Secured Multi Channel Wireless Temperature Monitoring and Controlling System for Shell Moulding Process

M.Dinesh\textsuperscript{1}, M.Mathivijay\textsuperscript{2}

\textsuperscript{1}(PG scholar, Department of ECE, Karpagam University, Coimbatore, India, engr.dinesh@ymail.com)
\textsuperscript{2}(Assistant professor, Department of ECE, Karpagam University, Coimbatore, India, mmathivijay@gmail.com)

Abstract—In this paper we achieved these secured automation in temperature control and also power wastage diminishing in heater coils of electric heater based machineries. In this concern we handled the technique in shell molding machineries. The most important considerations in this paper are low cost, long life and high efficiency. In this work the system uses a single temperature controller with four channels and K type thermocouple for measurement. Moreover the system uses Zigbee for secure communication between the system and the server. This method achieves a temperature error of plus or minus 6 degree Celsius, and the acceptable range for the foundry industries is plus or minus 10 degree Celsius. This system provides temperature monitoring and controlling of the process and provides the means to evaluate the equipment’s current condition and detect the failures in thermocouples and the heater coils at an early stage. The system software at the server can be used to monitor and control the temperature set point, temperature upper and lower limits with authorized access.

Keywords—Secured automation, Temperature control and monitoring, Power wastage diminish.

1. INTRODUCTION

The foundry industries are complex and demanding work environment, it needs high quality and reliable automation in engineering and manufacturing process. Still most of the foundry industries run with man dependency process which results increasing trend in overall expenses as well as rejections. To overcome these hurdle stones, foundries have to move towards the automation in their process. Moreover the most important considerations in foundry automations are low cost, long life and high efficiency.

In this work the system uses a single temperature controller with two channels and manual pressure gauge to measuring, to replace two separate temperature controllers for each channel in the core shooter process. Moreover the system uses the CAN for secure communication between the system and the server. Previous works shows that the thermocouple nonlinearity compensation is a difficult process, which requires more processing power. In this work a new method of thermocouple nonlinearity compensation is used by selecting proper multiplication factor, which requires very less processing power. This method achieves a temperature error of ±6°C, and the acceptable range for the foundry industries is ±10°C. This system provides 24/7 online temperature, pressure monitoring & controlling of the core shooter process and provides the means to evaluate the equipment’s current condition and detect the failures in thermocouples and the heater coils at an early stage. Since this system can be placed in locations usually not accessible with an infrared camera, this work can either complement or replace an infrared thermography system. The software tool at the server can be used to monitor and control the temperature set point and the limit values. The overall power reduction of 4% is achieved when employing this system in the core shooter process.

1.1. SHELL MOLDING

Shell molding, also known as shell mold casting, is an expendable mold casting process that uses a resin covered sand to form the mold. As compared to sand casting, this process has better dimensional accuracy, a higher productivity rate, and lower labor requirements. Shell mold casting is a metal casting process similar to sand casting, in that molten metal is poured into an expendable mold. However, in shell mold casting, the mold is a thin-walled shell created from applying a sand-resin mixture around a pattern. The pattern, a metal piece in the shape of the desired part, is reused to form multiple shell molds. A reusable pattern allows for higher production rates, while the disposable molds enable complex geometries to be cast. Shell mold casting requires the use of a metal pattern, oven, sand-resin mixture, dump box, and molten metal.

Shell mold casting allows the use of both ferrous and non-ferrous metals, most commonly using cast iron, carbon steel, alloy steel, stainless steel, aluminum alloys, and copper alloys. Typical parts are small-to-medium in size and require high accuracy, such as gear housings, cylinder heads, connecting rods, and lever arms. The shell mold casting process consists of the following steps:

- Pattern creation - A two-piece metal pattern is created in the shape of the desired part, typically from iron or steel. Other materials are sometimes used, such as aluminum for low volume production or graphite for casting reactive materials.
- Mold creation - First, each pattern half is heated to 175-370°C (350-700°F) and coated with a lubricant to facilitate removal. Next, the heated pattern is clamped to a dump box, which contains a mixture...
of sand and a resin binder. The dump box is inverted, allowing this sand-resin mixture to coat the pattern. The heated pattern partially cures the mixture, which now forms a shell around the pattern. Each pattern half and surrounding shell is cured to completion in an oven and then the shell is ejected from the pattern.

