International Journal of Computing and Corporate Research
IJCCR India

DESIGN, DEVELOPMENT AND ANALYSIS OF LOW COST, LOW POWER SCADA SYSTEM FOR SUGAR INDUSTRY BASED ON ATMEL - CAN NETWORKING MCU 89C51CC03: A CASE STUDY

SWATI JAYADE

ABSTRACT:

The paper describes the design, development and analysis of low cost and low power SCADA system for sugar industries in Vidarbha region of Maharashtra. In fact, most of the sugar factories in the Vidarbha region of Maharashtra are 20-25 years old and still using manual settings of process control parameters. Hence, the quality of the end product, i.e. sugar, is not satisfactory and hence profit margin is low. PLC based SCADA is too costly as most of the sugar industries are running in loss and about to be closed. The case study of the employability of the low cost SCADA based on the recent microcontroller was carried out by the authors. The results are very promising as the quality of the end product is found to be good after precise control of the process control parameters using newly designed SCADA system.

Keywords: SCADA, Microcontrollers, Sugar Industries, Microcontroller based Instrumentation, Atmel, CAN Networking CPU, Low cost design

1. INTRODUCTION

SCADA is an acronym for Supervisory Control and Data Acquisition. SCADA systems are used to monitor and control a plant or equipment in industries such as telecommunications, water and waste control, energy, oil and gas refining and transportation. SCADA systems can be relatively simple, such as one that monitors environmental conditions of a small office building, or incredibly complex, such as a system that monitors all the activity in a nuclear power plant or the activity of a municipal water system. SCADA systems were first used in the 1960s. It generally refers to industrial control systems: computer systems that monitor and control industrial, infrastructure, or facility-based processes, as described below:

- Industrial processes include those of manufacturing, production, power generation, fabrication, and refining, and may run in continuous, batch, repetitive, or discrete modes.
- Infrastructure processes may be public or private, and include water treatment and distribution, wastewater collection and treatment, oil and gas pipelines, electrical power transmission and distribution, Wind farms, civil defense siren systems, and large communication systems.
- Facility processes occur both in public facilities and private ones, including buildings, airports, ships, and space stations. They monitor and control HVAC, access, and energy consumption, access, and energy consumption.

2. COMMON SYSTEM COMPONENTS

A SCADA System usually consists of the following subsystems:

- A Human-Machine Interface or HMI is the apparatus which presents process data to a human operator, and through this, the human operator monitors and controls the process.
- A supervisory (computer) system, gathering (acquiring) data on the process and sending commands (control) to the process.
- Remote Terminal Units (RTUs) connecting to sensors in the process, converting sensor signals to digital data and sending digital data to the supervisory system.

- Programmable Logic Controller (PLCs) used as field devices because they are more economical, versatile, flexible, and configurable than special-purpose RTUs.
- Communication infrastructure connecting the supervisory system to the Remote Terminal Units.

In most of the cases mentioned above, it is found that the cost is too high. It requires skilled man power to operate. Thirdly, annual maintenance cost is also high. Hence, low cost SCADA based on recent microcontroller is found to be the need of time. It is designed, developed and its use is analyzed by the authors for the sugar industry.

3. SYSTEMS CONCEPTS: PLC BASED SCADA

The term SCADA usually refers to centralized systems which monitor and control entire sites, or complexes of systems spread out over large areas (anything between an industrial plant and a country). Most control actions are performed automatically by Remote Terminal Units ("RTUs") or by programmable logic controllers ("PLCs"). PLC may control the flow of cooling water through part of an industrial process, but the SCADA system may allow operators to change the set points for the flow, and thus enable alarm conditions at different set points. The feedback control loop passes through the RTU or PLC, while the SCADA system monitors the overall performance of the loop.

Data acquisition begins at the RTU or PLC level and includes meter readings and equipment status reports that are communicated to SCADA as required. Data is then compiled and formatted in such a way that a control room operator using the Human Machine Interface (HMI) can take supervisory decisions to adjust or override normal RTU (PLC) controls. SCADA systems typically implement a distributed database, commonly referred to as a tag database, which contains data elements called tags or points. A point represents a single input or output value monitored or controlled by the system.

A Human-Machine Interface or HMI is the apparatus which presents process data to a human operator for final process control. An HMI is usually linked to the SCADA system's databases and software programs. The HMI system usually presents the information to the operating personnel graphically, in the form of a mimic diagram. This means that the operator can see a schematic representation of the plant being controlled. The operator can then switch the pump off. The HMI software will show the flow rate of the fluid in the pipe decrease in real time. Mimic diagrams may consist of line graphics and schematic symbols to represent process elements, or may consist of digital photographs of the process equipment overlain with animated symbols.

An important part of most SCADA implementations is alarm handling. The system monitors whether certain alarm conditions are satisfied, to determine when an alarm event has occurred. Once an alarm event has been detected, one or more actions are taken. In designing SCADA systems, care is needed in coping with a cascade of alarm events occurring in a short time, otherwise the underlying cause may get lost in the noise.

