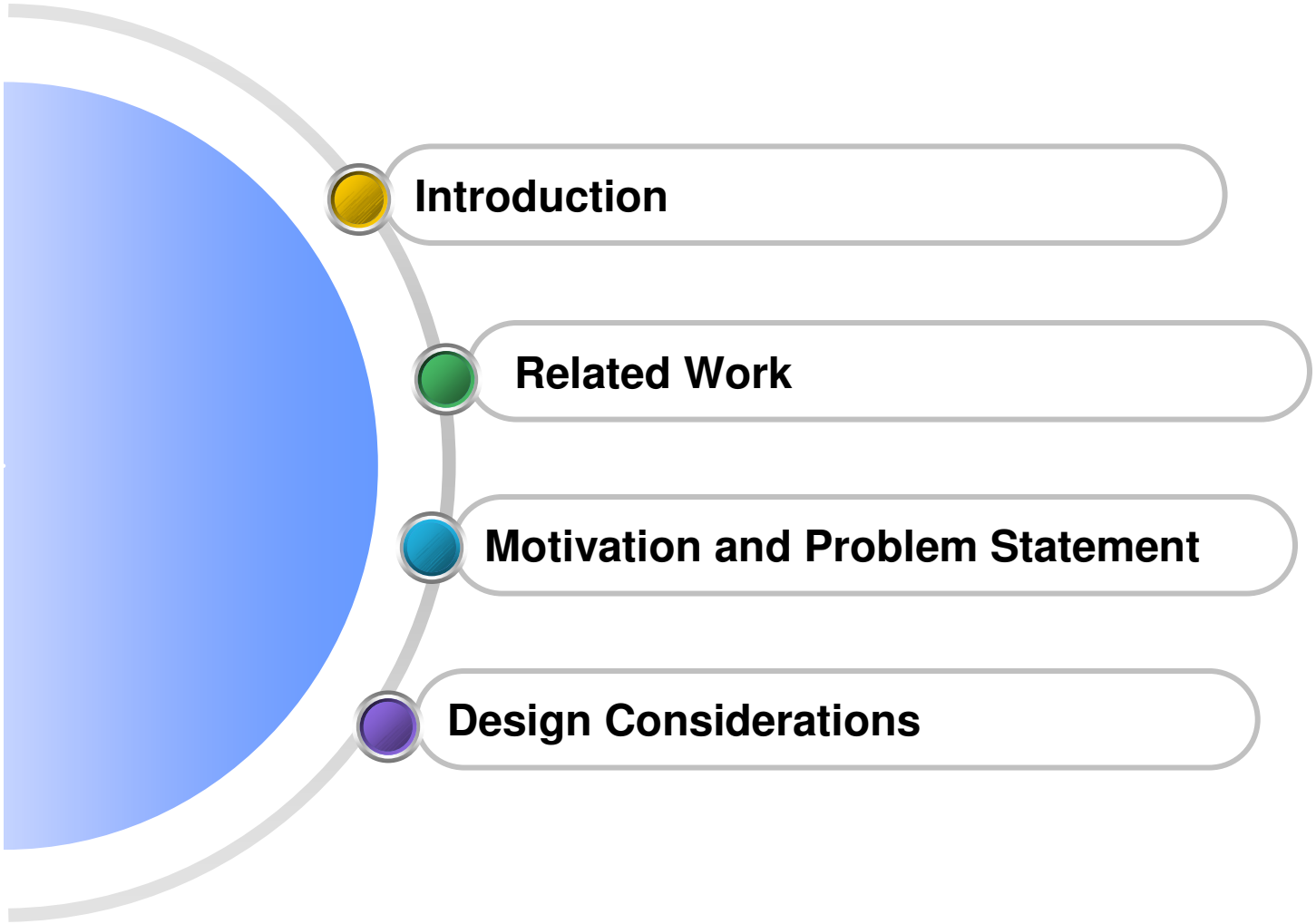
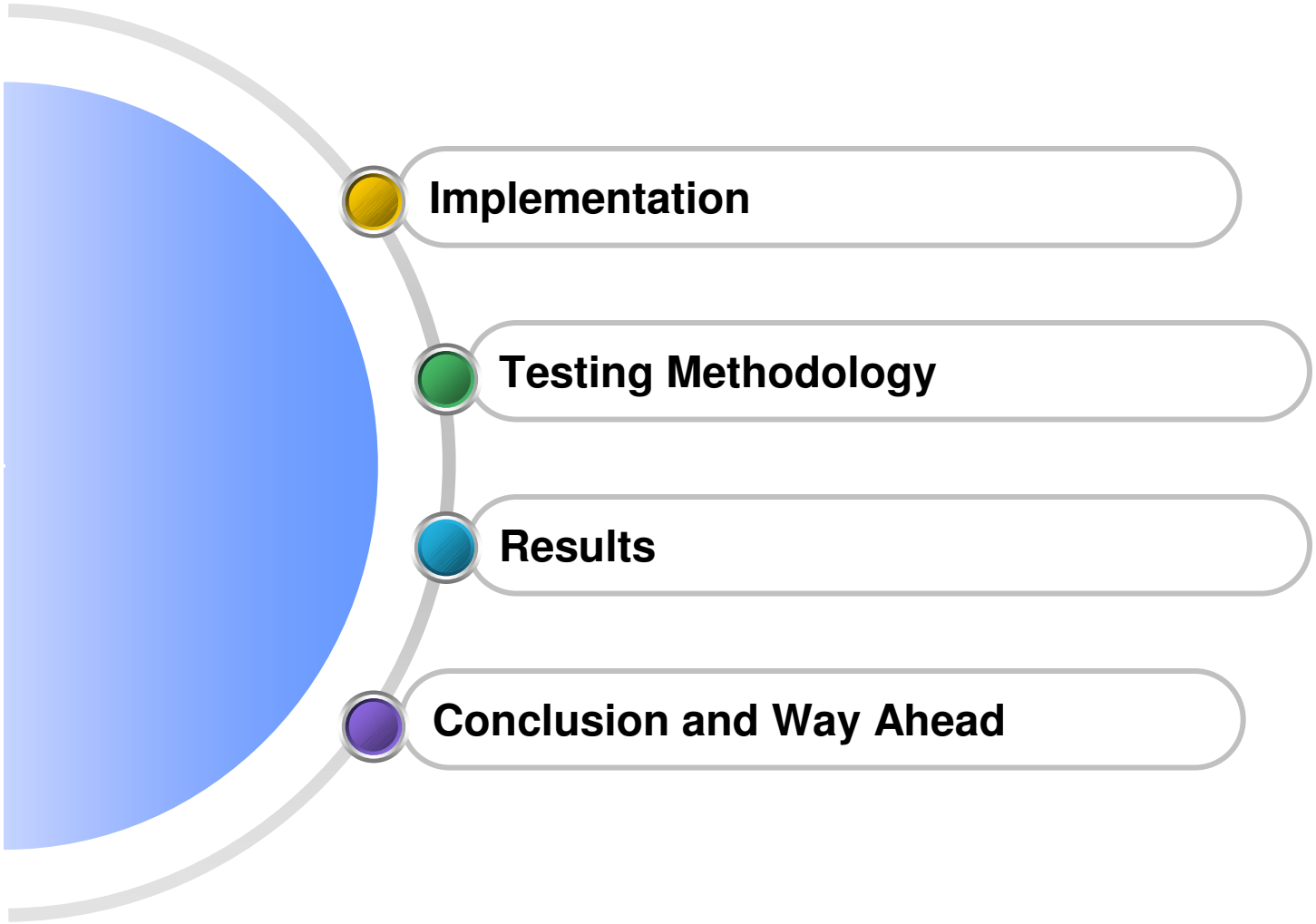
KMote - Design and Implementation of a low cost, low power platform for wireless sensor networks

