

SPEED CONTROL OF THE INDUCTION DRIVE BY TEMPERATURE AND LIGHT SENSORS VIA PIC

Nader Barsoum

Curtin University of Technology, Sarawak, Malaysia
nadir.b@curtin.edu.my

Received September 2009, Revised January 2010, Accepted March 2010

Abstract

PIC is a family of Harvard architecture microcontrollers made by Microchip Technology, derived from the PIC1640 originally developed by General Instrument's Microelectronics Division. The name PIC initially referred to "Peripheral Interface Controller". PICs are popular with the developers and the hobbyists due to their low cost, wide availability, large user base, extensive collection of application notes, free development tools, and serial programming (and re-programming with flash memory) capability. In modern days, PIC microcontrollers are used in the industrial world to control many types of equipment, ranging from consumer to specialized devices. They have replaced older types of controllers, including microprocessors. Also, there is a growing need for off-line support of a computer's main processor. The demand is going to grow with more equipment uses more intelligence. In the engineering field for instance, PIC has brought a very positive impact in designing an automation control system and controlling industrial machineries. Accordingly, this paper shows the change in the motor speed by the use of PIC in accordance to the light and level of temperature. The project focuses on programming the PIC by embedded software that detects the temperature and light signals and send it to 3 phase induction motor of 240 volt. A theoretical analysis and the practical approach in achieving this work goal have proved that PIC plays an important role in the field of electronics control.

Keywords: PIC, speed drive, microcontroller, program, induction motor, mechatronic

1. Introduction

These days, microcontroller development has made it possible to store hundreds of thousands of transistors into one chip. In 1971, the first microprocessors were made by adding external peripherals such as memory, input-output lines, timers and other (Matic, 2003). This product increase dramatically each year. As a result this creates another new device called integrated circuits. These integrated circuits contained both

processor and peripherals. That is how the first chip containing a microcomputer, or to be known as a microcontroller came about [11].

The Peripheral Interface Controller (PIC) is a new device in the field of electronics control. It is a complete computer control system on a single chip [10]. PIC microcontroller include EPROM program memory, user RAM for storing program data, timer circuits, an instruction set, special function registers, power on reset, interrupts, low power consumption and on board A to D converters. In real life application, PIC microcontroller is used to replace conventional methods of controlling industrial machineries such as motor speed control [6, 7, 8, 9].

The objective of this paper is to provide a better understanding on PIC microcontroller as a control device. Researchers will have the opportunity to learn how to program the PIC by using the MPLAB IDE (MPLAB integrated Development Environment) as the development tool, to write the programs in assembly language, to convert the analog to digital signals in PIC [4, 5], to detect the temperature and light intensity from an analog temperature and light sensor, to display the value of the temperature and light signal, to inject the code into PIC and to accomplish this project big mission which is to vary the speed of the motor in accordance to the temperature reading and light signals by using PIC. In addition, this project will also explain how to control the speed of a motor by using the pulse width modulation technique and vary the duty cycle in proportion to the temperature reading and light intensity [1, 2, 3].

2. PCB components and PIC boards

Microcontroller differs from a microprocessor in many ways. First and the most important is its functionality. In order for a microprocessor to be used, other components such as memory, or components for receiving and sending data must be added to it. In short that microprocessor is the heart of the computer. On the other hand, microcontroller is designed to be all of that in one. No other external components are needed for its application because all necessary peripherals are already built into it. Thus, the time and space needed to construct devices are saved. Figure 1 describes the set of the microcontroller.

The PICDEM Mechatronics of figure 2 is intended to be a learning tool for individuals interested in Mechatronic design. Mechatronics refers to integrating electronic controls into mechanical systems or replacing mechanical components with electronic solution. PIC microcontrollers are ideal for use in Mechatronic systems due to their small size, high efficiency, speed and abundance of peripheral configurations. Mechatronic systems range in complexity from a simple electromechanical switch with output multiplexing to the complex stabilizer control of a supersonic jet. The appliance and automotive markets are notable markets in which mechatronic designs are becoming more common place. This chapter introduces the PICDEM™ Mechatronics development board.

The PICDEM™ Mechatronics Development Kit contains the following items:

1. The PICDEM Mechatronics Printed Circuit Board (PCB) with motors, figure 3.
2. Pre-programmed PIC16F917 microcontroller device
3. 10 wire jumpers
4. 8 2-pin shunts
5. CD-ROM including:
 - PICDEM™ Mechatronics Users Guide
 - Workshop-in-a-Box presentation for training students on the board (Speaker notes included)
 - Data sheets for the PIC16F91X, PIC16F685/687/ 689/690 and motors
 - Application notes and other technical documentation
6. