CHAPTER 5

HARDWARE IMPLEMENTATION IN DEFECT IDENTIFICATION

5.1 INTRODUCTION

A Microcontroller is a small computer on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals. Program memory in the form of NOR flash is often included on chip, as well as a typically small amount of RAM. Microcontrollers are designed for embedded applications, in contrast to the microprocessors used in personal computers or other general purpose applications.

Microcontrollers are used in automatically controlled products and devices, such as automobile engine control systems, implantable medical devices, remote controls, office machines, appliances, power tools, and toys. By reducing the size and cost compared to a design that uses a separate microprocessor, memory, and input/output devices, microcontrollers make it economical to digitally control even more devices and processes. Mixed signal microcontrollers are common, integrating analog components needed to control non-digital electronic systems expressed by Ajay (2007).

Some microcontrollers may use four-bit words and operate at clock rate frequencies as low as 4 kHz, for low power consumption (milliwatts or microwatts). They will generally have the ability to retain functionality while waiting for an event such as a button press or other interrupt; power
consumption while sleeping (CPU clock and most peripherals off) may be just nanowatts, making many of them well suited for long lasting battery applications. Other microcontrollers may serve performance-critical roles, where they may need to act more like a Digital Signal Processor (DSP), with higher clock speeds and power consumption.

A microcontroller can be considered a self-contained system with a processor, memory and peripherals and can be used as an embedded system. The majority of microcontrollers in use today are embedded in other machinery, such as automobiles, telephones, appliances, and peripherals for computer systems. These are called embedded systems. While some embedded systems are very sophisticated, many have minimal requirements for memory and program length, with no operating system, and low software complexity. Typical input and output devices include switches, relays, solenoids, LEDs, small or custom LCD displays, radio frequency devices, and sensors for data such as temperature, humidity, light level etc. Embedded systems usually have no keyboard, screen, disks, printers, or other recognizable I/O devices of a personal computer, and may lack human interaction devices of any kind.

Microcontrollers must provide real time (predictable, though not necessarily fast) response to events in the embedded system they are controlling. When certain events occur, an interrupt system can signal the processor to suspend processing the current instruction sequence and to begin an Interrupt Service Routine (ISR). The ISR will perform any processing required based on the source of the interrupt before returning to the original instruction sequence. Possible interrupt sources are device dependent and often include events such as an internal timer overflow, completing an analog to digital conversion, a logic level change on an input such as from a button being pressed, and data received on a communication link. Where power consumption is important as in battery operated devices, interrupts may also
Microcontroller programs must fit in the available on-chip program memory, since it would be costly to provide a system with external, expandable, memory. Compilers and assemblers are used to convert high-level language and assembler language codes into a compact machine code for storage in the microcontroller's memory. Depending on the device, the program memory may be permanent, read-only memory that can only be programmed at the factory, or program memory may be field-alterable flash or erasable read-only memory.

5.2 FEATURES OF MICROCONTROLLERS

Microcontrollers usually contain from several to dozens of General Purpose Input/Output pins (GPIO). GPIO pins are software configurable to either an input or an output state. When GPIO pins are configured to an input state, they are often used to read sensors or external signals. Configured to the output state, GPIO pins can drive external devices such as LEDs or motors.

Many embedded systems need to read sensors that produce analog signals. This is the purpose of the Analog-to-Digital Converter (ADC). Since processors are built to interpret and process digital data, either 1s or 0s, they are not able to do anything with the analog signals that may be sent to it by a device. So the Analog to Digital converter is used to convert the incoming data into a form that the processor can recognize. The less common feature on some microcontrollers is a Digital-to-Analog converter (DAC) that allows the processor to output analog signals or voltage levels.

In addition to the converters, many embedded microprocessors include a variety of timers as well. One of the most common types of timers is
the Programmable Interval Timer (PIT). A PIT may either count down from some value to zero, or up to the capacity of the count register, overflowing to zero. Once it reaches zero, it sends an interrupt to the processor indicating that it has finished counting. This is useful for devices such as thermostats, which periodically test the temperature around them to see if they need to turn the air conditioner on, the heater on, etc.

Time Processing Unit (TPU) is a sophisticated timer. In addition to counting down, the TPU can detect input events, generate output events, and perform other useful operations. A dedicated Pulse Width Modulation (PWM) block makes it possible for the CPU to control power converters, resistive loads, motors, etc., without using lots of CPU resources in tight timer loops. Universal Asynchronous Receiver/Transmitter (UART) block makes it possible to receive and transmit data over a serial line with very little load on the CPU. Dedicated on-chip hardware also often includes capabilities to communicate with other devices (chips) in digital formats such as I C and Serial Peripheral Interface (SPI).