Naveen Madabhushi

Presentation Outline



Presentation Outline



Introduction

- Remote Sensing
  - Ability to observe large scale spaces
  - Area of interest very large, does not provide high fidelity
- In Situ Sensing
  - Provides observations at required resolution
  - Possible to observe changes in obstructed environments

Gulf of Kutch, Gujarat (Source: Google Maps)

Introduction

- Wireless Sensor Networks (WSNs)
  - A new paradigm in distributed sensing
  - Provides in situ sensing
  - Dense deployment of inexpensive sensor nodes
  - Sensors sense parameter of interest and relay to a central unit
  - Wirelessly network these nodes
  - Myriad experimental deployments worldwide

Introduction

- WSN applications
  - Monitor space
  - Monitor things
  - Monitor interactions of things in an encompassing space

  (Source: Lecture notes of Dr Bhaskaran Raman)

Introduction

- Monitor Space

Habitat Monitoring

Monitor Rivers

Introduction

- Monitor Things

Railway Bridges

Machinery Spaces

Introduction

- Monitor interaction of things in an encompassing space

Volcano monitoring

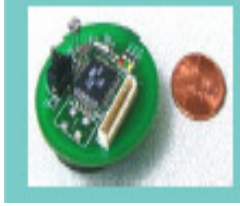
Zebra tracking

Introduction

- Wireless sensor node
  - Building blocks of WSNs
  - A hardware platform consisting of processor, radio, memory and sensors
  - Tiny in size and hence called *mote*
- Characteristics of Mote
  - Low Cost
  - Low Power Consumption
  - Battery Powered
  - Onboard Sensors and ability to interface external sensors
  - Rugged and easy to deploy

Related work

- A well researched field
- Mote evolution dates back to 1998 and continues

WeC 1998	René 1999	René 2 2000	Dot 2000	Mica 2001	Mica2Dot 2002	Mica 2 2002	Telos 2004
							

