, poolla@berkeley.edu

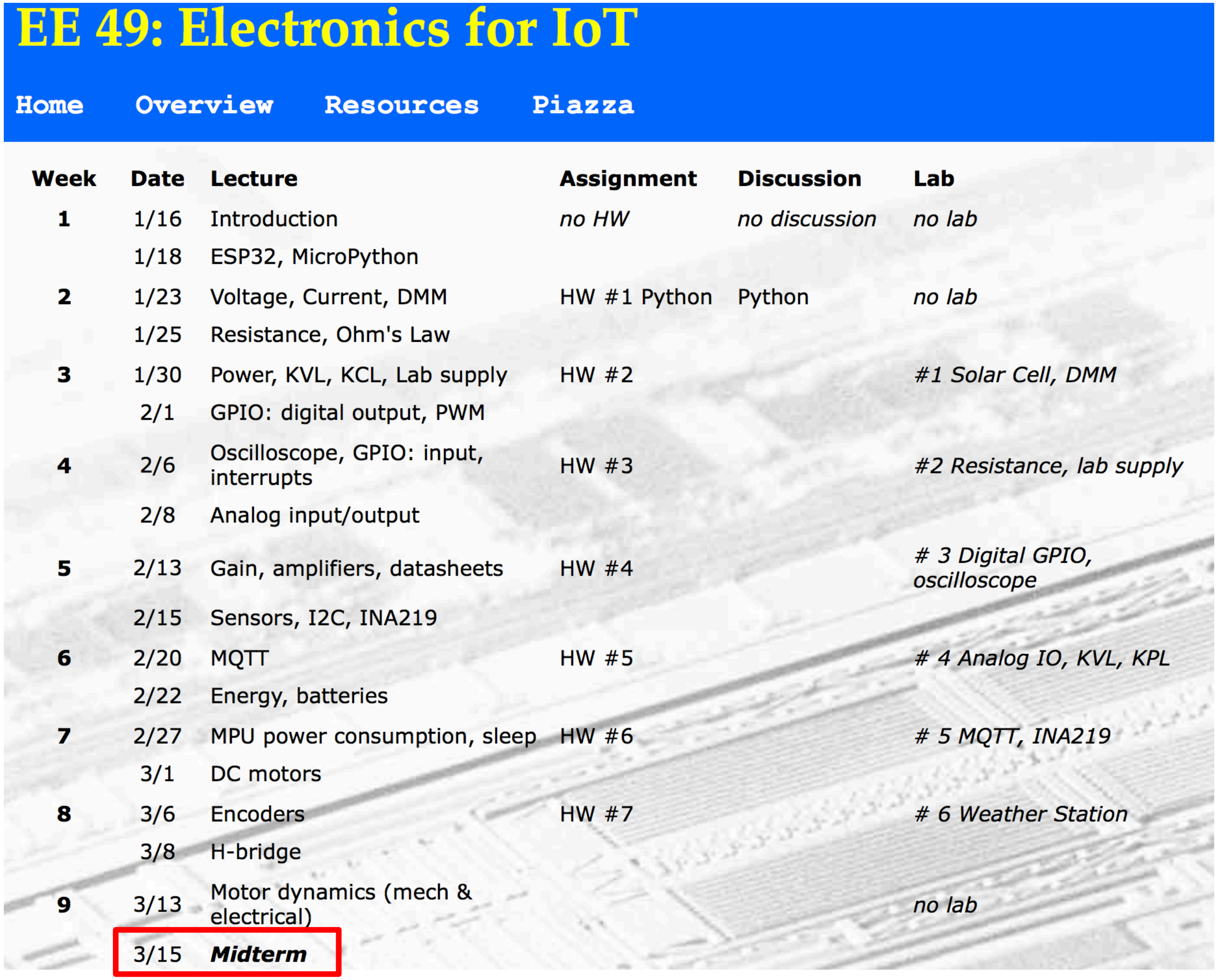
Description

Design interfaces between the physical world and the internet using sensors, motors, and inexpensive microcontrollers.

Outline

Electronics has become pervasive in our lives as a powerful technology with applications in a wide range of fields including healthcare, home automation, environmental monitoring, robotics, or entertainment. This course teaches how to build electronic circuits that interact with the physical world through sensors and actuators and communicate wirelessly with the internet to cooperate with other devices and with humans. In the laboratory participants design and build representative samples such as an autonomous solar weather station and robots that exchange information with and are controlled from the cloud.

Course Schedule

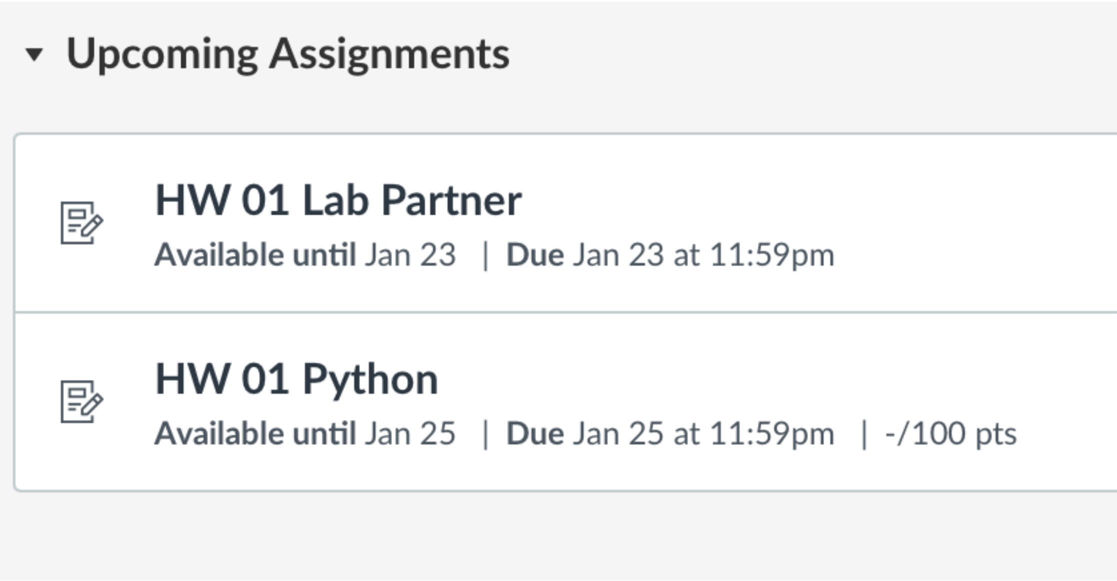


EE 49: Electronics for IoT

Home Overview Resources Piazza

Week	Date	Lecture	Assignment	Discussion	Lab
1	1/16	Introduction	*no HW*	*no discussion*	*no lab*
	1/18	ESP32, MicroPython			
2	1/23	Voltage, Current, DMM	HW #1 Python	Python	*no lab*
	1/25	Resistance, Ohm's Law			
3	1/30	Power, KVL, KCL, Lab supply	HW #2		*#1 Solar Cell, DMM*
	2/1	GPIO: digital output, PWM			
4	2/6	Oscilloscope, GPIO: input, interrupts	HW #3		*#2 Resistance, lab supply*
	2/8	Analog input/output			
5	2/13	Gain, amplifiers, datasheets	HW #4		*# 3 Digital GPIO, oscilloscope*
	2/15	Sensors, I2C, INA219			
6	2/20	MQTT	HW #5		*# 4 Analog IO, KVL, KPL*
	2/22	Energy, batteries			
7	2/27	MPU power consumption, sleep	HW #6		*# 5 MQTT, INA219*
	3/1	DC motors			
8	3/6	Encoders	HW #7		*# 6 Weather Station*
	3/8	H-bridge			
9	3/13	Motor dynamics (mech & electrical)			*no lab*
	3/15	***Midterm***			

