Chapter-2

System Design
2.1 Introduction

Many hardware as well as software components required to develop the embedded system. The functional requirements needed for developing the embedded system are Hardware components and development system requirements. Figure (2.1) shows the architectural design of our embedded system. The Schematics is divided into two parts,

1. A Remote Area Sub Station
2. A Server Station

A Remote area substation consists of web enabled networked microcontroller i.e. TINI microcontroller and simple microcontroller interfaced with different sensors for sensing environmental parameters like Temperature, Humidity and wind direction using Weather Cock.

![Figure (2.1) The Architectural Design Of Distributed Embedded System for Weather Station](image)

These sensors are placed at different locations over the remote substation. For the purpose of remote monitoring, these remote substations must necessarily be connected to server station. To establish the communication between remote substation and server station, it is the need of the system to bridge the gap between remote station and the internet. In this regard the TINI
microcontroller developed by Dallas semiconductor provides the ready solution. The TINI microcontroller board is web enabled so it can be directly connected to internet. The sensors used here are 1-wire sensor. 1-Wire temperature sensor is a semiconductor device. The name is derived from the fact that it only uses 1 wire for communication. Many of such 1 wire sensors can be connected. The operating range of DS18B20 is -55 C to +125 C with an accuracy of ±0-5C. The data which is collected by the sensors is then transferred to the database. Like many of the commercial and agricultural weather stations, 1-Wire offers a basic package with many options to suite particular client needs.

The weather Cock is accessible through the serial port of embedded system, and is used for detecting Wind direction and wind speed.

2.2 Functional Requirements for design & development of Embedded System

The hardware requirements to develop the embedded system are as follows:

1. TINI Module version TINIm 400 from Dallas Semiconductor.
2. TINI's socket board.
4. Weather Cock
5. Base Board.
6. Cables—Straight through serial and network cable.
7. DB9 Male to DB9 Female Converter.
8. +5v power supply.

2.2.1 Design Metrics

TINI microcontroller is capable of adding intelligence to many everyday devices that require small size, low power consumption, affordability and continuous uptime. This platform is standards-based both in its operating environment and in the support it gives to the underlying Internet technology, including protocols such as HTTP, TCP, UDP, IGMP and PPP. TINI acts as a bridge to the Internet as well as local, off-line control TINI supports coprocessor connections with its serial and parallel APIs. The TINI I/O is kept open-ended using the 1-Wire net API to actuate switches and sense a distributed environment. The path to the Internet can be wired or wireless. With the ability to support both wired and wireless networking, TINI is ideally situated
for serving data from embedded devices. DS80C400 also increases system performance with its faster processing & enhanced I/O capabilities.

2.2.1.1 TINI board model-400

The TINI board model 400 (TBM400), is embedded in E-10 socket board. It is an 8051 based high speed microcontroller with 4 DPTR (data Pointer),3 serial ports, ROM less WDT CAN controller, 4 timers/counters,IEEE802.3 Ethernet interface with TCP/IP in ROM. 1-wire net controller,64 IO lines+ address+ data bus,16 interrupt/3 priority levels. It is available in 256 bytes on-chip RAM, 9KB on-chip SRAM, 16/32 bit math coprocessor, An on-chip math accelerator allows the DS80C400 to perform 16- and 32-bit multiplication, division, shifting, and normalization. It supports optional 10 bit stack pointer and addresses up to 16 m bytes external memory. The TINI board module is available in 144 pin SODIMM (Small Outline Dual Inline Memory Module).

2.2.1.2 The E-10 Socket

A socket board’s main function is to provide the physical connectors to interface the TBM400 with other equipment such as an Ethernet network, a serial device or a 1-wire network. It is mostly used to connect necessities such as serial, Ethernet and power. The E-10 socket board is 160mm*120mm, which is shown in Figure (2.2) with TINI board inserted.

![Figure(2.2) E-10 Socket Board With TINI Module Inserted.](image-url)
The E-10 socket board provides following physical connectors.