(Source: Hotchips 2004)

Related Work

Mote Type (Year)	WeC 1998	Rene 1999	Rene2 2000	Dot 2000	Mica 2001	Mica2Dot 2002	Mica2 2002	Telos 2004
Processor	AT90LS8535		ATMega163		ATMega128			MSP430
Radio Chip	TR1000					CC1000		CC2420
Flash	32 KB				512 KB			128 KB
Sensors	No on-board sensors			Yes	No on-board sensors			Yes
Frequency of Operation	916.5 Mhz					868 Mhz		2.4 Ghz
Transmit Power	0 dBm							
Max Data Rate	10 Kbps				40 Kbps	38.4 Kbps		250 Kbps
Power Consumption	36 mW					42 mW		35 mW
Sleep Power (µW)	45				75			6
wakeup Time (µS)	1000		36		180			6
Interface	IEEE 1284 and RS232							USB

(Source - Telos:Enabling ultra-low power wireless research,IPSN ’05)

Related work

- A similar work undertaken at IIT Delhi
- Developed and demonstrated a mote called *Rete*
- Rete operates in 433 Mhz spectrum
- Sales and marketing by US based company, ElfSys

Related work

Comparison of Popular motes

	MicaZ	Tmote	Tiny Node
Processor	AT Mega 128	MSP 430	MSP 430
Radio	CC 2420	CC 2420	XE 1205
Frequency	2.4 Ghz	2.4 Ghz	868 – 870 Mhz
Tx Power (Max)	0 dBm	0 dBm	0 to +12 dBm
Receiver Sensitivity	- 90 dBm	- 95 dBm	-121 @1.2Kbps -101 @152.3Kbps
Max Data Rate	250 Kbps	250 Kbps	152.3 Kbps
Range	Upto 100m	Upto 150m	200m @ 76.8 Kbps
Power Consumption (Max Power Txn)	28 mA @ 0dBm	21mA@ 0dBm	25mA@0 dBm 62mA@+12 dBm
Interface	USB/Serial	USB	Serial
Operating System	Tiny OS	Tiny OS	Tiny OS

(Source : Respective datasheets)

Motivation

- Motes not easily available to educational and research institutions
- Need to be imported. Not all institutes have administrative procedures in place
- Lead time for import in the order of 2-3 months
- Expensive – Rs 7740 for Tmote

Problem Statement

- To design and implement a mote
- Low cost, low power and easy availability in India.
- Hardware Design
  - A simple prototype
  - Use telos as reference design
- Software Integration
  - Modify/add software modules to make it compatible with TinyOS

Expected Thesis Contributions

- Design of a low cost, low power hardware platform for use in sensor networks.
- Build a simple working prototype that can be improved upon.
- Easy availability of motes for research/educational purposes.