Assignments



▾ Upcoming Assignments

HW 01 Lab Partner

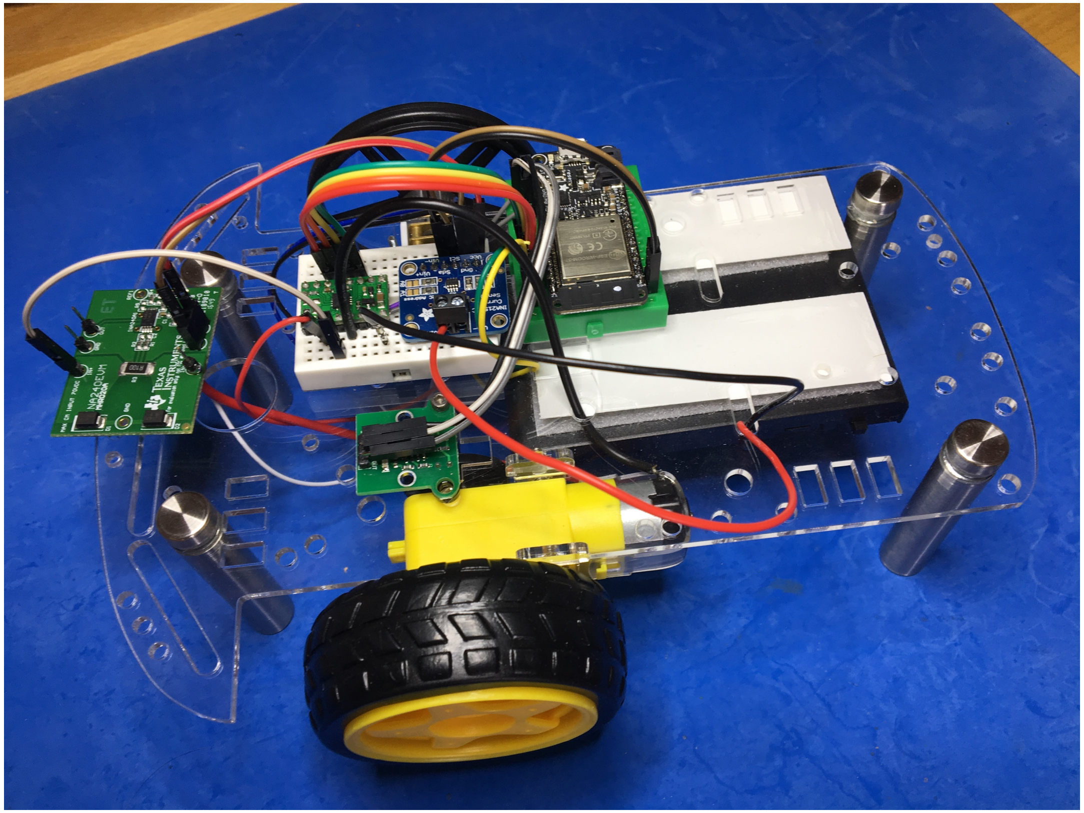
Available until Jan 23 | **Due** Jan 23 at 11:59pm

HW 01 Python

Available until Jan 25 | **Due** Jan 25 at 11:59pm | -/100 pts

Labs

- Weekly
  - Start 1/30/2018
- Mandatory
  - Makeup labs
- Project



Lab Organization

- Work in <u>teams of two</u>
  - Find team mate by next week!

- Lab assignment
  - Attend only lab you are enrolled in
  - Discussion: any is ok, subject to capacity

Friday 8am Lab Cancelled!

- If you signed up for that lab, you need to sign up for a different one!
- Tuesday 2pm lab is oversubscribed:
  - Workstations available for only 24 students
  - 25 signed up

Parts

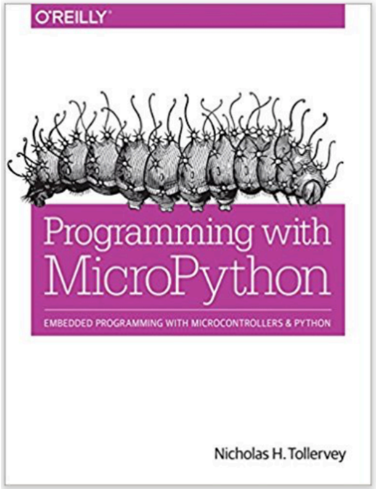
- Each **group** of two students gets a kit
- Use throughout semester!