1. **144–pins SODIMM connector** - The SODIMM connector accepts the TINI board.
2. **9-pin female DB9 connector** - This connector provides connection to a standard PC serial port using a straight through serial cable. This port is usually used for loading runtime environment and bootstrap applications.
3. **9-pin male DB9 connector** - This connector provides serial port for straight through connections to analog modems. Most TINI applications that control serial devices use the DTE port. In this case TINI is the DTE device, replacing the PC or workstation.
4. **RJ45** - the RJ45 connector accepts a standard 10 Base-T –Ethernet cable providing connectivity to an Ethernet network. A straight through cable can be used for connecting TINI directly to the network using a hub. And a crossover cable can be used for connecting TINI to a PC or Workstation.
5. **RJ11** - the RJ11 connector provides access to the 1-Wire network using standard telephone cable.
6. **Power Jack** - The E-10 accepts a regulated +5v DC supply.

### 2.2.1.3 Assembling of the DSTINIm400

1. Slide the DSTINIm400 module into the 144-pin connector of the DSTINIs400 Evaluation Board at a 30º angle into J10 and carefully press down the DSTINIm400 until it snaps into place.
2. Connect the crossover Ethernet cable from your PC to J8, the RJ-45 ethernet connector on the DSTINIs400 socket board.
3. Connect the serial cable to J12, the Loader–Serial 0 of the DSTINIs400 socket board and connect the other end to the COM port on your PC.
4. Plug the power adapter to the power jack of the DSTINIs400 and connect the 5V (+5%) 300mA, center-positive power supply to J2 of the DSTINIs400 socket board and a wall outlet. The power LED (DS1) of the DSTINIs400 module board should light up.
5. Plug the network cable into the Ethernet interface of the TINI Evaluation board.

### 2.2.1.4 Architecture of TINI

The DS80C400 network microcontroller offers the highest integration available in an 8051 device. The peripherals include a 10/100 Ethernet MAC, three serial ports, a CAN 2.0B controller, 1-Wire® Master, and 64 I/O pins.
To enable access to the network, a TCP IPv4/6 network stack and OS are provided in ROM. The network stack supports up to 32 simultaneous TCP connections and can transfer up to 5Mbps through the Ethernet MAC. Its maximum system-clock frequency of 75MHz results in a minimum instruction cycle time of 54ns. Access to large program or data memory areas is simplified with a 24-bit addressing scheme that supports up to 16MB of contiguous memory. To accelerate data transfers between the microcontroller and memory, the DS80C400 provides four data pointers, each of which can be configured to automatically increment or decrement upon
execution of certain data pointer-related instructions. The DS80C400's hardware math accelerator further increases the speed of 32-bit and 16-bit multiply and divide operations as well as high-speed shift, normalization, and accumulate functions.

TINI supports 4 timers and counters. Its function is to keep track of time. It counts the events themselves and generates the baud rates for the serial port. Fig (2.3) shows the block diagram of TINI – DS80C400 module [1].

2.2.1.5 TINI software Architecture

The TINI software architecture is made up of various components, such as TINI shell, task scheduler, TINI operating system, java virtual machine, java communication API, is shown in figure (2.4). The architecture comprises of four main parts, Task scheduler, and Java virtual Machine (JVM), TINI operating system services and the hardware. The task scheduler and java virtual machine along with native methods interacts with TINI operating system services. The operating system services then directly interacts with the hardware. TINI's ROM houses both the native code, which implements the OS, the JVM (Java Virtual Machine), and the Java API classes. A micro-controller timer is used to update a real-time clock every millisecond.

The task scheduler is a simple round-robin scheduler that provides constant 4-millisecond time slices. With the exception of the garbage collector, all tasks driven by the OS are Java applications. TINI includes a small command shell that provides access to remote hosts. This Java application starts during the last phase of system initialization. During construction, slush creates threads that maintain server sockets to listen for and accept client requests for FTP and Telnet connections. After a successful login, the remote user can execute Unix-style shell commands to manipulate the file system, set or get configuration information, and start or stop other Java applications. Slush is also extensible.

TINI OS is at the lowest level of TINI runtime environment. TINI OS is very small and provides a file system, memory, I/O managers, and task scheduling. TINI OS is designed to switch heavyweight tasks. It can switch between multiple executing instances of a Java byte code interpreter, and hence running multiple Java applications. The OS manages multiple native/kernel processes through cooperative multitasking.