In contrast to general-purpose CPUs, micro-controllers may not implement an external address or data bus as they integrate RAM and non-volatile memory on the same chip as the CPU. Using fewer pins, the chip can be placed in a much smaller, cheaper package.

Integrating the memory and other peripherals on a single chip and testing them as a unit increases the cost of that chip, but often results in decreased net cost of the embedded system as a whole. Even if the cost of a CPU that has integrated peripherals is slightly more than the cost of a CPU and external peripherals, having fewer chips typically allows a smaller and cheaper circuit board, and reduces the labour required to assemble and test the circuit board.
A Micro-controller is a single integrated circuit, commonly with the following features:

### 5.2.1 Central Processing Unit (CPU)

CPU ranges from small and simple 4-bit processors to complex 32-or 64-bit processors discrete input and output bits, allowing control or detection of the logic state of an individual package pin serial input/output such as serial ports (UARTs), other serial communications interfaces like PC, Serial Peripheral Interface and Controller Area Network for system interconnect peripherals such as timers, event counters, PWM generators, and watchdog. Volatile memory (RAM) for data storage ROM, EPROM, EEPROM or Flash memory for program and operating parameter storage.

### 5.2.2 Clock Generator

It is often an oscillator for quartz timing crystal, resonator or RC circuit many include analog-to-digital converters in-circuit programming and debugging support.

This integration drastically reduces the number of chips and the amount of wiring and circuit board space that would be needed to produce equivalent systems using separate chips. Furthermore, and on low pin count devices in particular, each pin may interface with several internal peripherals, along with the pin function selected by software. This allows a part to be used in a wider variety of applications than if pins had dedicated functions. Micro-controllers have proved to be highly popular in embedded systems since their introduction in the 1970s.

Some microcontrollers use Harvard architecture: separate memory buses for instructions and data, allowing accesses to take place concurrently.
Where Harvard architecture is used, instruction words for the processor may be a different bit size than the length of internal memory and registers. The decision of which peripheral to integrate is often difficult. The microcontroller vendors often trade operating frequencies and system design flexibility against time-to-market requirements from their customers and overall lower system cost. Manufacturers have to balance the need to minimize the chip size against additional functionality.

Microcontroller architectures vary widely. Designs include general-purpose microprocessor cores, with one or more ROM, RAM, or I/O functions integrated into the package. Other designs are purposely built for control applications. A micro-controller instruction set usually has many instructions intended for bit-wise operations to make control programs more compact. A general purpose processor might require several instructions to test a bit in a register and branch if the bit is set, where a micro-controller could have a single instruction to provide that commonly required function.

Microcontrollers typically do not have a math coprocessor, so floating point arithmetic is performed by software. About 55% of all CPUs sold in the world are 8-bit microcontrollers and microprocessors. According to Semi conductors, over four billion 8-bit microcontrollers were sold in 2006. A typical home in a developed country is likely to have only four general-purpose microprocessors but around three dozen microcontrollers. A typical mid-range automobile has as many as 30 or more microcontrollers. They can also be found in many electrical devices such as washing machines, microwave ovens, and telephones.

Manufacturers have often produced special versions of their microcontrollers in order to help the hardware and software development of the target system. Originally these included EPROM versions that have a window on the top of the device through which program memory can be erased by
ultraviolet light, ready for reprogramming after a programming (burn) and test cycle. Since 1998, EPROM versions are rare and have been replaced by EEPROM and flash, which are easier to use (can be erased electronically) and cheaper to manufacture.

Other versions may be available where the ROM is accessed as an external device rather than as internal memory, however these are becoming increasingly rare due to the widespread availability of cheap microcontroller programmers. The use of field-programmable devices on a microcontroller may allow field update of the firmware or permit late factory revisions to products that have been assembled but not yet shipped. Programmable memory also reduces the lead time required for deployment of a new product where, hundreds of thousands of identical devices are required; using parts programmed at the time of manufacture can be an economical option discussed by Muhammed et al (2008).

5.3 PROGRAMMING ENVIRONMENTS

Microcontrollers were originally programmed only in assembly language, but various high-level programming languages are now also in common use to target Microcontrollers. These languages are either designed especially for the purpose, or versions of general purpose languages such as the C programming language. Compilers for general language purpose will typically have some restrictions as well as enhancements to better support the unique characteristics of Microcontrollers. Some Microcontrollers have environments to aid developing certain types of applications. Microcontroller vendors often make tools freely available to make it easier to adopt their hardware.

Many microcontrollers are so eccentric that they effectively require their own non-standard dialects of C, which prevent using standard tools (such
as code libraries or static analysis tools) even for code unrelated to hardware features. Interpreters are often used to hide such low level quirks.