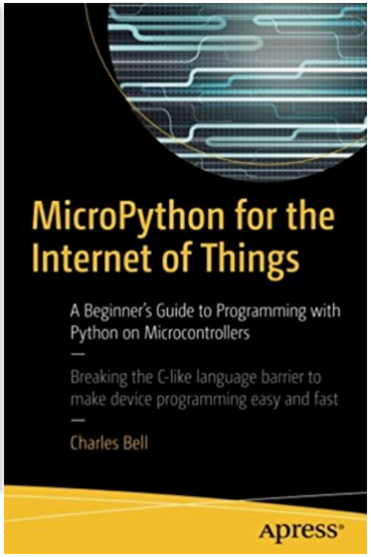
Prerequisites

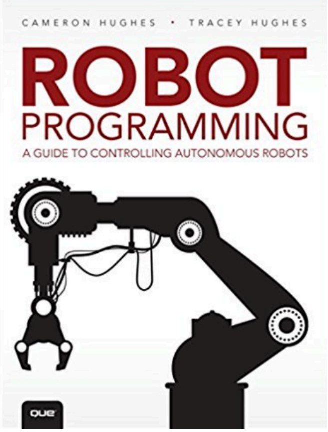
- Familiarity with computer programming such as taught in E7, CS10, CS61A or DS8.
- The course uses the (Micro)Python programming language.
  - Same as “normal” Python 3
  - Summary in next lecture
  - Discussions next week
  - Web resources
  - Homework #1

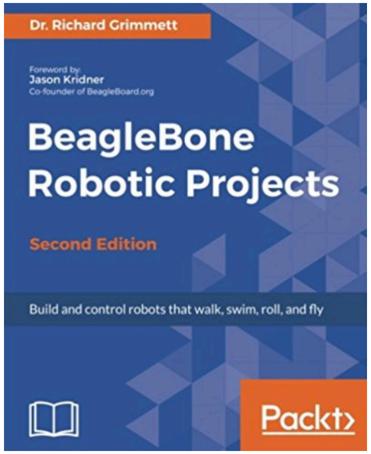
Textbook

- None
  - Save $200
  - Take notes …
- References
  - Google search
  - Programming:
    - https://stackoverflow.com
    - http://www.micropython.org
    - …
  - Electronics:
    - Manufacturers (datasheets, app notes)
    - http://www.instructables.com
    - …


O'REILLY
Programming with MicroPython
EMBEDDED PROGRAMMING WITH MICROCONTROLLERS & PYTHON
Nicholas H. Tollervey


MicroPython for the Internet of Things
A Beginner's Guide to Programming with Python on Microcontrollers
Breaking the C-like language barrier to make device programming easy and fast
Charles Bell
Apress


CAMERON HUGHES • TRACEY HUGHES
ROBOT PROGRAMMING
A GUIDE TO CONTROLLING AUTONOMOUS ROBOTS
QUE


Dr. Richard Grimmett
Foreword by: Jason Kridner Co-founder of BeagleBoard.org
BeagleBone Robotic Projects
Second Edition
Build and control robots that walk, swim, roll, and fly
Packt

Grades

- Weighted average of:
  - Labs (30%)
  - Project (15%)
  - Homework (15%)
  - Quizzes (20%)
  - Midterm (20%)
- Quizzes:
  - In-class (beginning of lecture)
  - ~ 2 weeks
- Lowest homework and quiz score omitted from grade calculation
  - Absences, birthdays, trips, other class midterm, …

Fun!