4. SYSTEMS CONCEPTS: MICROCONTROLLER BASED SCADA

PLC based SCADA has got certain disadvantages, such as, very high initial installation cost , need of special maintenance schedules, costly AMC and need of a skilled operator. Above all, the production of the commodity is hampered due to the maintenance schedule. This cannot be tolerated by low-profit and seasonally “ON” old sugar factories in the Vidarbha region. Hence, low cost, low power microcontroller based SCADA can be thought as a suitable solution to this. There is a variety of 8-bit and 16 bit microcontrollers available. Author has selected ATMEL - CAN NETWORKING MCU 89C51CC03**.** Details of which are given below.

5. ATMEL - CAN NETWORKING MCU 89C51CC03

I. Features

• 80C51 Core Architecture

• 256 Bytes of On-chip RAM

• 2048 Bytes of On-chip ERAM

• 64K Bytes of On-chip Flash Memory

– Data Retention: 10 Years at 85℃

– Read/Write Cycle: 100K

• 2K Bytes of On-chip Flash for Bootloader

• 2K Bytes of On-chip EEPROM

Read/Write Cycle: 100K

• Integrated Power Monitor (POR: PFD) To Supervise Internal Power Supply

• 14-sources 4-level Interrupts

• Three 16-bit Timers/Counters

• Full Duplex UART Compatible 80C51

• High-speed Architecture

– In Standard Mode:

40 MHz (Vcc 3V to 5.5V, both Internal and external code execution)

60 MHz (Vcc 4.5V to 5.5V and Internal Code execution only)

– In X2 mode (6 Clocks/machine cycle)

20 MHz (Vcc 3V to 5.5V, both Internal and external code execution)

30 MHz (Vcc 4.5V to 5.5V and Internal Code execution only)

• Five Ports: 32 + 4 Digital I/O Lines

• Five-channel 16-bit PCA with

– PWM (8-bit)

– High-speed Output

– Timer and Edge Capture

• Double Data Pointer

• 21-bit WatchDog Timer (7 Programmable Bits)

• A 10-bit Resolution Analog to Digital Converter (ADC) with 8 Multiplexed Inputs

• SPI Interface, (PLCC52 and VPFP64 packages only)

• Full CAN Controller
– Fully Compliant with CAN Rev 2.0A and 2.0B
– Optimized Structure for Communication Management (Via SFR)
– 15 Independent Message Objects
– Each Message Object Programmable on Transmission or Reception
– Individual Tag and Mask Filters up to 29-bit Identifier/Channel
– 8-byte Cyclic Data Register (FIFO)/Message Object
– 16-bit Status and Control Register/Message Object
– 16-bit Time-Stamping Register/Message Object
– CAN Specification 2.0 Part A or 2.0 Part B Programmable for Each Message Object
– Access to Message Object Control and Data Registers Via SFR
– Programmable Reception Buffer Length Up To 15 Message Objects
– Priority Management of Reception of Hits on Several Message Objects at the Same Time (Basic CAN Feature)
– Priority Management for Transmission
– Message Object Overrun Interrupt
– Supports
– Time Triggered Communication
– Auto baud and Listening Mode
– Programmable Automatic Reply Mode
– 1-Mbit/s Maximum Transfer Rate at 8 MHz (1) Crystal Frequency in X2 Mode
– Readable Error Counters
– Programmable Link to On-chip Timer for Time Stamping and Network Synchronization
– Independent Baud Rate Prescaler
– Data, Remote, Error and Overload Frame Handling

On-chip Emulation Logic (Enhanced Hook System)
• Power Saving Modes
– Idle Mode
– Power-down Mode

• Power Supply: 3 volts to 5.5 volts

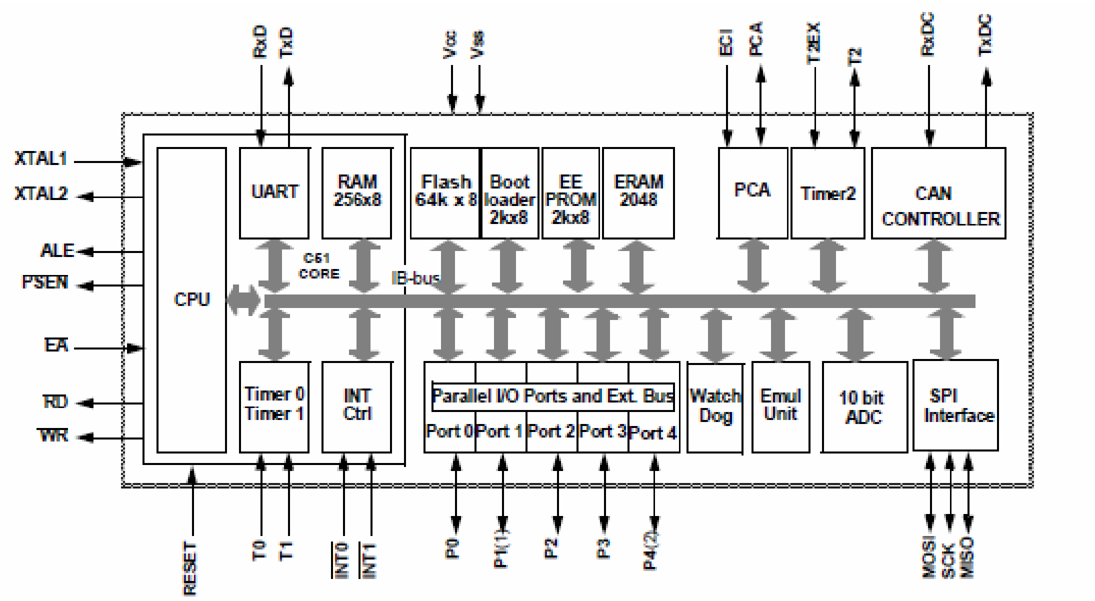
• Temperature Range: Industrial (-40°to +85℃), Au tomotive (-40℃ to +125℃)

• Packages: VQFP44, PLCC44, VQFP64, PLCC52

II. Description

The AT89C51CC03 is a member of the family of 8-bit microcontrollers dedicated to CAN network applications. In X2 mode a maximum external clock rate of 20 MHz reaches a 300 ns cycle time. Besides the full CAN controller AT89C51CC03 provides 64K Bytes of Flash memory including In-System Programming (ISP), 2K Bytes Boot Flash Memory, 2K Bytes EEPROM and 2048 byte ERAM. Primary attention is paid to the reduction of the electro-magnetic emission of AT89C51CC03. Basic block diagram is seen in fig. 1.

III. Block diagram



Notes: 1. 8 analog Inputs/8 Digital I/O
2. 5-Bit I/O Port

Fig. 1 Internal architecture of Atmel 89C51CC03

IV. CAN Controller Description

CAN Protocol

The CAN protocol is an international standard defined in the ISO 11898 for high speed and ISO 11519-2 for low speed.

Principles

CAN is based on a broadcast communication mechanism. This broadcast communication is achieved by using a message oriented transmission protocol. These messages are identified by using a message identifier. Such a message identifier has to be unique within the whole network and it defines not only the content but also the priority of the message.

The priority at which a message is transmitted compared to another less urgent message is specified by the identifier of each message. The priorities are laid down during system design in the form of corresponding binary values and cannot be changed dynamically. The identifier with the lowest binary number has the highest priority. Bus access conflicts are resolved by bit-wise arbitration on the identifiers involved by each node observing the bus level bit for bit. This happens in accordance with the "wired and" mechanism, by which the dominant state overwrites the recessive state. The competition for bus allocation is lost by all nodes with recessive transmission and dominant observation. All the "losers" automatically become receivers of the message with the highest priority and do not re-attempt transmission until the bus is available again.

Message Formats

The CAN protocol supports two message frame formats, the only essential difference being in the length of the identifier. The CAN standard frame, also known as CAN 2.0 A, supports a length of 11 bits for the identifier, and the CAN extended frame, also known as CAN 2.0 B, supports a length of 29 bits for the identifier.

CAN access

The CAN Controller accesses are made through SFR. Several operations are possible by SFR:

- Arithmetic and logic operations, transfers and program control (SFR is accessible by direct addressing).
- 15 independent message objects are implemented; a pagination system manages their accesses.

Any message object can be programmed in a reception buffer block (even non-consecutive buffers). For the reception of defined messages one or several receiver message objects can be masked without participating in the buffer feature. Interrupt is generated when the buffer is full. The frames following the buffer-full interrupt will not be taken into account until at least one of the buffer message objects is re-enabled in reception. Higher priority of a message object for reception or transmission is given to the lower message object number. The programmable 16-bit Timer (CANTIMER) is used to stamp each received and sent message in the CANSTMP register. This timer starts counting as soon as the CAN controller is enabled by the ENA bit in the CANGCON register. The Time Trigger Communication (TTC) protocol is supported by the AT89C51CC03. Fig. 2 depicts CAN controller architecture.

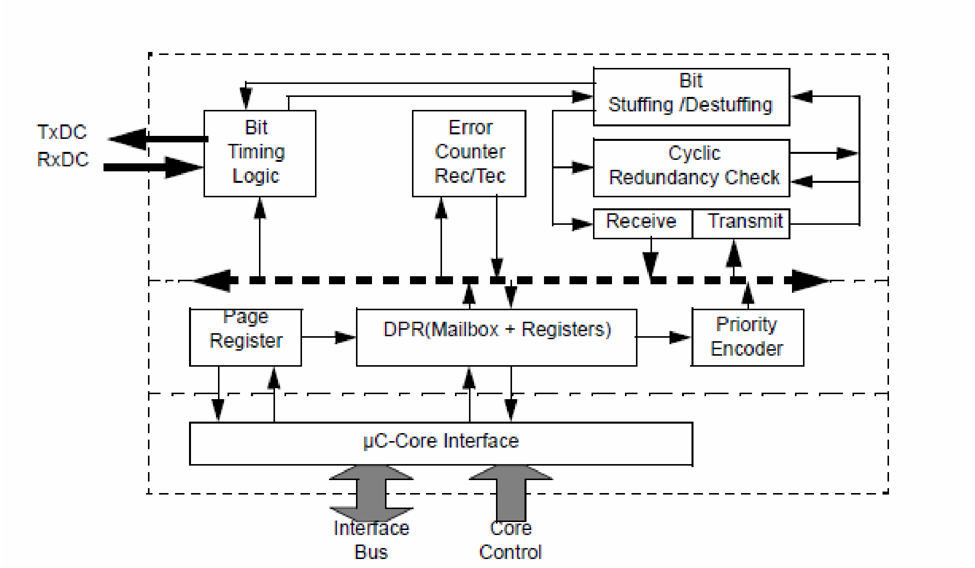


Fig. 2 CAN Controller Block Diagram

6. SCADA OVERVIEW

SCADA systems are used to monitor and control a plant or equipment in industries such as telecommunications, water and waste control, energy, oil and gas refining and transportation. These systems encompass the transfer of data between a SCADA central host computer and a number of Remote Terminal Units (RTUs) and/or Programmable Logic Controllers (PLCs), and the central host and the operator terminals. A SCADA system gathers information (such as where a leak on a pipeline has occurred), transfers the information back to a central site, then alerts the home station that a leak has occurred, carrying out necessary analysis and control, such as determining if the leak is critical, and displaying the information in a logical and organized fashion. These systems can be relatively simple, such as one that monitors environmental conditions of a small office building, or very complex, such as a system that monitors all the activity in a nuclear power plant or the activity of a municipal water system. Traditionally, SCADA systems have made use of the Public Switched Network (PSN) for monitoring purposes.

Today many systems are monitored using the infrastructure of the corporate Local Area Network (LAN)/Wide Area Network (WAN). Wireless technologies are now being widely deployed for purposes of monitoring.

SCADA systems consist of:

- One or more field data interface devices, usually RTUs, or PLCs, which interface to field ensign devices and local control switchboxes and valve actuators
- A communications system used to transfer data between field data interface devices and control units and the computers in the SCADA central host. The system can be radio, telephone, cable, satellite, etc., or any combination of these.
- A central host computer server or servers , sometimes called a SCADA Center, master
  station or Master Terminal Unit (MTU)
- A collection of standard and/or custom software [sometimes called Human Machine Interface (HMI) software or Man Machine Interface (MMI) software] systems used to
  provide the SCADA central host and operator terminal application, support the communications system, and monitor and control remotely located field data interface
  devices

I. Field Data Interface Devices

Field data interface devices form the "eyes and ears" of a SCADA system. Devices such as reservoir level meters, water flow meters, valve position transmitters, temperature transmitters, power consumption meters, and pressure meters all provide information that can tell an experienced operator how well a water distribution system is performing. In addition, equipment such as electric valve actuators, motor control switchboards, and electronic chemical dosing facilities can be used to form the "hands" of the SCADA system and assist in automating the process of distributing water. However, before any automation or remote monitoring can be achieved, the information that is passed to and from the field data interface devices must be converted to a form that is compatible with the language of the SCADA system. To achieve this, some form of electronic field data interface is required. RTUs, also known as Remote Telemetry Units, provide this interface. They are primarily used to convert electronic signals received from field interface devices into the language (known as the communication protocol) used to transmit the data over a communication channel.

The instructions for the automation of field data interface devices, such as pump control logic, are usually stored locally. This is largely due to the limited bandwidth typical of communications links between the SCADA central host computer and the field data interface devices. Such instructions are traditionally held within the PLCs, which have in the past been physically separate from RTUs. A PLC is a device used to automate monitoring and control of industrial facilities. It can be used as a stand-alone or in conjunction with a SCADA or other system. PLCs connect directly to field data interface devices and incorporate programmed intelligence in the form of logical procedures that will be executed in the event of certain field conditions. PLCs have their origins in the automation industry and therefore are often used in manufacturing and process plant applications. The need for PLCs to connect to communication channels was not great in these applications, as they often were only required to replace traditional relay logic systems or pneumatic controllers. SCADA systems, on the other hand, have origins in early telemetry applications, where it was only necessary to know basic information from a remote source. The RTUs connected to these systems had no need for control programming because the local control algorithm was held in the relay switching logic.

PLCs have their origins in the automation industry and therefore are often used in manufacturing and process plant applications. The need for PLCs to connect to communication channels was not great in these applications, as they often were only required to replace traditional relay logic systems or pneumatic controllers. SCADA systems, on the other hand, have origins in early telemetry applications, where it was only necessary to know basic information from a remote source. The RTUs connected to these systems had no need for control programming because the local control algorithm was held in the relay switching logic. As PLCs were used more often to replace relay switching logic control systems, telemetry was used more and more with PLCs at the remote sites. It became desirable to influence the program within the PLC through the use of a remote signal. This is in effect the "Supervisory Control" part of the acronym SCADA. Where only a simple local control program was required, it became possible to store this program within the RTU and perform the control within that device. At the same time, traditional PLCs included Communications modules that would allow PLCs to report the state of the control program to a computer plugged into the PLC or to a remote computer via a telephone line. PLC and RTU manufacturers therefore compete for the same market. As a result of these developments, the line between PLCs and RTUs has blurred and the terminology is virtually interchangeable. For the sake of simplicity, the term RTU will be used to refer to a remote field data interface device; however, such a device could include automation programming that traditionally would have been classified as a PLC.

II. Communications Network

The communications network is intended to provide the means by which data can be transferred between the central host computer servers and the field-based RTUs. The Communication Network refers to the equipment needed to transfer data to and from different sites. The medium used can either be cable, telephone or radio. The use of cable is usually implemented in a factory. This is not practical for systems covering large geographical areas because of the high cost of the cables, conduits and the extensive labor in installing them. The use of telephone lines (i.e., leased or dial-up) is a more economical solution for systems with large coverage. The leased line is sed for systems requiring on-line connection with the remote stations. This is expensive since one telephone line will be needed per site. Dial-up lines can be used on systems requiring updates at regular intervals (e.g., hourly updates). Here ordinary telephone lines can be used. The host can dial a particular number of a remote site to get the readings and send commands. Remote sites are usually not accessible by telephone lines. The use of radio offers an economical solution. Radio modems are used to connect the remote sites to the host. An on-line operation can also be implemented on the radio system. For locations where a direct radio link cannot be established, a radio repeater is used to link these sites.

Historically, SCADA networks have been dedicated networks; however, with the increased deployment of office LANs and WANs as a solution for interoffice computer networking, there exists the possibility to integrate SCADA LANs into everyday office computer networks. The foremost advantage of this arrangement is that there is no need to invest in a separate computer network for SCADA operator terminals. In addition, there is an easy path to integrating SCADA data with existing office applications, such as spreadsheets, work management systems, data history databases, Geographic Information System (GIS) systems, and water distribution modeling systems.

III. Central Host Computer

The central host computer or master station is most often a single computer or a network of computer servers that provide a man-machine operator interface to the SCADA system. The computers process the information received from and sent to the RTU sites and present it to human operators in a form that the operators can work with. Operator terminals are connected to the central host computer by a LAN/WAN so that the viewing screens and associated data can be displayed for the operators. Recent SCADA systems are able to offer high resolution computer graphics to display a graphical user interface or mimic screen of the site or water supply network in question. Historically, SCADA vendors offered proprietary hardware, operating systems, and software that was largely incompatible with other vendors' SCADA systems. Expanding the system required a further contract with the original SCADA vendor. Host computer platforms characteristically employed UNIX-based architecture, and the host computer network was physically removed from any office-computing domain. However, with the increased use of the personal computer,

computer networking has become commonplace in the office and as a result, SCADA systems are now available that can network with office-based personal computers. Indeed, many of today's SCADA systems can reside on computer servers that are identical to those servers and computers used for traditional office applications. This has opened a range of possibilities for the linking of SCADA systems to office-based applications such as GIS systems, hydraulic modeling software, drawing management systems, work scheduling systems, and Information databases.

7. SENSORS USED

I. PRECISION CENTIGRADE TEMPERATURE SENSORS

Features

- Calibrated directly in ° Celsius (Centigrade)
- Linear + 10.0 mV/℃ scale factor
- 0.5℃ accuracy guarantee able (at +25℃)
- Rated for full -55° to +150℃ range
- Suitable for remote applications
- Low cost due to wafer-level trimming
- Operates from 4 to 30 volts
- Less than 60 µA current drain
- Low self-heating, 0.08℃ in still air
- Nonlinearity only ±1.4℃ typical
- Low impedance output, 0.1 W for 1 mA load

II. Humidity Sensor (HIH-3610 Series)

Features

- Melded thermoset plastic housing with cover
- Linear voltage output vs %RH
- Low power
- High accuracy
- Fast response time
- Chemically resistant *(output is not disturbed due to the presence of chemicals in the air……)*
- The HIH-3610 Series humidity sensor is designed specifically for high volume OEM (Original Equipment Manufacturer) users
- Direct input to a controller due to sensor's linear voltage output.
- With a typical current draw of only 200 mA, the HIH-3610 Series is ideally suited for low drain, battery operated systems.
- The HIH-3610 Series delivers instrumentation-quality RH (Relative Humidity) sensing performance in a low cost, solderable SIP (Single In-line Package).
- Available in two lead spacing configurations, the RH sensor is a laser trimmed thermoset polymer capacitive sensing element with on-chip integrated signal conditioning.
- The sensing element's multilayer construction provides excellent resistance to application hazards such as wetting, dust, dirt, oils, and common environmental chemicals.

III. SHT10 - Digital Humidity Sensor

- Economic relative humidity sensor for low cost applications.
- RH operating range: 0 – 100% RH
- Temperature operating range: -40 – +125℃
- RH response time: 8 sec

- Output: digital

IV. Pressure Sensor

- A pressure sensor measures pressure, typically of gases or liquids.
- Pressure is an expression of the force required to stop a fluid from expanding, and is usually stated in terms of force per unit area.
- A pressure sensor usually acts as a transducer; it generates a signal as a function of the pressure imposed.

We have preferred one arrangement of geared wheels for the measurement of pressure. It uses two gear wheels out of which one is attached to the pressure valve & other is attached to the lever for linear motion. Linear motion is given to the pot. Hence when pressure valve is rotated then bigger gear wheel moves and then small gear wheel moves hence we get linear motion so that potentiometer resistance is varied.

8. EXPERIMENTATION SET UP

It consisted the hardware which includes AT89C51CC03 microcontroller which is a member of 8-bit microcontrollers family dedicated to CAN network applications. The sensors discussed above are the part of the hardware interfacing. The hardware is tested in the sugar industry in the Vidarbha region and process control readings are taken. Few site photographs are shown where the SCADA readings are taken.

