CSE160 – Project 0

Setup and Getting started ...

Originally done by Ashish Yadav and Alberto Cerpa
ayadav6@ucmerced.edu
acerpa@ucmerced.edu

Modified and updated by Hamid Rajabi
hrajabi2@ucmerced.edu

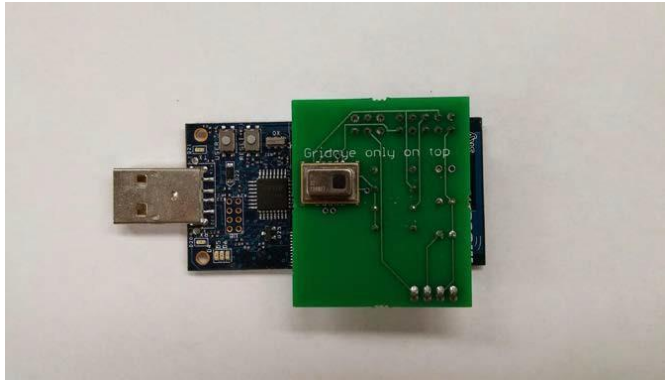
Wireless Sensor Networks (WSN)

-- The CSE 160 projects are aimed for Wireless Sensor Networks purposes.

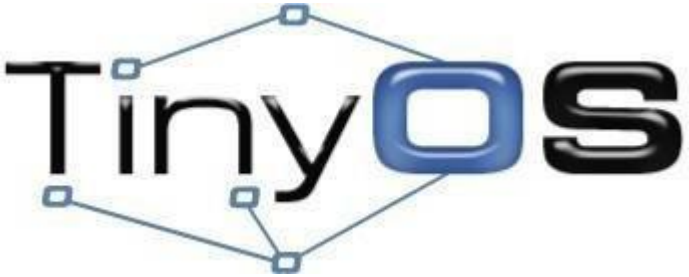
-- A wireless sensor network (WSN) is a wireless network consisting of spatially distributed autonomous sensors to monitor physical or environmental conditions.

Potential applications:

Environmental monitoring of air, water, and soil -- Structural monitoring for buildings and bridges -- Industrial machine monitoring, etc …



Software: TinyOS



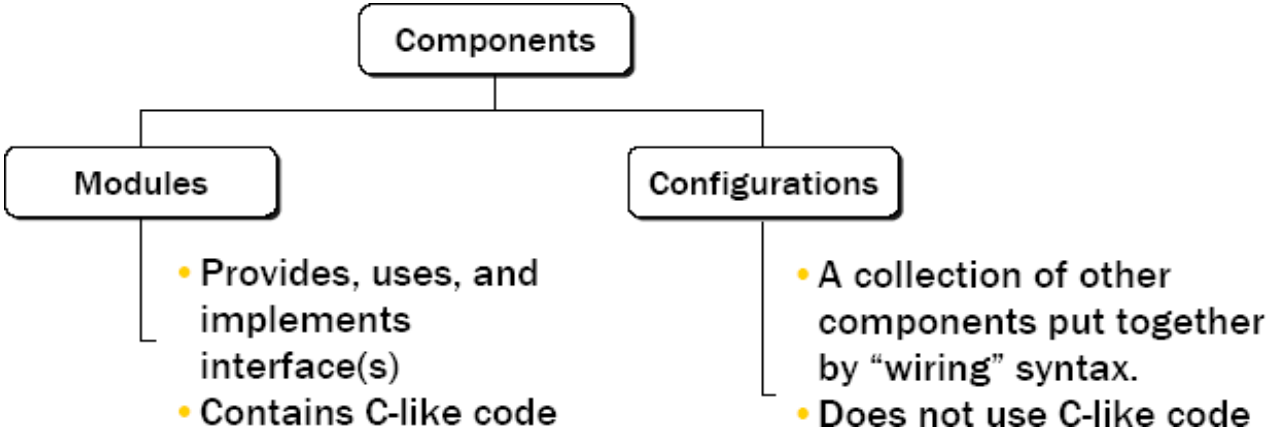
- Open Source operating system designed for low-power wireless devices
- Uses in Sensor networks, Personal area networks, Smart buildings, etc.

- Traditional OSes are not suitable for networked sensors (they are huge!)
- Real time

http://www.tinyos.net/

NesC

- Coding language
  - NesC (Network Embedded System C)
    - Basic unit of nesC code is a component
    - Components connect via interfaces
      - Connections called “wiring”



TOSSIM

- Simulates entire TinyOS application
- Replaces components with simulation implementation
- Core code = tos/lib/tossim
- 2 interfaces :
  - C
  - Python

Project Outlines

Project #0 - Getting Started

Project #1 – Flooding and Neighbor Discovery – 5 %

Project #2 – Routing Table – 15 %

Project #3 – Reliable Transport - 25 %

Project #4 – Application Layer – 5 %

You are given a Skeleton code ☺

Words of Advice

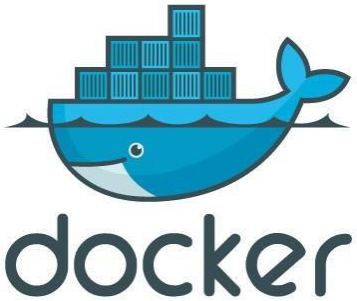
- Allocate enough **TIME** to it, try to finish early.
- Partial credit is better than no credit.

- All students need to demo their work individually, no matter if they have done it with a partner or not.

- Please do your demo ON-TIME; you need to demo each project BEFORE the next project due. So, please don't postpone it for a later time in the semester!

- In each lab session, the priority would be for those who have formally registered in that session. The rest of the students (if any) may be evaluated pending the availability of time.

Suggestions and Expectations



- Get comfortable with programming
- C / Python
- Start using Linux

- Docker -- (Makes development a breeze) – not related to the course

  --- Docker is another platform on which you run your TinyOS code

  --- So, you need to install Docker and then pull the TinyOS image to run your code.

  --- You don’t have to be expert in Docker and TinyOS env. (Just know how to program in C and Python)

TODO list:

1) **Install Docker on your machine:**

   Windows: https://docs.docker.com/desktop/windows/install/

   Mac: https://docs.docker.com/desktop/mac/install/

2) **Install TinyOS:**

   Files -> Projects - > Project 0 -> Project 0 - Docker installation instruction.pdf

3) **Get familiar with a few useful Docker commands:**

   1. Watch my recorded video: Media Gallery -> Docker commands
   2. Go to: Discussions -> Useful Docker commands

4) **Learn how to organize a NesC code:**

   1. Media Gallery -> About the Skeleton codes
   2. Media Gallery -> wiring the modules
   3. Discussions -> An example of how to organize a code in nesC (this is not my video, BTW!)
   4. Find a collected info about past students' understandings: Files -> Projects -> Project 0 -> TinyOS Modules