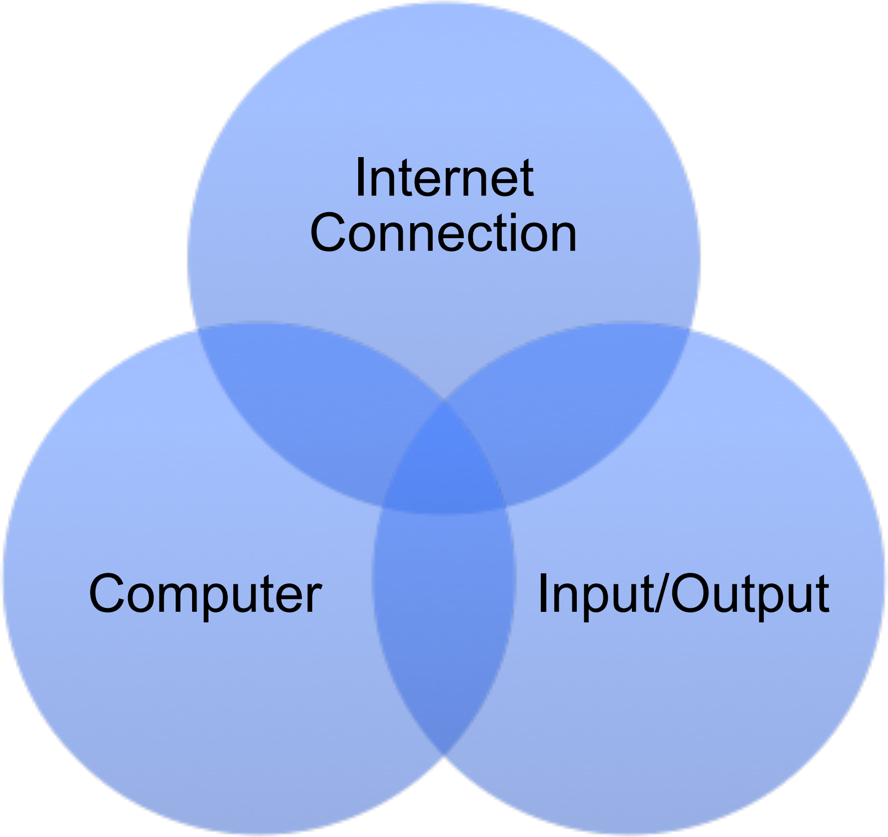
Questions?