Sugar cane being crushed

Raw juice being boiled

Boiled juice in the cooling tank

9. RESULTS

Conventional meter readings and SCADA meter readings are obtained for clear juice and boiled juice for 15 days at a stretch. Few readings and comparative are produced here.

9.1 Clear Juice DAY 3

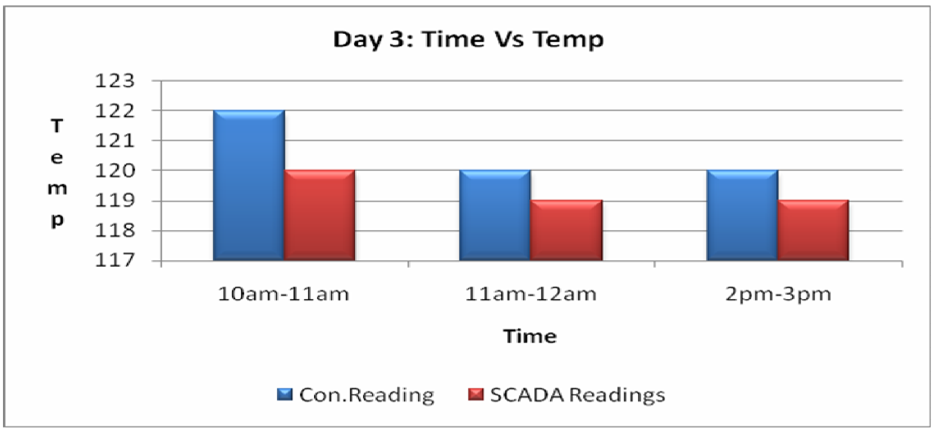
a. Conventional meter Reading on 27th Oct 2010

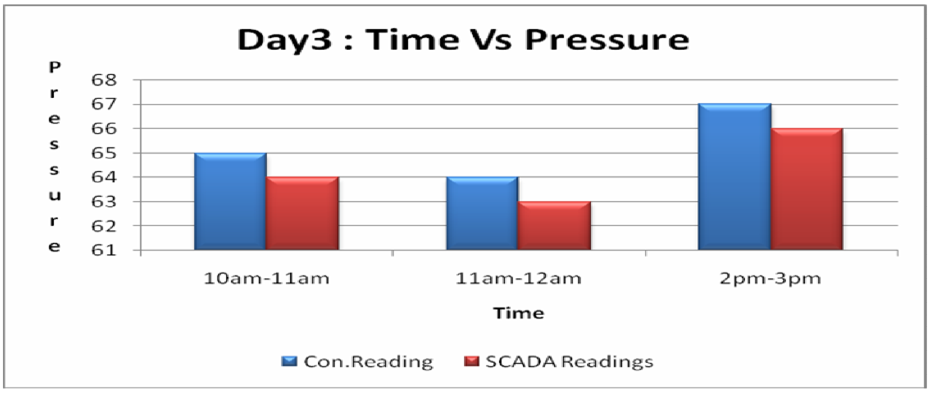
Name of the Process (Time)	**Temp (^{0}C)**	**Humidity (%RH)**	**Pressure (Kg/M^2)**	**Water Level (%)**
Raw Juice (10am-11am)	**122**	**--**	**65**	**--**
Raw Juice (11am-12am)	**120**	**--**	**64**	**--**
Raw Juice (2pm-3pm)	**120**	**--**	**67**	**--**

b. SCADA Reading on 26th Oct 2010

Name of the Process (Time)	Temp (^{0}C)		Humidity (%RH)		Pressure (Kg/M^2)		Water Level (%)	
	L	*H*	*L*	*H*	*L*	*H*	*L*	*H*
Raw Juice (10am-11am)	*119*	*121*	*13*	*44*	*60*	*70*	*00*	*100*
	120		**--**		**64**		**--**	
Raw Juice (11am-12am)	*119*	*121*	*13*	*44*	*60*	*70*	*00*	*100*
	119		**--**		**63**		**--**	
Raw Juice (2pm-3pm)	*119*	*121*	*13*	*44*	*60*	*70*	*00*	*100*
	119		**--**		**66**		**--**	

Comparative Graphs





9.2 Boiled Juice

Day 3

a. Conventional meter Reading on 25th Oct 2010   DAY 3

Name of the Process (Time)	**Temp (^{0}C)**	**Humidity (%RH)**	**Pressure (Kg/M^2)**	**Water Level (%)**
Raw Juice (10am-11am)	**131**	**--**	**92**	**50**
Raw Juice (11am-12am)	**127**	**--**	**94**	**45**
Raw Juice (2pm-3pm)	**128**	**--**	**92**	**35**

b. SCADA Reading on 26th Oct 2010

Name of the Process (Time)	**Temp (^{0}C)**		**Humidity (%RH)**		**Pressure (Kg/M^2)**		**Water Level (%)**	
	L	*H*	*L*	*H*	*L*	*H*	*L*	*H*
Raw Juice (10am-11am)	*125*	*130*	*13*	*44*	*80*	*95*	*00*	*100*
	129		**--**		**93**		**50**	
Raw Juice (11am-12am)	*125*	*130*	*13*	*44*	*80*	*95*	*00*	*100*
	128		**--**		**92**		**44**	
Raw Juice (2pm-3pm)	*125*	*130*	*13*	*44*	*80*	*95*	*00*	*100*
	127		**--**		**94**		**32**	