Design Considerations

Comparison of Tmote and Tiny Node

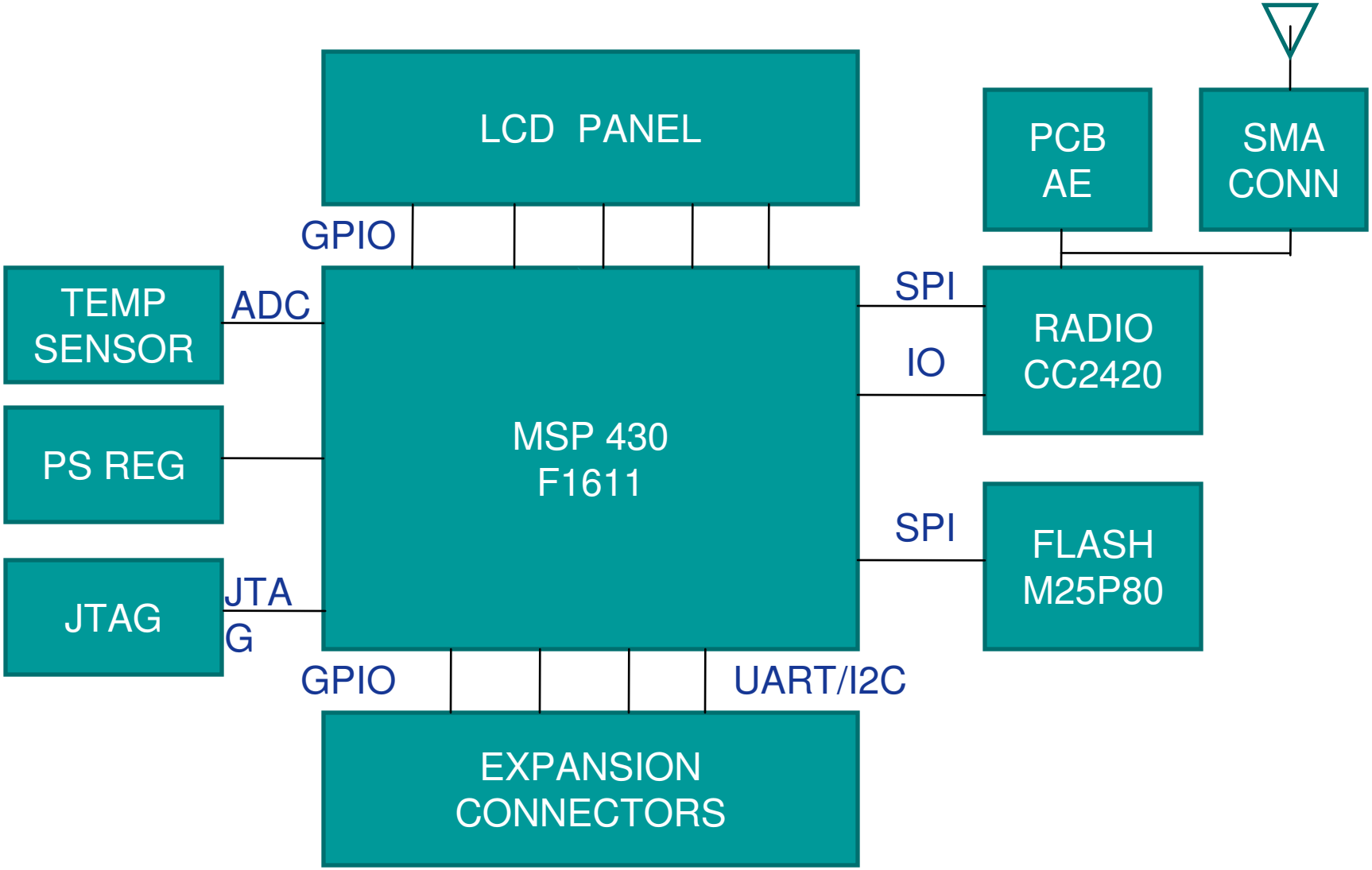
	Tmote	Tiny Node
Processor	MSP 430	MSP 430
Radio	CC 2420	XE 1205
Frequency	2.4 Ghz	868 – 870 Mhz
Tx Power (Max)	0 dBm	0 to +12 dBm
Receiver Sensitivity	- 95 dBm	-121 @1.2Kbps -101 @152.3Kbps
Max Data Rate	250 Kbps	152.3 Kbps
Range	Upto 150m	200m @ 76.8 Kbps
Power Consumption (Max Power Txn)	21mA@ 0dBm	25mA@0 dBm 62mA@+12 dBm
Interface	USB	Serial
Operating System	Tiny OS	Tiny OS

(Source : Respective datasheets)

Design Considerations

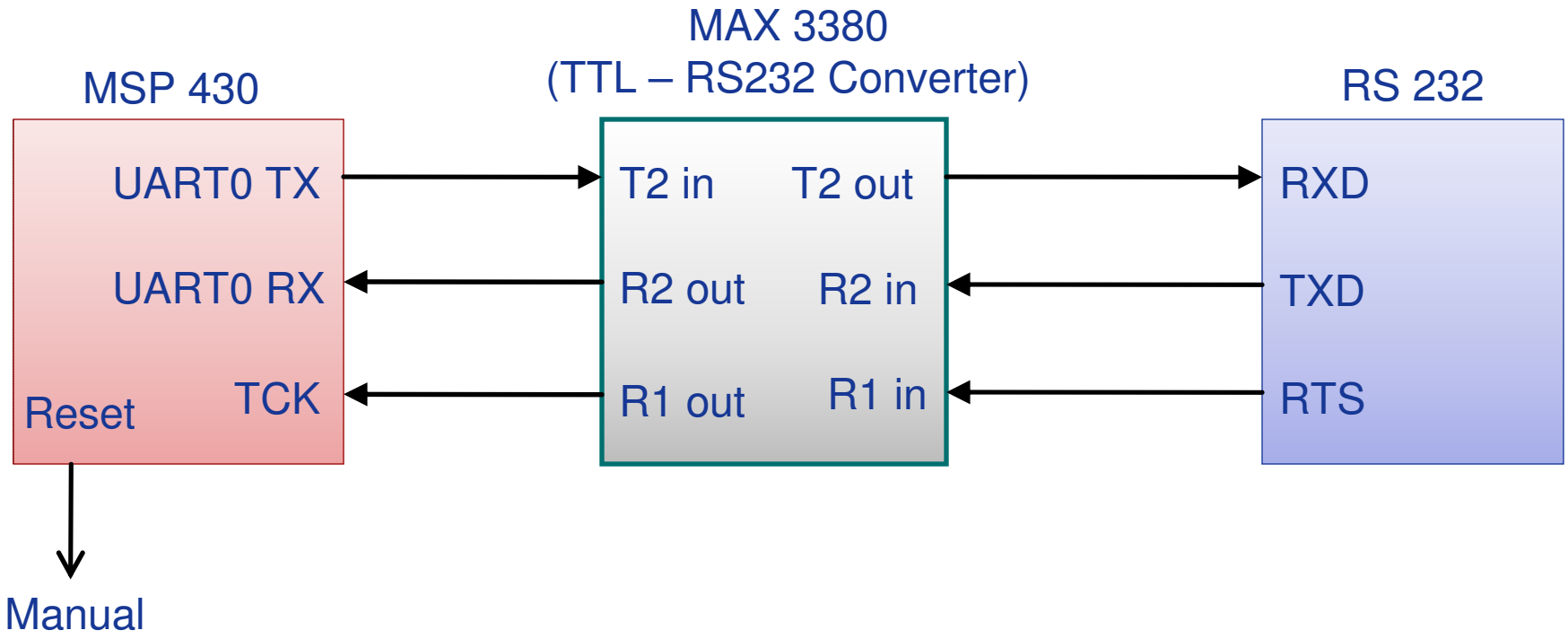
- **Differences from Tmote design**
  - Serial interface in place of USB
  - Integration of temperature sensor
  - LCD Panel interface
  - GPIO pins increased to 26
  - Form factor larger to accommodate LCD Panel
  - Mounting holes increased to 4 for better mechanical strength