The JVM sits on top of TINI OS. In between there is a native interface layer, so TINI OS is probably not written in Java. We can invoke assembly code functions to solve low-level problems from Java applications using this native layer. The I/O library uses the native interface
layer to call functions written in assembler to read inputs and write outputs. It keeps track of all networking activities such as TCP/IP stack and Non-networking I/O such as Serial, CAN, parallel bus etc.

Applications programs written in Java application sit on top of JVM. The JVM supports the Java API and libraries. The main function of memory management is to allocate memory from heap for all processes. It also manages automatic garbage collection and the File system. The file system is situated in SRAM on heap. It consists of linked lists of 512 byte blocks. The files must be made contiguous in order to be interpreted by JVM. The file system is non-volatile. The garbage collectors halt the execution of an application while they traverse the entire heap looking for and freeing any unclaimed memory that causes long execution delays [3]. To load TINI’s Runtime environment, the Java Development Environment, TINI Software Development Kit, and Java Communication API must also be correctly installed on the host machine.

**2.2.1.6 TINI Boot Sequence**

On every power ON the TINI module undergoes some series of actions that load a portion of code found in non-volatile memory. Initially the system starts by reset vector at address 0x0000 containing the starting address of bootstrap loader. This action causes control to jump to that location and execute the code found there. The bootstrap loader can perform two separate actions depending on whether the reset was triggered by power being applied or an external reset was
issued by bringing the RST low. If the power on reset has occurred after the power has been applied and reached a minimum stable level, the boot loader transfers control to the runtime environment. The figure (2.5) shows the boot sequence of TINI [4].

![Diagram of boot sequence of TINI](image)

Figure (2.5) Boot Sequence of TINI

On the other hand if the reset was caused by an external reset, the bootstrap loader initializes the Serial 0 port and waits for a pre-set sequence of bits to arrive. On receipt of this signal, a small command interpreter shell is run that allows a user to reload the flash RAM by downloading from a host machine over the serial port.
2.2 Sensing Techniques

Sensors are the transducers which convert some physical parameter to electrical parameter current or voltage in analog form. Sensors usually output one of two types of signal namely- Analog signal or a discrete i.e. Digital signal. To read the analog signal, we need to convert them into digital using analog to digital convertors. The translation produces some sort of output value that the Microcontroller can use. Whereas in case of digital signals we directly get digital data. Hence we do not need any calibration. If we have all type of sensors in 1-Wire, we do not need to think of how the conversion is being done. But in case of analog sensors, the conversion is not always linear so now-a-days digital sensors are preferred over analog ones.

Microcontrollers usually deal with discrete or digital signals hence Analog signals are to be converted to digital one using a device called Analog to Digital Converter (ADC). Depending upon the type of signal produced the following are the types of sensors used for sensing the parameters [5].

2.2.1 Analog temperature sensors

Analog sensors are fall in following categories. Such as thermocouples, Thermistor and RTD’s. All these are used for temperature monitoring. Thermocouples are inexpensive, rugged, and have a fast response time but are less accurate and the least stable and sensitive. Thermocouples also read only relative temperature difference between the tip and the leads while RTD’s and Thermistor read absolute temperature.

RTD’s are the best choice for repeatability. These are most stable and accurate. As they have a slow response time they have a low amount of self heating. RTDs offer several advantages over a Thermocouple or Thermistor Sensor such as RTD’s supports a wide temperature range -50 to 500°C for thin-film and -200 to 850°C for wire-wound. It offers good accuracy better than thermocouples. These sensors provide good interchangeability and long-term stability. Hence the use of RTD’s is more instead of a thermocouple or Thermistor sensor. With a temperature range up to 850°C, RTDs can be used in all but the highest-temperature industrial processes. Each type of temperature sensor has a particular set of conditions for which it is best suited [6].

Each of these sensor technologies cater to specific temperature ranges and environmental conditions. The sensor’s temperature range, ruggedness, and sensitivity are just a few
characteristics that are used to determine whether or not the device will satisfy the requirements of the application. No one temperature sensor is right for all applications.

2.2.2 Digital Sensor
There are many different types of digital input sensors. Many of them are wired in the same form, which uses a pull-up resistor to force the line high, and to limit the amount of current that can flow [7]. Digital temperature sensors can flexibly reduce their current consumption. This is particularly useful in battery operated applications where minimal power consumption is important. The user can also program temperature limits (low, high) where an alarm is required.