9.3 Boiled Juice DAY 2

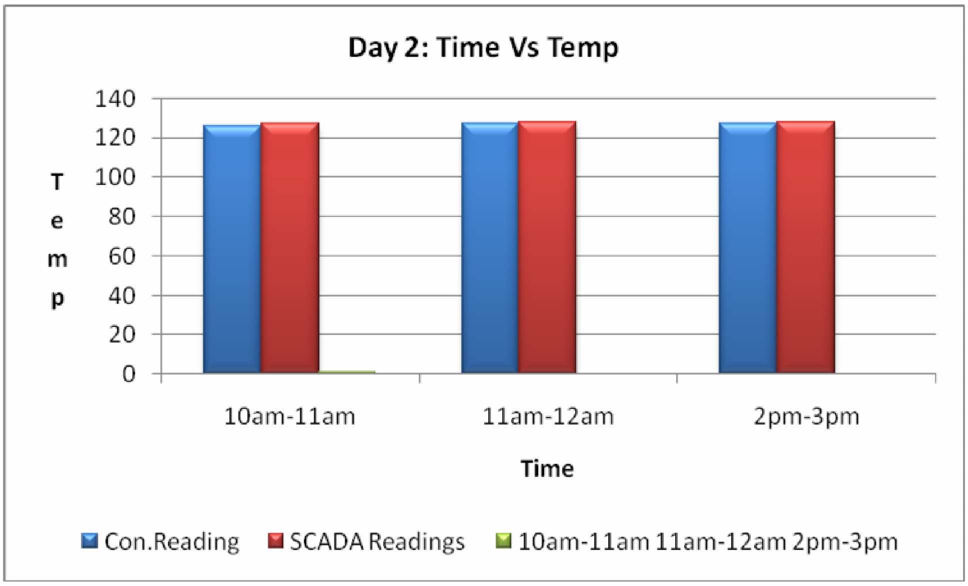
a. Conventional meter Reading on 26th Oct 2010

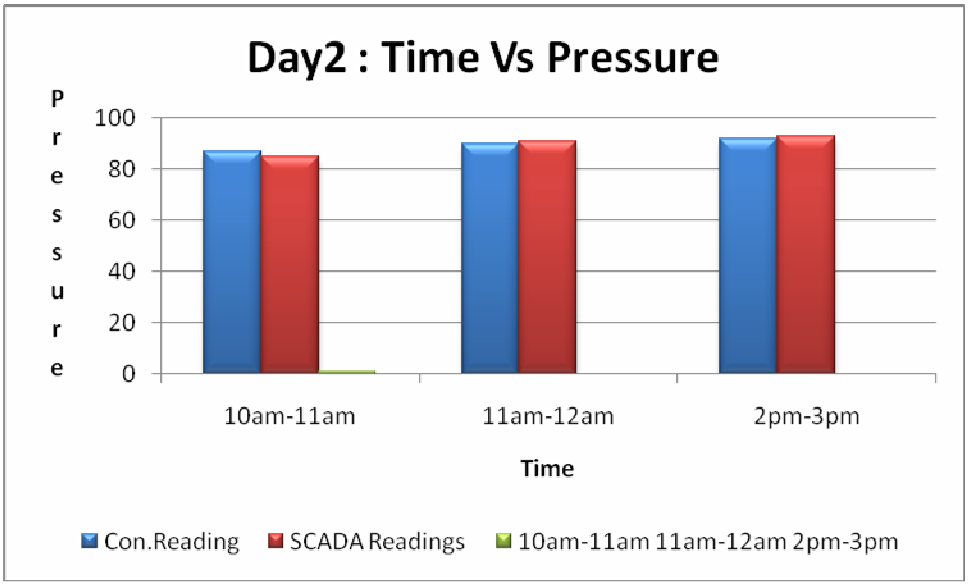
Name of the Process (Time)	**Temp (^{0}C)**	**Humidity (%RH)**	**Pressure (Kg/M^2)**	**Water Level (%)**
Raw Juice (10am-11am)	**128**	**--**	**91**	**25**
Raw Juice (11am-12am)	**129**	**--**	**93**	**25**
Raw Juice (2pm-3pm)	**130**	**--**	**93**	**20**

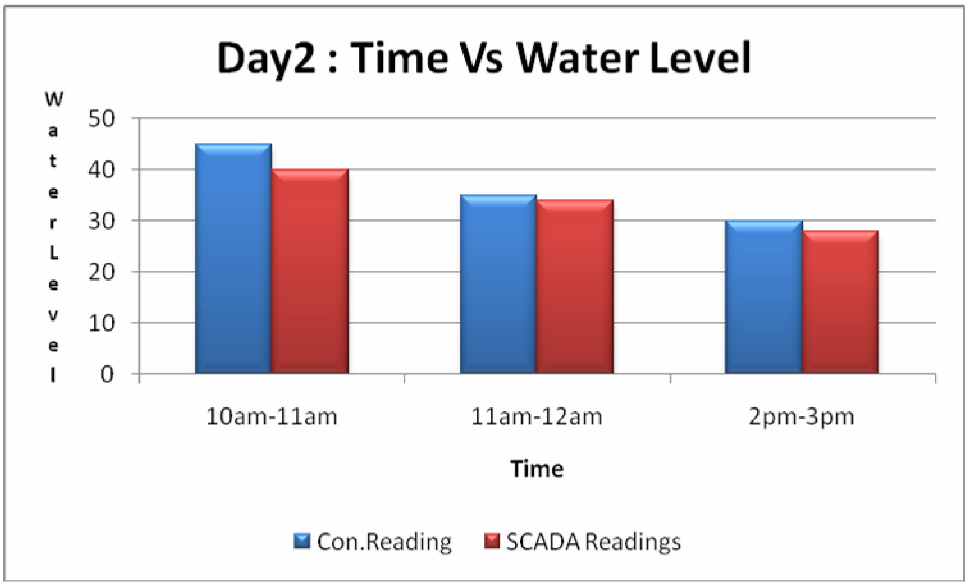
b. SCADA Reading on 26th Oct 2010

Name of the Process (Time)	**Temp (^{0}C)**		**Humidity (%RH)**		**Pressure (Kg/M^2)**		**Water Level (%)**	
	L	*H*	*L*	*H*	*L*	*H*	*L*	*H*
Raw Juice (10am-11am)	*125*	*130*	*13*	*44*	*80*	*95*	*00*	*100*
	126		**--**		**92**		**28**	
Raw Juice (11am-12am)	*125*	*130*	*13*	*44*	*80*	*95*	*00*	*100*
	127		**--**		**89**		**24**	
Raw Juice (2pm-3pm)	*125*	*130*	*13*	*44*	*80*	*95*	*00*	*100*
	129		**--**		**93**		**20**	

Comparative Graphs







9. CONCLUSIONS

I. By performing various tests for the measurement of Temperature, Pressure, Humidity, Water Level 15 consecutive days and studying the comparative graphs it has been observed that the overall quality of the Khandsari Sugar and Jaggery produced was comparatively good due to precise control with the help of newly designed SCADA.

II. The use of recent microcontrollers in the advanced SCADA and Remote SCADA systems in the sugar industries can increase further the quality of sugar and jaggery. It is highly recommended for the use in sugar industries with lack of advanced instrumentation due to financial constraints.

III. Design engineers can use these advanced features for optimizing performance with decrease in design cost due to less number of components.

IV. Sugar and Jaggery yield can further increase as frequent plant shut down is rare due to maintenance-free SCADA system.

V. However, selection of a microcontroller depends not only on the features and the cost factor but also the market trends, manufactures profile and available design expertise.

REFERENCES

1] W H Payne (1990), Embedded Controller Forth For The 8051 Family,Academic Press, USA

2] T W Schultz (1993), C and the 8051: Programming for Multitasking,PTR Prentice Hall, USA

3] M Predko (2007), Programming and Customizing the PIC Microcontroller,McGraw-Hill Professional, USA

4] D W Smith (2002), PIC in Practice: An Introduction to the PIC Microcontroller,, Newnes, USA

5] G McComb (2002), Robot Builder's Sourcebook: Over 2500 Sources for Robot

Parts, McGraw-Hill Professional, USA

6] J Iovine (2004), PIC Microcontrollers, Mc-Graw Hill, NY, USA

7] M Predko (1999), Programming and Customizing The 8051 Microcontroller, Mc-Graw Hill / TAB Electronics, NY, USA

8] D Calcutt, F Cowan, H Parchizadeh (2004), 8051 Microcontrollers – An Application Based Introduction,Elsevier, UK

9] Nasdaq : ATML(February 21, 2005), Atmel Announces Three New Low Power TinyAVR Microcnotrollers with Internal Temperature Sensor Capabilities, Atmel (R) Corporation, San Jose, California, p.p. 1- 4, Thomas Publishing Company (Press Release)

10] NASDAQ: ATML (December 19, 2006), Atmel Offers New AVR Microcontrollers with Integrated LCD Controllers and picoPower Technology, Atmel (R) Corporation, San Jose, California, p.p. 1 - 4, Thomas Publishing Company (Press Release)

11] Nasdaq : ATML(February 21, 2005) , Atmel Announces Three New Low Power TinyAVR Microcnotrollers with Internal Temperature Sensor Capabilities, Atmel (R) Corporation, San Jose, California, p.p. 1- 4, Thomas Publishing Company (Press Release)

12] Databook on current PIC Microcontroller Family Product, (2004) p.p. 12 and Databook on 8- bit AVR Microcontroller with 8-k bytes in-System Programmable Flash (2006) p.p. 07 - 24.

13] Claire O'Keeffe and Brian Black, “Choosing an IC Temperature Sensor” http://www.electronicproducts.com/

14] “National Semiconductor’s Temp Sensor Handbook”, www.national.com/appinfo/tempsensors/files/temphb.pdf

15] “National Semiconductor’s Analog University”, www.national.com/AU/design/1,4678,18_0_,00.html

16] http://www.wrh.noaa.gov/slc/wxcalc/formulas/pressureAltitude.pdf