Your Cool Project

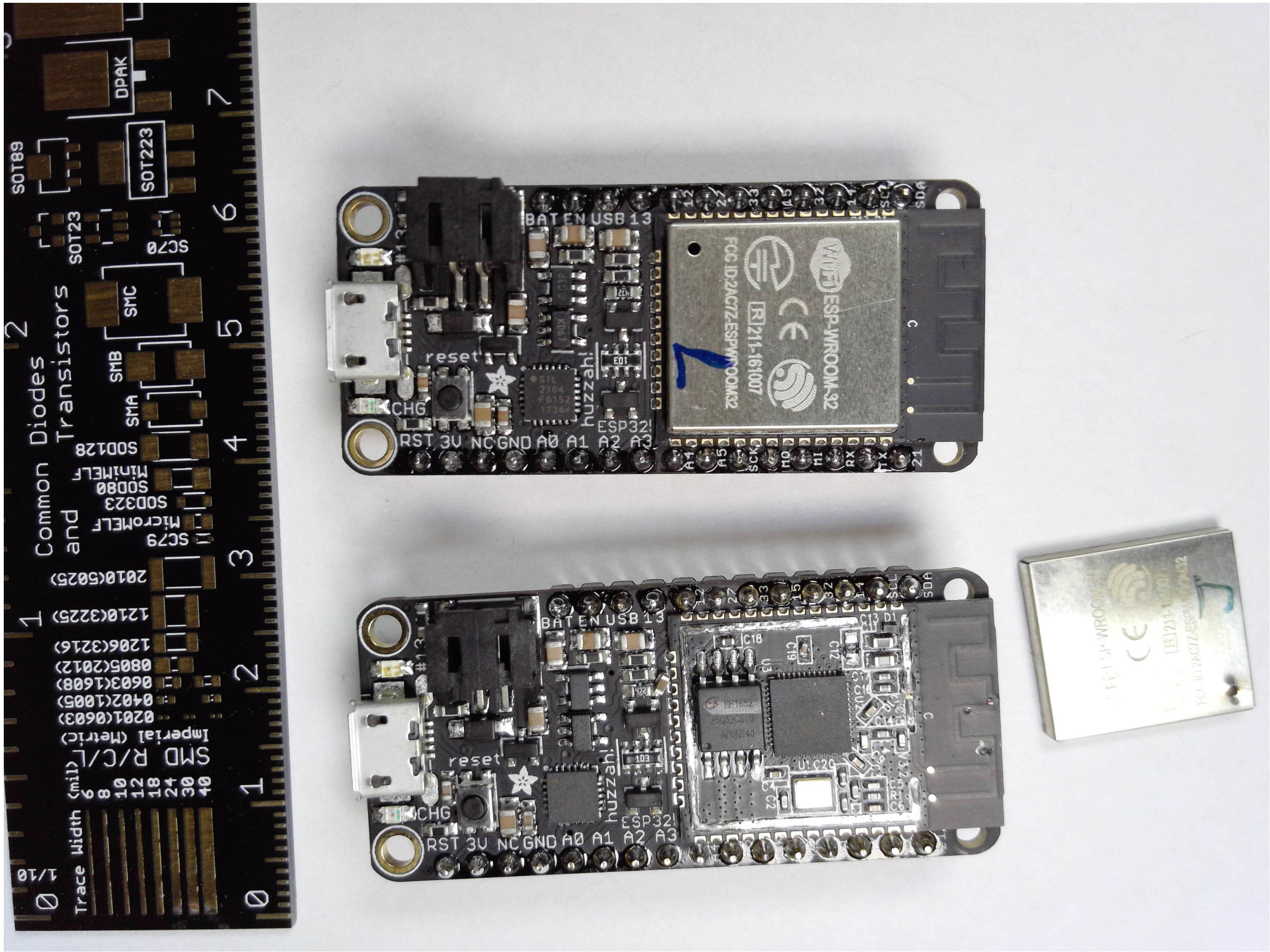
Internet

?

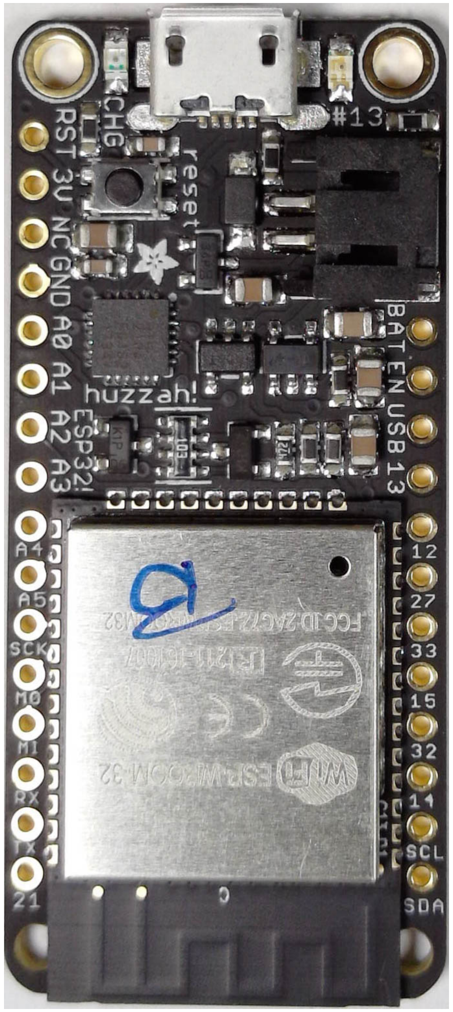
Microcontroller for IOT



EE 49 Microcontroller



EE 49 MCU: ESP32



Key Features

1. Internet
   – WiFI
2. Computer
   – Like any other, just smaller
3. Input / Output
   – Not your typical keyboard & display …

Espressif ESP32 Specs

- Espressif
  - http://espressif.com

- ESP32
  - https://esp-idf.readthedocs.io/en/latest/
  - Tensilica, dual core
  - 240MHz clock
  - 520 kB RAM
  - 4 MB flash (off-chip)
  - << $5 per chip (in volume)
  - ~ $20 for the board we are using
    - "HUZZAH32", https://www.adafruit.com/product/3405

Let’s Check it out!

Why (Micro)Python?

- Computer’s do not (yet?) understand plain English
- Typical micro-computer programming languages:
  - Assembly
    - Machine instructions, e.g. fetch byte from memory, add, …
    - Very low level
  - C
    - Big step up in usability from assembly
    - But still rather low level
    - Compiled: cumbersome to make changes to program
    - Most commonly used to program microcomputers today
  - Python
    - Powerful library & datatypes
    - Interpreted – changes take effect immediately
    - Drawback: slower, needs more memory

Next Time

- Python!