In any embedded system data is moved between RAM and CPU. There are many methods and techniques for data transfer. Each technique having its own pros and cons. So different data transfer technique is used in different situations. Mainly there are two data transfer modes – Parallel data transfer and serial data transfer [8].

2.3 Parallel Transfer
In this mode a number of bits are transferred at a time. Thus they require as much electrical line as the number of bits to be transferred at once. This method is fast but disadvantage is more number of lines. So they are basically used when the transmitter and receiver are physically close to each other for a long time. E.g. CPU and RAM.

2.4.2.1 Serial Transfer
In this mode only one bit is transferred at a time. To transfer 8bits, 8 cycles are required. So these require less number of physical lines but more cycles. Serial Communication requires coordination between the sender and receiver. There are two ways to synchronize the two ends of the communication those are asynchronous and synchronous communication.

2.4.2.2 Asynchronous serial communications
Asynchronous communication is a communication in which the sender and receiver are not concurrently engaged in communication [9]. Asynchronous protocol sends the signal prior to each byte. Asynchronous communication is the used in personal computer industry, because it is easier to implement. And the bytes can be sent whenever they are ready.

2.4.2.3 RS232
The RS-232 is a standard for serial binary single-ended data and control signals connecting between a Data Terminal Equipment and Data Circuit-terminating Equipment. It is commonly used in computer serial ports. The standard defines the electrical characteristics and timing of
signals. A minimal “3-wire” RS-232 connection consisting only of transmits data, receives data and ground, and is commonly used when the full facilities of RS-232 are not required. A two-wire connection data and ground can be used if the data flow is one way. When only hardware flow control is required in addition to two-way data, two lines may be added and used as 5-wire version.

2.4.2.4 Synchronous Serial Communications
The sender and receiver must synchronize with one another before data is sent. In synchronous transmission, the stream of data to be transferred is encoded and sent on one line. This can be achieved using following protocols.

2.4.2.5 4-Wire –SPI
SPI is Serial Peripheral Interface. This is a bidirectional full duplex four wire serial bus. The four wires are a clock, a data in, a data out, a select signal. The full-duplex capability makes SPI very simple and efficient for single master/single slave applications. Some devices implement an efficient, high-speed data stream for applications.

2.4.2.6 3-Wire-Microwire
Micro wire is a serial interface designed by National Semiconductor. It is a three wire interface. This interface is similar to the SPI interface. Many microprocessors with SPI interface may be used to connect Micro wire memories and peripheral devices.

2.4.2.7 2-Wire - I\(^2\)C Bus
This is a bidirectional two wire serial bus. The I2C bus physically consists of 2 active wires and a ground connection. The active wires, called SDA and SCL, are both bi-directional. SDA is the Serial Data line, and SCL is the Serial Clock line. A device has unique 7 or 10 bit address[10]. Every device hooked up to the bus has its own unique address. Each of these chips can act as a receiver and/or transmitter, depending on the functionality.

2.4.2.8 1-Wire Bus
1-wire communication protocol is synchronous serial communication protocol. 1-wire devices provide network connectivity to entities that lack the ability to communicate with the outside world. An object connected to a 1-wire chip is capable of joining a 1-wire network. The network of devices is defined with an open drain master/slave multidrain architecture. It uses a register pull –up to a 5v power supply at the master. The network of devices is called as a 1-wire net. Sometimes it is called as MicroLAN. In a 1-wire network one or more devices are connected
which are uniquely addressable that share a single conductor for communication and power. To establish the connections a flat modular cable for very short distance is used. Or for long distance twisted pair CAT cable is used.

For the development weather monitoring station we have used 1-wire protocol. It provides easy and cheap solution. As 1-wire sensors communicates over a 1-Wire bus using only one data line for communication with a central microcontroller. The parasite power can be given to the 1-wire network. This can be done by connecting the communication line of the sensor to the power line of the microcontroller. Parasite powering is most useful in remote temperature monitoring systems.