- Mold assembly - The two shell halves are joined together and securely clamped to form the complete shell mold. If any cores are required, they are inserted prior to closing the mold. The shell mold is then placed into a flask and supported by a backing material.
- Pouring - The mold is securely clamped together while the molten metal is poured from a ladle into the gating system and fills the mold cavity.
- Cooling - After the mold has been filled, the molten metal is allowed to cool and solidify into the shape of the final casting.
- Casting removal - After the molten metal has cooled, the mold can be broken and the casting removed. Trimming and cleaning processes are required to remove any excess metal from the feed system and any sand from the mold.

1.1.1. PROCESS OF CREATING A SHELL MOULD

The process of creating a shell mold consists of six steps, which is shown in the Fig. 1.2. The steps involved are,

1. Fine silica sand that is covered in a thin (3–6%) thermosetting phenolic resin and liquid catalyst is dumped, blown, or shot onto a hot pattern. The pattern is usually made from cast iron and is heated to 230 to 315 °C (450 to 600 °F). The sand is allowed to sit on the pattern for a few minutes to allow the sand to partially cure.

2. The pattern and sand are then inverted so the excess sand drops free of the pattern, leaving just the "shell". Depending on the time and temperature of the pattern the thickness of the shell is 10 to 20 mm (0.4 to 0.8 in).

3. The pattern and shell together are placed in an oven to finish curing the sand. The shell now has a tensile strength of 350 to 450 psi (2.4 to 3.1 MPa).

4. The hardened shell is then stripped from the pattern.

5. Two or more shells are then combined, via clamping or gluing using a thermo set adhesive, to form a mold. This finished mold can then be used immediately or stored almost indefinitely.

Fig.1.1. Steps involved in a shell molding process.

6. For casting the shell mold is placed inside a flask and surrounded with shot, sand, or gravel to reinforce the shell.

The machine that is used for this process is called a Shell Molding Machine. It heats the pattern, applies the sand mixture, and bakes the shell.

1.2. EMBEDDED SYSTEMS

An embedded system is a computer system with a dedicated function within a larger mechanical or electrical system, often with real-time computing constraints. It is embedded as part of a complete device often including hardware and mechanical parts. By contrast, a general-purpose computer, such as a personal computer (PC), is designed to be flexible and to meet a wide range of end-user needs. Embedded systems control many devices in common use today.

Modern embedded systems are often based on microcontrollers (i.e. CPUs with integrated memory and/or peripheral interfaces) but ordinary microprocessors (using external chips for memory and peripheral interface circuits) are also still common, especially in more complex systems. In either case, the processor(s) used may be types ranging from general purpose to very specialized in certain class of computations, or even custom designed for the application at hand. A common standard class of dedicated processors is the digital signal processor (DSP).

The key characteristic, however, is being dedicated to handle a particular task. Since the embedded system is dedicated to specific tasks, design engineers can optimize it to reduce the size and cost of the product and increase the reliability and performance. Some embedded systems are mass-produced, benefiting from economies of scale.

1.3. TOOLS

As with other software, embedded system designers use compilers, assemblers, and debuggers to develop embedded system software. However, they may also use some more specific tools:
- In circuit debuggers or emulators.
- Utilities to add a checksum or CRC to a program, so the embedded system can check if the program is valid.
- For systems using digital signal processing, developers may use a math workbench such as Scilab / Scicos, MATLAB / Simulink, EICASLAB, MathCAD, Mathematical, or Flowstone DSP to simulate the mathematics. They might also use libraries for both the host and target which eliminates developing DSP routines as done in DSPnano RTOS and Unison Operating System.
- A model based development tool like VisSim lets you create and simulate graphical data flow and UML State chart diagrams of components like digital filters, motor controllers, communication protocol decoding and multi-rate tasks. Interrupt handlers can also be created graphically. After simulation, you can automatically generate C-code to the VisSim RTOS which handles the main control task and preemption of background tasks, as well as automatic setup and programming of on-chip peripherals.
- Custom compilers and linkers may be used to optimize specialized hardware.
- An embedded system may have its own special language or design tool, or add enhancements to an existing language such as Forth or Basic.
- Another alternative is to add a real-time operating system or embedded operating system, which may have DSP capabilities like DSPnano RTOS.
- Modeling and code generating tools often based on state machines

Software tools can come from several sources:

- Software companies that specialize in the embedded market
- Ported from the GNU software development tools
- Sometimes, development tools for a personal computer can be used if the embedded processor is a close relative to a common PC processor

As the complexity of embedded systems grows, higher level tools and operating systems are migrating into machinery where it makes sense. For example, cell phones, personal digital assistants and other consumer computers often need significant software that is purchased or provided by a person other than the manufacturer of the electronics. In these systems, an open programming environment such as Linux, NetBSD, OSGi or Embedded Java is required so that the third-party software provider can sell to a large market.

1.4. WIRELESS SENSOR NETWORK

A wireless sensor network (WSN) of spatially distributed autonomous sensors to monitor physical or environmental conditions, such as temperature, sound, pressure, etc. and to cooperatively pass their data through the network to a main location. The more modern networks are bi-directional, also enabling control of sensor activity.

The WSN is built of "nodes" – from a few to several hundreds or even thousands, where each node is connected to one (or sometimes several) sensors. Each such sensor network node has typically several parts: a radio transceiver with an internal antenna or connection to an external antenna, a microcontroller, an electronic circuit for interfacing with the sensors and an energy source, usually a battery or an embedded form of energy harvesting.

![Fig. 1.2 Typical multi-hop wireless sensor network architecture](image)

AThe cost of sensor nodes is similarly variable, ranging from a few to hundreds of dollars, depending on the complexity of the individual sensor nodes. The Fig. 1.1 shows a typical WSN architecture. Size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and communications bandwidth. The topology of the WSNs can vary from a simple star network to an advanced multi-hop wireless mesh network. The propagation technique between the hops of the network can be routing or flooding.

1.4.1. CHARACTERISTICS

- Power consumption constraints for nodes using batteries or energy harvesting
- Ability to cope with node failures
- Mobility of nodes
- Communication failures
- Heterogeneity of nodes
- Scalability to large scale of deployment
- Ability to withstand harsh environmental conditions
- Ease of use
- Cross-layer design

1.5. OBJECTIVE OF THE WORK

- To centralize the temperature readings and the set point values for every channel in all the units.
- Provide secure and limited access to the settings of the units.
- To automate the complete process using wireless automation system.
To log each and every reading obtained from the system and to maintain separate databases for all the parameter.

2. LITERATURE REVIEW

2.1 THERMOCOUPLE: A SENSOR USED TO MEASURE THE TEMPERATURE

Kim Fowler and John Schmalzel (2005), in this introduced many of the common sensor strategies that constitute the first stage in the measurement chain.

![Fig. 2.1 Typical Thermocouple circuit for measuring Temperature.](image)

While every element of that chain is important, the choice of an appropriate, robust sensor is an important decision.

They gave examples from test stands for testing rocket engines [7]. Measurements of the weight of liquids, their temperatures, and their flow rates in these rocket engine test stands require sensors and the understanding of how to use them. The Fig. 2.1 shows a typical thermocouple circuit for temperature measurement.

Each sensor and sensor type has its characteristic values for a wide variety of descriptive parameters, such as accuracy, resolution, stability, specificity, linearity, and hysteresis.

2.2 MULTIVARIABLE SELF TUNING TEMPERATURE CONTROL FOR MOLDING PROCESS

Chi-Huang Lu and Ching-Chih Tsai (1998) in their paper show about an adaptive decoupling temperature control for an extrusion barrel in a plastic injection molding process [2]. After establishing a stochastic polynomial matrix model of the system, a corresponding decoupling system representation was then developed. The decoupling control design was derived based on the minimization of a generalized predictive performance criterion. The set-point tracking, disturbance rejection, and robustness capabilities of the proposed method can be improved by appropriate adjustments to the tuning parameters in the criterion function.

A real-time control algorithm, including the recursive least-squares method, is proposed which was implemented using a digital signal processor TMS320C31 from Texas Instruments. Through the experimental results, the proposed method has been shown to be powerful under set-point changes, load disturbances, and significant plant uncertainties. The proposed control law is shown to be less computational and more effective than other well-known multivariable control strategies, and more powerful than single-loop temperature-zone control policies.

2.3 IDENTIFICATION AND AUTOTUNING OF TEMPERATURE

Tao Liu, Ke Yao and Furong Gao (2009), proposes two identification methods and a unified control scheme for general temperature control design [6]. Based on the unity step response, corresponding to the full heating response in a temperature-control system, an identification method is developed to obtain an integrating model for heating-up control design which is shown in the Fig. 2.2. By using a relay test around the set point temperature, another identification method is proposed to obtain a model of stable or integrating type for control-system design to reject load disturbances during system operation.

![Fig. 2.2 Block diagram of the Temperature-Control structure](image)
amounts of heating power may result in different response characteristics. In view of that, relay-feedback test is suitable for online identification and can be used to observe the repeatable response characteristics around the set point temperature; a relay-based identification method is proposed to capture the fundamental response characteristics at the steady state for control-system design.

2.4 XBEE SERIES 2 RF MODULES

The XBee Series 2 RF Modules were engineered to operate within the Zigbee protocol and support the unique needs of low-cost, low-power wireless sensor networks. The modules require minimal power and provide reliable delivery of data between remote devices. The modules operate within the ISM 2.4 GHz frequency band.

High Performance & Low Cost

Indoor/Urban: up to 133’ (40 m)
- Outdoor line-of-sight: up to 400’ (120 m)
- Transmit Power: 2 mW (+3 dBm)
- Receiver Sensitivity: -95 dBm
- RF Data Rate: 250,000 bps

Advanced Networking & Security
- Retries and Acknowledgements
- DSSS (Direct Sequence Spread Spectrum)
- Each direct sequence channel has over 65,000 unique network addresses available
- Point-to-point, point-to-multipoint and peer-to-peer topologies supported
- Self-routing, self-healing and fault-tolerant mesh networking

Low Power XBee Series 2
- TX Current: 40 mA (@3.3 V)
- RX Current: 40 mA (@3.3 V)
- Power-down Current: < 1 µA @ 25°C

2.5 AUTHENTICATED THE MASTER IN THE MODBUS PROTOCOL

According to Gen-Yih Liao, Yu-Jen Chen, Wen-Chung Lu, and Tsung-Chieh Cheng (2008), the power systems relying on the Modbus protocol for distributed control render their facilities vulnerable since there is no security consideration in the protocol. An intruder, without being authenticated, may harm the system by issuing malicious commands. This letter presents a cryptographic scheme that aims to enhance the Modbus protocol with authenticity.

Distributed control systems (DCS) enable modern power plants to monitor and manipulate remote devices, sometimes geographically located miles away. Various network protocols have been proposed to specify communication formats between a control center and its subordinate devices. Modbus is one example that has been widely deployed on the latest power plants.

1) The attacker fakes as the master to maliciously send inappropriate commands that may cause remote devices to malfunction.
2) The attacker may capture a slave device and uncover any secret (i.e., cryptographic keys) stored in that device.

Authentication by using symmetric encryption does not work if an attacker can capture slave devices to learn the secret keys. Public-key cryptosystems can eliminate the concerns that have occurred due to device hijacking, but the computational load is not negligible, especially to those slave devices with limited resources.

This aims to enhance the Modbus protocol with authenticity. More specifically, remote slave devices can authenticate received commands coming from the master in cases where an attacker is able to send messages over the wire. Besides, efficiency is also an important design requirement. This means the protocol should not consume too much memory and processing power of resource-constrained devices.