Simulators are available for some Microcontrollers, such as in Microchip's Microprocessor lab environment. These allow a developer to analyze what the behaviour of the microcontroller and their program should be if they were using the actual part. A simulator will show the internal processor state and also that of the outputs, as well as allowing input signals to be generated. While on the one hand most simulators will be limited from being unable to simulate much other hardware in a system, they can exercise conditions that may otherwise be hard to reproduce at will in the physical implementation, and can be the quickest way to debug and analyze problems.

5.4 MAIN CIRCUIT AND OPERATION

A prototype is proposed to examine the technique in real time and is represented in Figure 5.3. This prototype is designed by the Figure 5.1. The fabric images are acquired under a source of sufficient illumination by a camera. The camera is synchronized to the fabric motion and used to acquire high-resolution, vibration-free images of the fabric under construction. A central processing unit or lap-top is employed for processing the acquired images using software. The results of the processing are used to detect and characterize fabric defects. Also, it is used to take actions for reporting and correcting these defects to replace or remove these parts from the production line. The prototype has to be robust.

Thus, it should adapt automatically and achieve consistently high performance despite irregularities in illumination and accommodate uncertainties in angles, positions etc. Figure 5.1 represents the main circuit which shows the detail of microcontroller implementation. A 230V AC regulated power supply is given to the circuit. An AC power supply typically
takes the voltage from the input source and lowers it to the desired voltage.

Then the output is connected to the bridge rectifier constructed with Diodes 1N4007. Here the rectified output is given to the voltage Regulator 7805. This Voltage Regulator gives a constant output of 5V. A capacitor of 1000μF/25V is connected across the voltage regulator to minimise the ripples.

A NPN Transistor is connected to the relay to drive the circuit when it is overloaded. This transistor drives up to 800mA. When the relay coil is energized with direct current, a diode D1 is placed across the coil to dissipate the energy from the collapsing magnetic field at deactivation, which would otherwise generate voltage spikes. A 12V relay is used to protect the circuit. The relay output is connected to NPN transistor which is used to amplify the output current up to 800mA.

Resistors of 2.2K and 1K are connected in parallel and connected to the PIC16F877 Microcontroller to limit the current. PIC16F877 controller used has following features:

i) Operating speed: DC – 20 MHz clock input and DC – 200 ns instruction Cycles

ii) High Interrupt capability

iii) Operating speed: DC – 20 MHz clock input and DC – 200 ns instruction Cycles

iv) Programmable code-protection

The relay is connected to the DC Motor of 5V output. Then it is connected to the conveyor setup. MAX232 IC is used to communicate the controller with Personal
Computer (PC) to establish the connection with the circuit. The PC is programmed in such a way that it is used as an interface through the serial cable by DB9 Female connector to the PC.

A Crystal oscillator of 20000 MHZ is used to generate clock signal for controller activation. Capacitors C1 and C2 are used as filters. This controller is interfaced with PC through the MAX232 IC and serial interface cable RS232 to the system. A resistor of 1k is connected to pin number 1 and grounded. This Microcontroller is programmed through Port A. The NN and BN comparison fabric image output is given to the microcontroller through MATLAB programming.

This online defect detection technique is evaluated through the shown prototype which gathers fabric images continuously using a camera. This camera has an ability to acquire a 1 meter image wide at 580 dpi horizontal and vertical resolution. Various images of fabrics are gathered under a unit with corresponding accessories and high-performance PC or lap-top which enables scalability concerning fabric production speed and width.

As shown in Figure 5.3, the camera is installed in the middle of the conveyor set up at 10cm distance from the fell of the cloth and 75 cm in height of the loom with 90° angle against the produced fabric. In addition, the camera and lighting are delivered in a stable frame. The scan speed is around 500 lines/second while the scanning line is around 300 mm wide.

Fabric surface images are obtained at a resolution of 580 dpi along the scanning line. Images are digitized in pixels and stored in a computer as gray scale data for image analysis. Recognition was achieved when the maximum correlation value of the scaled and rotated power spectra was similar to the autocorrelation of the power spectrum of the pattern fabric sample.
In this system as the recognizer identifies a fault of any type taken for analysis, will immediately recognize the type of fault which in return will trigger the sensor in order to display the upper offset and the lower offset of the faulty portion. The upper offset and the lower offset implies the 2 inches left and 2 inches right offset of faulty portion. This defective area needs to be extracted from the roll.

For this the automated system generates a signal to stop the rotation of the DC motor and cut off the faulty portion. Whenever the signal is generated the controller circuit stops the movement of the carrying belt and the defective portion of the fabric is removed from the roll. Then after eliminating the defective part which takes 0 to 5 minutes. Again a signal is generated to start the DC motor and continue the further process. Here the whole system implementation is done in a very simple way.