Architecture of KMote



PCB Design – Design Issues

- **Serial Interface**
  - **USB interface replaced with serial interface for simplicity**
  - **MAX 3380 used. It has two transceivers.**



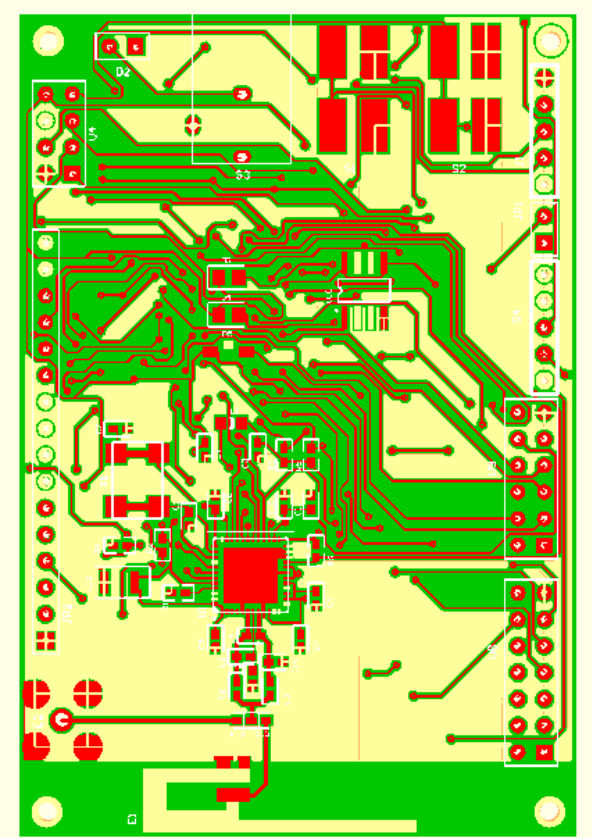
PCB Design – Design Issues

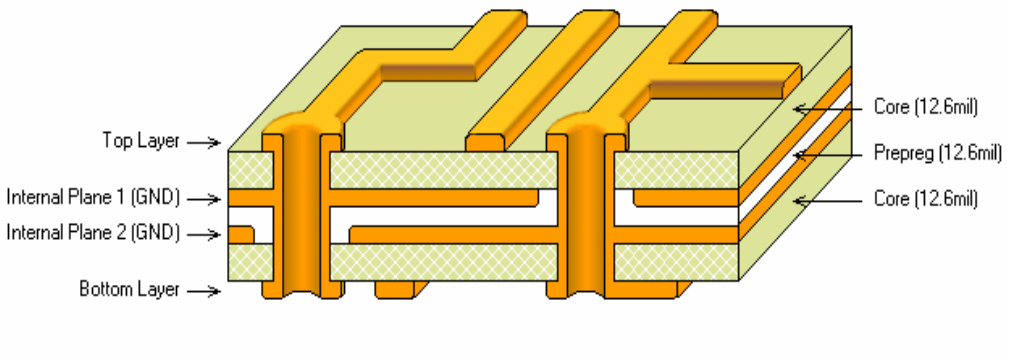
- LCD Panel Interface
  - LCD panel included to facilitate debugging.
  - LCD Panel without controller
    - MSP430 F16X series does not have onboard LCD controller
  - LCD Panel with controller
    - GPIO based
    - 7 GPIO pins of MSP 430 used for this interface

PCB Design – Design Issues

- Provision of external power supply adapter
- Subsequent Voltage regulation circuit
- Integration of temperature sensor

Implementation - PCB Design

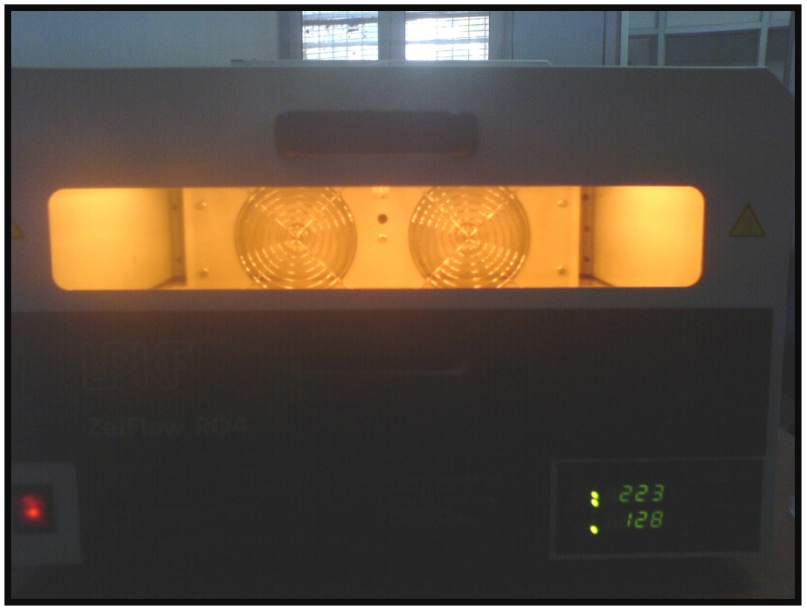


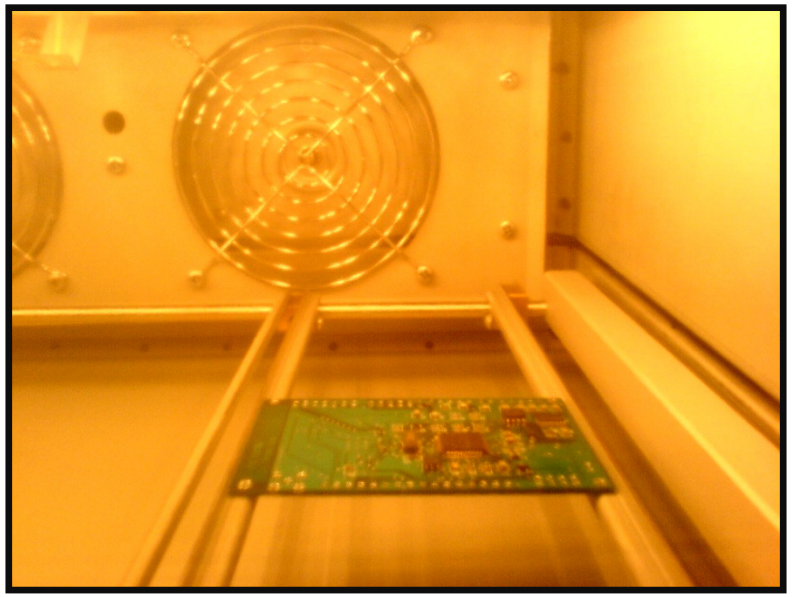


- PCB designed in Protel DXP.
- 4 layer PCB
- Size: 2” x 3”
- Design ready by Dec 06

Implementation

- Manufacture and Assembly
  - 10 PCBs manufactured by CSIL, Gandhinagar
  - Boards sent for fabrication in Jan 07 and received in Mar 07
  - One board soldered in 4i lab at IITK, 4 at Bangalore





Software Integration

- Steps involved while porting TinyOS to KMote
  - Create a new platform directory
    - Create .platform file
    - Contains basic compiler parameter information for the platform

Software Integration

```
push( @includes, qw(

%T/chips/cc2420
%T/chips/msp430
%T/chips/msp430/adc12
%T/chips/msp430/dma
%T/chips/msp430/pins
%T/chips/msp430/timer
%T/chips/msp430/usart
%T/chips/msp430/sensors
%T/lib/timer
%T/lib/serial
) );

@opts = qw(

 -gcc=msp430-gcc
-mmcu=msp430x1611
-fnesc-target=msp430
-fnesc- scheduler=TinySchedulerC,TinySchedulerC.TaskBasic,TaskBasic,TaskBasic,
    runTask,postTask
 );
```

Example Code

Software Integration

- **Steps involved while porting TinyOS to KMote**
  - **Create hardware.h file that defines platform-specific constants, pin names, or also include other "external" header files**

```
#ifndef _H_hardware_h
#define _H_hardware_h
#include "msp430hardware.h"

// LEDs

TOSH_ASSIGN_PIN(RED_LED, 5, 4); TOSH_ASSIGN_PIN(GREEN_LED, 5, 5);
TOSH_ASSIGN_PIN(YELLOW_LED, 5, 6);

#endif
```