2.5 The Temperature Sensor

To establish communication between the sensors and microcontroller we have used the 1-wire communication protocol. To use this protocol we need such a sensor that supports the 1-wire communication. The DS18B20 digital thermometer provides the solution for this. Hence we have used the 1-wire network of DS18B20 temperature sensors. The DS18B20 sensor provides 9-bit to 12-bit Celsius temperature measurements and has an alarm function with nonvolatile user-programmable upper and lower trigger points. It has an operating temperature range of -55°C to +125°C and is accurate to ±0.5°C over the range of -10°C to +85°C. The DS1820 measures temperature by counting the number of clock cycles that an oscillator with a low temperature coefficient goes through during a gate period determined by a high temperature coefficient oscillator. The counter is preset with a base count that corresponds to -55°C. If the counter reaches zero before the gate period is over, the temperature register, which is also preset to the -55°C value, is incremented, indicating that the temperature is higher than -55°C.

In addition, the DS18B20 can derive power directly from the data line “parasite power”, this eliminate the need for an external power supply. This has been shown in the schematics for 1-wire interface in Appendix I. Each DS18B20 has a unique 64-bit serial code, which allows multiple DS18B20s to function on the same 1-Wire bus. Thus, by using TINI microcontroller we have controlled many DS18B20s distributed over a large area [11].

Another feature of DS18B20 temperature sensor is that it does not need external power. It can use parasite power. The DS18B20 uses a 1-Wire communication protocol to ensure data integrity. Several signal types are defined by this protocol: reset pulse, presence pulse, write 0,
All communication with the DS18B20 begins with an initialization sequence that consists of a reset pulse from the master followed by a presence pulse from the DS18B20. When the DS18B20 sends the presence pulse in response to the reset, it is indicating to the master that it is on the bus and ready to operate. During the initialization sequence the bus master transmits (TX) the reset pulse by pulling the 1-Wire bus low for a minimum of 480µs. The bus master then releases the bus and goes into receive mode (RX). When the bus is released, the 5kΩ pull up resistor pulls the 1-Wire bus high. When the DS18B20 detects this rising edge, it waits 15µs to 60µs and then transmits a presence pulse by pulling the 1-Wire bus low for 60µs to 240µs.

2.6 The Humidity Sensor
Humidity measurement is the more difficult problems in basic meteorology [12]. Relative humidity is a function of temperature and absolute moisture content, so small temperature variations will affect the relative humidity. Generally Hygrometers are used for humidity monitoring. For humidity measurement we have used HC3223 sensor. This sensor proposes dual functionality to monitor humidity and temperature. The sensor is mostly used where reliable and accurate measurement is needed. It is cost effective and easy to connect. We can connect this sensor directly to microcontroller; hence we have interfaced this sensor to 89C51RD2
microcontroller. This microcontroller along with the other sensors and Base Board is interfaced with the TINI microcontroller.

![Temperature & Humidity Sensor](image)

**Figure (2.7) Temperature & Humidity Sensor**

In respect to humidity, the sensor gives fast response time. It is highly resistant to chemicals. The results of humidity are not affected by water immersion. The operating range of humidity is 0 to 99% RH.

With respect to temperature monitoring, the sensor is a high quality Thermistor. It is high sensitivity temperature sensor. The operating range of the sensor is -30°C to 80°C.

### 2.7 Weather Cock

Eight reed switches have been used for detecting eight main directions. The **reed switch** is an electrical switch operated by an applied magnetic field. It was invented at Bell Telephone Laboratories in 1936 by W. B. Ellwood. It consists of a pair of contacts on ferrous metal reeds in a hermetically sealed glass envelope. The contacts may be normally open, and closed when a magnetic field is present, or normally closed and opening when a magnetic field is applied. The switch may be actuated by a coil, making a reed relay, or by bringing a magnet near to the switch. Once the magnet is pulled away from the switch, the reed switch will go back to its original position. Reed switches are operated by a magnetic field, via a magnet or a current carrying coil. When the field is removed the switch reverts to its previous state. Operation by a magnet can be achieved in a large variety of ways, either moving the magnet towards and away from the reed either perpendicularly, or parallel to the glass.

For the development of our weather cock we have used a magnet. The magnet is placed on a cock, which is fixed on a small ball bearing for free movement, in such a way that at any given time it is over one of the reed switches. The closure of the reed switch gives information about wind direction.
The fly wheel placed at the tail of the cock is fitted with hall sensor for wind speed information. These signals, 8 from reed switches and one from hall sensor are fed to ports of embedded system. Number of pulses received per second is converted to get wind speed in kmph [13, 14, & 15].

2.8 Base Board

The base board is the heart of the system and is used for displaying the parameter at local substation. The board is developed using Philips 89C51RD2 IC [16], Liquid Crystal Display(LCD) and all the sensors. The LCD used serves two purposes i.e. parameters display and displaying of message received from remote user. The use of 1-wire temperature sensor reduces the glue logic. The humidity sensor also needs only one port line as it gives frequency corresponding to relative humidity. Only weather cock requires full port (8 lines) to read eight directions.

This device is a Single-Chip 8-Bit Microcontroller manufactured in an advanced CMOS process and is a derivative of the 80C51 microcontroller family. The instruction set is 100% compatible with the 80C51 instruction set. It has 1kbytes of RAM, 64kbytes of flash memory. In addition to four 8-bit I/O ports, three 16-bit timer/event counters, a multi-source, four-priority-level, nested interrupt structure, an enhanced UART and on-chip oscillator and timing circuits & I²C Protocol Support. The 89C51RD2 controller has got ISP (In System Programming) facility, so we can download the firmware without removing the microcontroller from base board. This
makes the development very fast and reduces the chances of damaging the device due to multiple insertions and removals. The same serial port is used for communicating to TINI.

2.9 Liquid Crystal Display
The 20*4 column LCD is used for instant local feedback of the weather data. And the same can be used to pass the message for end user. This can be installed at remote weather station location. The use of LCD is optional. The LCD display is interfaced with TINI microcontroller through the user serial port. If any problem occurs at the remote location, the same can be noticed from web browser, used to access the weather information. If problem occurs in hardware then the authorized person will send the message to the LCD. This message will be sent through the web browser to TINI and it displayed on LCD at remote location. The remote location person who has given the training about the weather station installed will follow the instructions accordingly. If the problem occurs is in software, can be solved through server.

2.10 Cables
We have used Serial cable to communicate with TINI through host machine. A straight through Ethernet cable is used to connect the system to internet.

2.10.1 Serial Cable
The serial port sends and receives bytes of information one bit at a time. This is slower than parallel communication, which allows the transmission of an entire byte at once; it is simpler and can be used over longer distances. There are several different kinds of serial cables. The two most common types are null-modem cables (Cross Cable) and standard (“straight”) RS-232 cables [17].

2.10.1.1 Crossed Serial Cable
If it is not possible to connect the TINI board directly to hub we can connect the same to PC using crossed Ethernet cable. A crossed Ethernet cable is also known as a null modem. A null-modem cable passes some signals, such as “Signal Ground”, straight through, but switches other signals. For example, the “Transmitted Data” pin on one end goes to the “Received Data” pin on the other end [18].

2.10.1.2 Straight Through Serial Cable
For connecting TINI to internet, we are using two types of cable. If we are directly connecting TINI to router or hub we need a straight through serial cable for this purpose. A standard serial
cable passes all of the RS-232C signals straight through. That is, the “Transmitted Data” pin on one end of the cable goes to the “Transmitted Data” pin on the other end [19].

2.11 Development System Requirements
Since all of the required tools have been developed in Java, TINI applications can be developed on any of the following operating systems,

1. Any Win32 OS (Windows 95, 98, NT, 2000)
2. Linux
3. Solaris

The development system must have one RS-232 comport, with 115200 baud rate. One RS-232c serial cable DB9 male to DB9 female. A crossover Ethernet cable to connect the TINI to PC. And a straight through Ethernet cable is used for connecting TINI directly to router or hub.

2.12 Conclusion
The chapter dealt with the hardware aspects of for the development of weather monitoring system. The hardware components are chosen by considering the requirements of the system. The sensors are interfaces with the legacy hardware, which is in turn interfaced to TINI microcontroller. The serial port is used to establish communication between microcontroller and Base Board. The hardware components are assembled together to develop a distributed embedded system. The system is accessible over the network through Ethernet card. Application software is needed to develop to achieve the required system functionality.
2.13 References

1. TINI Design and Implementation Developers guide –By Don Loomis