Figure 5.1 Main Circuit for Micro Controller Operation
5.5 **FLOW CHART**

![Flow Chart Diagram]

- **Start**
- Supply given to circuit
- PIC controller is designed to detect the type of fault
- Classification of types of faults
- Hole (or) Scratch (or) Stain (or) Gout (or) Missing yarn (or) Knots (or) No defect
- Types of defect Identification by Programming
- If Fault is there:
  - **No** → Motor rotates
  - **Yes** → DC motor stops rotation and fault is identified
- **End**

*Figure 5.2 Flow Chart*
Figure 5.2 represents the flow chart of defect detection process in fabrics by microcontroller along with DC motor and conveyor set up.

### 5.6 ADVANTAGES OF PIC16F877 CONTROLLER

The Main advantages of PIC controller are listed below:

i) Operating frequency is high up to 20 MHZ.

ii) Resets and Delays timings are faster

iii) Data Memory bytes are high

iv) Serial and Parallel communications can be done easier than Atmel Controllers.

v) It has 16 instruction sets

vi) It can be used for interfacing external peripherals comfortably than other controllers

Figure 5.3 Hardware model of Defect Detection System
Figure 5.3 shows the real mechanism and detection process of faulty part using PIC16F877 controller devoted by Mursalin et al (2008)

The MATLAB program coding is used for interfacing the textural image with the Neural Network, Bayesian Network and the Microcontroller.

5.7 RESULTS AND DISCUSSION

Using PIC16F877 Microcontroller, the proposed defect detection system is evaluated and hardware performance is carried out. After analysing detection parameters, the detection time required to implement this defect detection technique on real fabric images is measured. In fact, such time has a relative importance as it depends on the hardware of the used system.

In addition, there are other means to reduce the required time. But measuring the time at this stage is important to give an indication the capability of the proposed technique to detect real fabric defects in a short time even with a normal system.

Usually there are two values such as maximum and minimum values for each main detection parameters which is achieved by training. This means that any tuning applied to the technique settings has no influence on the time needed to implement the technique. It is found also that defect type has no influence on this time.

Broadly, such time is around 0 to 0.7 second for all defect detection types. During this time, an image of pixels acquired at 500 dpi is scanned which is equivalent to 1.27 cm of fabric. Consequently, the detection technique is able to inspect at least one meter of fabric each minute. To eliminate the defect from the roll it takes around 0-5 minutes and there is no difference depending on the type of defect.
Industrially, high speed weaving machines run at 1000 picks/min. Therefore, the speed of the technique is 2-3 times the machine productivity. Which means that it is fast and consequently could be implemented for the online fabric inspection.

This online defect detection technique is evaluated through the developed prototype which is described in this chapter. The prototype grabs fabric images continuously using the camera that, with the provided optics, has an ability to acquire a 1.158 meter wide image at 580 dpi resolution. It is found that this online automated fabric inspection prototype is capable of identifying the existing fabric defects.

PIC controller can be easily interfaced with the hardware setup to obtain the output and their features, advantages are discussed.

Hardware circuit is explained and is represented in the form of flow chart. Figure 5.1 and 5.3 shows the hardware details of the defect detection process. Features and advantages of PIC controller and its implementation in the circuit are explained in this chapter.

5.8 CONCLUSION

The current chapter describes about the PIC16F877 Microcontroller, which is used in textile quality improvement in defect detection system. The design of the proposed system is verified through simulations using Microcontroller and DC motor setup. The proposed automated defect detection system in fabrics has the capability of improving the textile quality at the point of operation and enhances consumer satisfaction.

The analysis of the simulation and hardware results of the combined operation of image processing, Neural Networks, Bayesian networks and
PIC16F877 Microcontroller shows that the proposed systems is capable of compensating textile defects and improving the accuracy in performance. It is an approach that combines most advantages with lower drawbacks to be implemented as the base of constructing effective and accurate online automated fabric defect detection.

The advantage of this system is its direct interface with Personal Computer, Laptop by connecting through the MAX232 IC. Hardware results confirm that the proposed automated system operates accurately.

From the implementation of the technique on both simulated and real fabric images it is discovered that:

1. The results of defect detection were satisfactory and illustrated the potential of utility and applicability of the procedure to detect all fabric defects.

2. From a large amount of analysis performed, it is discovered that the important detection parameters which affect the detection results are the features of the image, the scanning step and the threshold. It is found that the speed of the technique is 2-3 times the machine productivity. Which means that it is fast and consequently could be implemented for the online fabric inspection.