2.6 AUTHENTICATED MODBUS PROTOCOL FOR CRITICAL INFRASTRUCTURE PROTECTION

Raphael C.-W. Phan (2012), in his research work discusses about Protecting a nation’s critical infrastructure, notably its power grid is crucial in view of increasing threats, such as international terrorism. We focus on the security of the Modbus protocol, a de-facto protocol for distributed control systems popularly used for power plants. Specifically, we analyze the security of a recently proposed authenticated Modbus protocol [5]. They present attacks on the protocol, discuss reasons behind these phenomena, and motivate how these problems can be addressed.

2.7 WIRELESS SENSORS NETWORK BASED ON MODBUS PROTOCOL

The information transmission is transparent for the user in the Zigbee wireless sensors network, which are lack of interactivity and self-constrain. The information in the Zigbee wireless sensors network cannot be viewed in a real time by a friendly interface. Modbus protocol is embedded into Zigbee stack, in this way, we can implement interaction well and the information can be viewed in a friendly interface. The paper present the measures to embed the Modbus protocol into the Zigbee stack provided by Chipcon company, which contains address bound mechanism, information centralized storage, and flexible monitoring, by
which we can monitor the real time information from the ZigBee wireless network and use some instructions to control the remote device in a friendly interface, which can be used well in the middle and small ZigBee monitoring wireless sensors network.

2.8 SERVICE-ORIENTED ARCHITECTURE FOR INDUSTRIAL AUTOMATION

Tommaso Cucinotta, Antonio Mancina, Gaetano F. Anastasi, Giuseppe Lipari, Leonardo Mangeruca, Roberto Checcozzo and Fulvio Rusinà (2009), in their work say that the Industrial automation platforms are experiencing a paradigm shift. New technologies are making their way in the area, including embedded real-time systems, standard local area networks like Ethernet, Wi-Fi and ZigBee, IP-based communication protocols, standard Service Oriented Architectures (SOAs) and Web Services. The experiments were performed on simple case studies designed in the context of industrial automation applications.

3 DESIGN OF HARDWARE FOR WIRELESS TEMPERATURE MONITORING SYSTEM

3.1. HARDWARE

The Fig. 3.1 and Fig. 3.2 show the block diagrams for the hardware in the system side and the machine side. The proposed system consists of the following major units,

- Thermocouple
- ADC
- Microcontroller
- Zigbee
- Power Supply
- Relay control unit
- PC with Monitoring Software

3.2. XC886 INDUSTRIAL GRADE MICROCONTROLLER

Fig. 3.3 The circuit diagram for the XC886 microcontroller module circuitry.

The Infineon XC800 Family is an 8-bit Microcontroller family, first introduced in 2005 with a dual cycle optimized 8051 “E-Warp” Core. The XC800 family is divided into two categories, the A-Family for Automotive and the I-Family for Industrial & Multi Market applications.

3.3. Key Features

Core

The instruction set consists of 45% one-byte, 41% two-byte and 14% three-byte instructions. Each instruction takes 1, 2 or 4 machine cycles to execute. In case of access to slower memory, the access time may be extended by wait cycles (one wait cycle lasts one machine cycle, which is equivalent to two wait states). The XC800 Core supports a range of debugging features including basic stop/start, Single-step execution, breakpoint support and read/write...
access to the data memory, program memory and special function registers.

Memory Organization

The 8-Bit MCUs have an embedded user-programmable non-volatile Flash memory that allows for fast and reliable storage of user code and data. It is operated with a single 2.5 V supply from the embedded voltage regulator (EVR) and does not require additional programming or erasing voltage. A Flash error correction (ECC) can detect double-bit errors and correct single-bit errors as well as protect against invalid code execution. Up to 3KB of RAM is featured, part of this memory being XRAM.

ADC

The Analog Digital Converter module (ADC) uses the successive approximation method to convert analog input values (voltages) to discrete digital values with 10-bit resolution. One ADC kernel (ADC0) operates on a user-selectable number of input channels. The input channels can be selected and arbitrated flexibly.

