Introduction to TinyOS & nesC Programming (II)

Recall from previous Lab

1. Interfaces
2. Components
   a. Configuration
   b. Modules
   c. Private

The question is why do we need component based Programming?

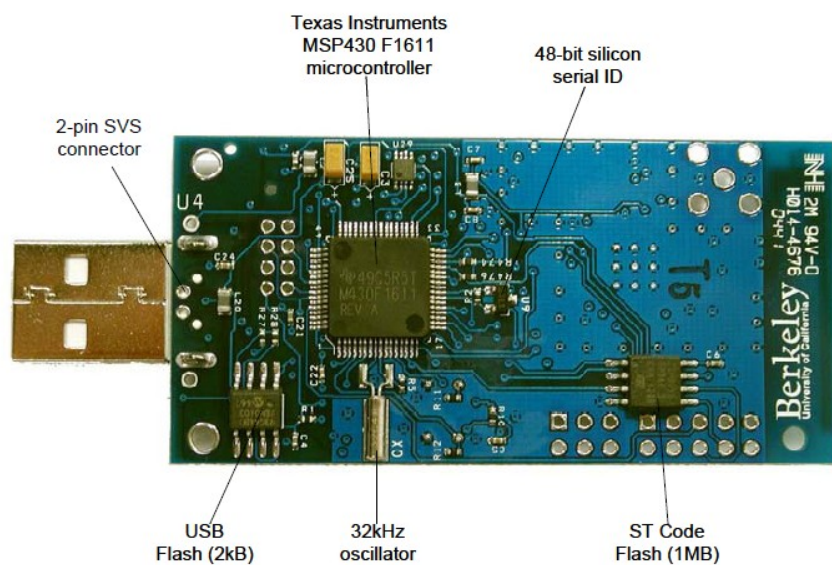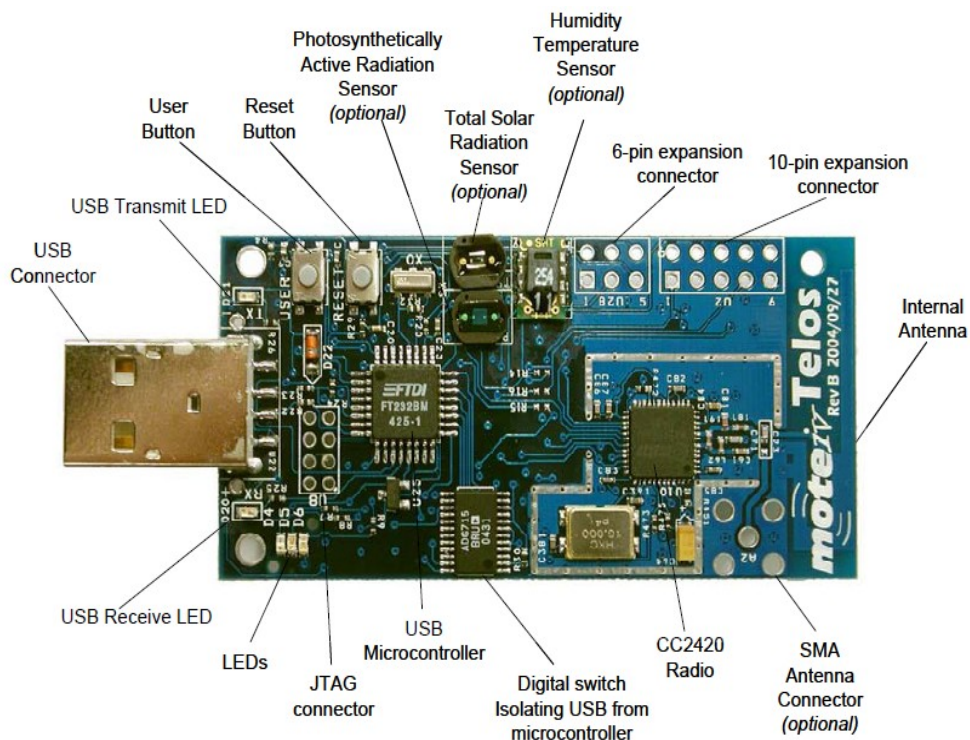1. Minimizing the code.
2. Split phase programming.
3. Unified API.

What is split phase programming?

Objective

The objective of this lab is

1. Familiarize your self with the telos B motes.
2. Know how to program the motes.
3. Write your first nesC code.

Crossbow TelosbB A.K.A Motive Tmote

Features

- 250kbps 2.4GHz IEEE 802.15.4 Chipcon CC2400 Wireless Transceiver
- Interoperability with other IEEE 802.15.4 devices
- 8MHz Texas Instruments MSP430 microcontroller (10k RAM, 48k Flash)
- Integrated ADC, DAC, Supply Voltage Supervisor, and DMA controller
- Integrated onboard antenna with 50m range indoors / 125m range outdoors
- Integrated Humidity, Temperature, and Light sensors
- Ultra low current consumption
- Fast wakeup from sleep (<6μs)
- Hardware link-layer encryption and authentication
- Programming and data collection via USB
- 16-pin expansion support and optional SMA antenna connector
- TinyOS support : mesh networking and communication implementation
- Built in un-calibrated temperature sensor.

The device is based on an open hardware design by UC Berkley

For more information see the datasheet.

Exercise 1

Write a TinyOS application that will increment a counter every one second and send it wirelessly to another mote whose ID is 10. Also use the LED's to indicate the success or failure of sending a message.

Exercise 2

Write a TinyOS application that receives a message from another mote and extracts the counter value from the packet then displays the lower 3 bits on the LED's.

The solution for the first exercise will be discussed in class. But you should submit the other one before the next lab.

Hint: use the following sources
1. TinyOS.net tutorials.
2. Look at the applications in the app folder of tinyOS.
3. Discuss any difficulties you are facing on the WebCT of the course.