Software Integration

- Steps involved while porting TinyOS to KMote
  - Setup the build environment
  - Define the platform in the TinyOS build system
  - Edit make file such that the new platform is recognised

PCB Design – Lessons Learnt

- Steep Learning curve of Protel
- Experience of some design required
- Design of accelerometer boards helpful
- Integrating radio chip and antenna quite challenging even when reference circuit available

PCB Design – Lessons Learnt

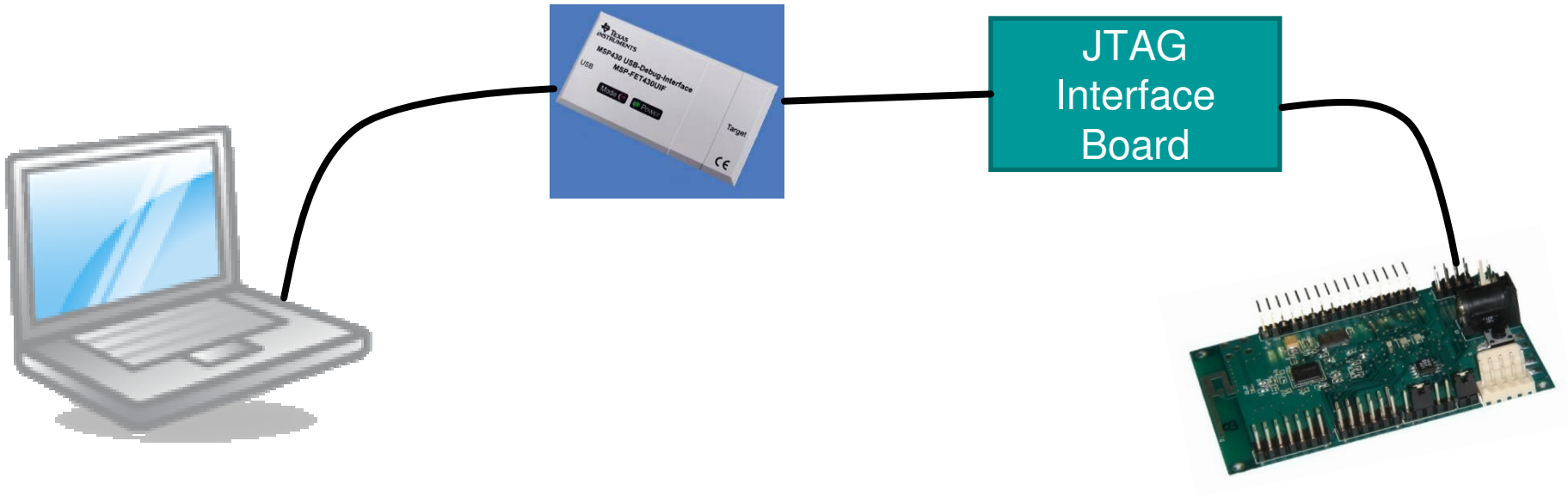
- Placement of components most critical to routing electrical connections
- Ensure there are no routing errors. One such routing error occurred in our design
- Usage of correct power supply.
  - Commercial adapters not reliable
  - Better to test the platform on battery
  - Design a precise power supply adapter

Testing Methodology

- Prepare JTAG interface
- Test Microcontroller
- Test RF Circuitry
- Check RF Ranges
- Check Power Consumption

Testing Methodology

- Prepare JTAG interface board
- Load a program through JTAG interface using gdb to blink the LEDs

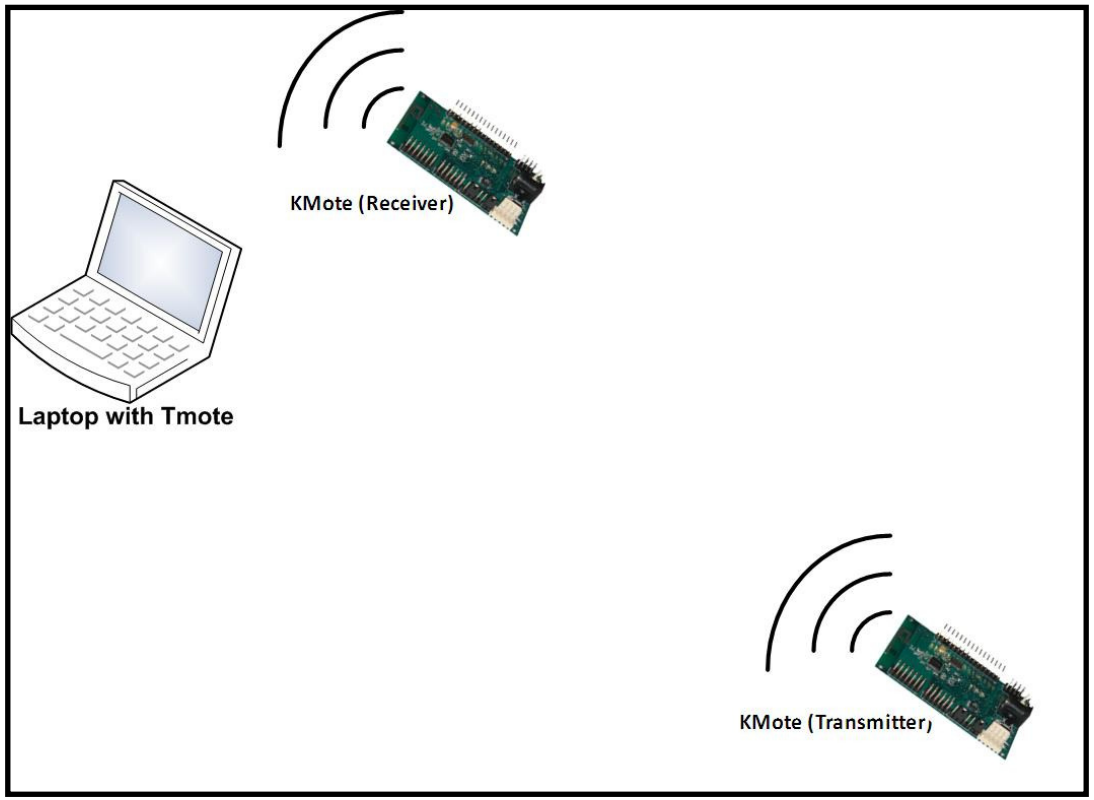


Testing Methodology

- RF Circuitry Test
- Load program through JTAG as earlier
- Program to transmit last 3 bits of a counter
- On reception, blink LEDs corresponding to the received value and transmit receiver's counter value

Testing Methodology

- RF Range Test
- 3 Motes
  - 2 Kmotes
  - 1 Tmote
- 255 Pkts transmitted at 20 ms
- Receiver stores RSSI and S.No of each pkt
- It then relays this data to Tmote
- Tmote displays the data on PC



Experiment Setup

RF Range Measurements

- Results – Comparison of Ranges with Tmote

Environment	RSSI (Avg) dBm	Pkt Error Rate (%)	Ranges	
			KMote	Tmote
Road	- 84.1	0%	150 m	70 m
Airstrip	- 84.0	0%	220 m	130 m

- KMote gave better ranges than Tmote

Testing Methodology

- Power Consumption Test
- Connect an ammeter in series
- Power consumption similar to Tmote

Measured Parameter	KMote	Tmote
Current Consumption: MCU on, Radio TX	19.3 mA	19.5 mA (Nominal) 21 mA (Max)
Current Consumption: MCU on, Radio RX	21.1 mA	21.8 mA (Nominal) 23 mA (Max)
Current Consumption: MCU on, Radio Off	1.5 mA	1.8 mA (Nominal) 2.4 mA (Max)
Current Consumption: MCU standby	6.7 µ A	5.1 µ A (Nominal) 21 µ A (Max)

Cost Analysis

- BOM Cost of KMote

Description	Cost
Cost of Manufacturing a PCB	Rs 100 (2.40 USD)
Cost of Components	
Microcontroller (MSP430)	Rs 372 (8.65 USD)
Radio (CC2420)	Rs 210 (4.86 USD)
Voltage Regulator(MCP3302)	Rs 13 (0.3 USD)
Flash (STM25P80)	Rs 155 (3.62USD)
Serial ID Generator (DS2411)	Rs 23 (0.54 USD)
RS232 Trans-receiver (MAX3380)	Rs 120 (2.78 USD)
Temperature Sensor (TMP36FSZ)	Rs 17 (0.4 USD)
16 Mhz Crystal (for Radio)	Rs 185 (4.3 USD)
Other Passive SMD Components	Rs 430 (10 USD)
Total	**Rs 1625 (37.85 USD)**

Cost Analysis

- Comparison of cost

Serial Number	Product	Cost
(a)	Tmote	Rs 5590 (130.00 USD(without sensors)) Rs 7740 (180 USD (with sensors))
(b)	Tiny Node	Rs 7740 (180.00 USD)
(c)	KMote	Rs 1625 (37.85 USD) (BOM cost only)

- BOM cost about 1/4th cost of Tmote/TinyNode

Conclusions

- An effort to design and implement a mote
- Aimed towards building a low cost, working prototype
- Enable easy availability of motes to research/educational institutions in India
- Facilitate growth of WSN applications that solve problems specific to India
- To dos
  - Software module to interface LCD Panel
  - Driver for serial interface

Future Scope

- Explore feasibility of using a 32-bit microcontroller (Atmel AVR32 UC3 family)
- Port a tiny version of Linux thereby increasing mote's flexibility
- Incorporate a power amplifier circuit to boost output power
- Design of an onboard directional antenna
- Use of an integrated microcontroller-radio chip

Thank You !