CCU6

The CCU6 is a capture and compare unit which generates PWM signals over different duty cycles and multiple output channels. It operates with 16-bit timers clocked at 48 MHz and can trigger the ADC operation to harmonize control loops.

Touch and LED Matrix control

LEDTSCU is a functional unit for the control of capacitive touch pads and a matrix of LEDs through the same pins. The principle of time multiplexed operation of two or more functions, in this case touch control and LED-control, reduces the amount of pins used.

Industrial Applications

The Industrial-Family also called I-Family product series ranging from 2KB to 64KB Flash memory and from 16- to 64-pin package options. It can be found in applications like motor control of eBikes, pumps and fans e.g. in air conditioners, as display or touch button controls or in digital controlled power supplies e.g. for motor drives or lighting.

3.4. MCP 3424-18-BIT MULTI-CHANNEL ΔΣ ANALOG TO DIGITAL CONVERTER WITH I²C INTERFACE AND ON BOARD REFERENCE

The MCP3424 device (MCP3422/3/4) is the low noise and high accuracy 18-Bit delta-sigma analog-to-digital (ΔΣA/D) converter family members of the MCP342X series from Microchip Technology Inc. These devices can convert analog inputs to digital codes with up to 18 bits of resolution.

The on-board 2.048V reference voltage enables an input range of ±2.048V differentially (full scale range = 4.096V/PGA).

Conversion modes: (a) One-Shot Conversion mode and (b) Continuous Conversion mode. In One-Shot conversion mode, the device performs a single conversion and enters a low current standby mode automatically until it receives another conversion command. This reduces current consumption greatly during idle periods. In Continuous conversion mode, the conversion takes place continuously at the set conversion speed. The device updates its output buffer with the most recent conversion data.

The devices operate from a single 2.7V to 5.5V power supply and have a two-wire I2C compatible serial interface for a standard (100 kHz), fast (400 kHz), or high-speed (3.4 MHz) mode. The I2C address bits for the MCP3423 and MCP3424 are selected by using two external I2C address selection pins (Adr0 and Adr1). The user can configure the device to one of eight available addresses by connecting these two address selection pins to VDD, VSS or float. The I2C address bits of the MCP3422 are programmed at the factory during production.

3.5. RESET CIRCUITRY

It takes a short time after MCU power-on for the internal circuitry to stabilize. During this interval, the CPU cannot be expected to perform normally. This problem is resolved by applying a reset signal to the reset pin on the MCU using the reset circuit shown in the Fig. 3.6. Setting the signal to active (LOW) causes the MCU to reset.

The circuit will reset and hold the voltage on the timing cap to VCC/3 so long as reset is high. VCC/3 is the voltage on the cap at the beginning of the timing cycle, so when you release reset, the length of the first cycle will be...
very close to all those that follow. The output (pin 3) will be high while reset is high.

3.6. RS232 – TTL BRIDGE

The MAX232 is an RS232 – TTL Bridge IC that converts signals from an RS-232 serial port to signals suitable for use in TTL compatible digital logic circuits. The MAX232 is a dual driver/receiver and typically converts the RX, TX, CTS and RTS signals.

The circuit in the Fig.3.7 provides RS-232 voltage level outputs (approx. ± 7.5 V) from a single +5 V supply via on-chip charge pumps and external capacitors. This makes it useful for implementing RS-232 in devices that otherwise do not need any voltages outside the 0 V to +5 V range, as power supply design does not need to be made more complicated just for driving the RS-232 in this case. The receivers reduce RS-232 inputs (which may be as high as ±25 V), to standard 5 V TTL levels. These receivers have a typical threshold of 1.3 V, and a typical hysteresis of 0.5 V.

3.7. PERIPHERALS OVER THE I2C BUS

3.7.1. I2C Protocol

I²C (Inter-Integrated Circuit) is a multimaster serial single-ended computer bus invented by the Philips semiconductor division, today NXP Semiconductors, and used for attaching low-speed peripherals to a motherboard, embedded system.

The DS1307 serial real time clock (RTC) is a low power, full binary coded decimal (BCD) clock/calendar plus 56 bytes of NV SRAM. Address and data are transferred serially through an I2C, bidirectional bus. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24 hour or 12 hour format with AM/PM indicator. The DS1307 has a built in power sense circuit that detects power failures and automatically switches to the backup supply. Timekeeping operation continues while the part operates from the backup supply.

3.8. RELAY DRIVER CIRCUIT USING ULN 2003A

A ULN2003A is a high-voltage, high-current Darlington transistor array. It consists of seven NPN Darlington pairs that feature high-voltage outputs with common-cathode fly back diodes for switching inductive loads.

The drivers can be paralleled for higher current capability, even stacking one chip on top of another, both electrically and physically has been done.

Features

- 500 mA rated collector current (single output)
- 50 V output
- Includes output flyback diodes
- Inputs compatible with various types of logic

The ULN2003 is a monolithic IC consists of seven NPN Darlington transistor pairs with high voltage and current capability. It is commonly used for applications such as relay
drivers, motor, display drivers, led lamp drivers, logic buffers, line drivers, hammer drivers and other high voltage current applications.

The driver shown in the Fig.3.9 provides open collector output, so it can only sink current, cannot source. Thus when a 5V is given to 1B terminal, 1C terminal will be connected to ground via Darlington pair and the maximum current that it can handle is 500A. From the above logic diagram we can see that cathode of protection diodes are shorted to 9th pin called COM. So for driving inductive loads, it must be connected to the supply voltage.

3.9. OPTOCOUPLED DIGITAL INPUTS FOR PRODUCTION COUNT

In electronics, an opto-isolator, also called an optocoupler, photo coupler, or optical isolator, is a component that transfers electrical signals between two isolated circuits by using light. Opto-isolators prevent high voltages from affecting the system receiving the signal.

Commercially available opto-isolators withstand input-to-output voltages up to 10 kV and voltage transients with speeds up to 10 kV/μs. A common type of opto-isolator shown in the Fig.3.10 consists of an LED and a phototransistor in the same opaque package.

4. EMBEDDED SOFTWARE AND APPLICATION SOFTWARE FOR THE WIRELESS TEMPERATURE MONITORING SYSTEM

4.1. EMBEDDED SOFTWARE

Embedded software is computer software, written to control machines or devices that are not typically thought of as computers. It is typically specialized for the particular hardware that it runs on and has time and memory constraints.

4.2. FUNCTIONS OF THE EMBEDDED SOFTWARE

The temperature controllers are used to maintain constant temperature of process or plant or any material. In this system there is one reference temperature called set point hat is the desired temperature that must be maintained. This reference temperature is set by external means.

Once this temperature is set, it senses current temperature, compares it with reference temperature and generates error signal. Then based on this error signal it controls the heating element to turn ON/OFF.

4.3. APPLICATION SOFTWARE

The system application software is broadly classified into two major parts as shown in the Fig.4.1.

One is the Windows Service Application and the other is the Windows Forms Application. Microsoft Windows services, formerly known as NT services, enable you to create long-running executable applications that run in their own Windows sessions. These services can be automatically started when the computer boots, can be paused and restarted, and do not show any user interface [8]. This makes services ideal for use on a server or whenever you need long-
running functionality that does not interfere with other users who are working on the same computer. It can also be run in the security context of a specific user account that is different from the logged-on user or the default computer account.

5. CONCLUSION

In this work a low cost Zigbee is utilized to implement the wireless communication. The security of the system is increased through password access for both the machine side and the server side. The OEE of machines has been improved effectively. The experimental results show that the system works effectively, and the overall power consumption of the Shell Molding unit is reduced by 4%. The drawbacks of the existing system like more number of controllers, low OEE and chances for malfunctions have been overcome. With the developed control scheme and the lowest cost implementation, the proposed scheme is suitable for even small scale industries.

6. REFERENCES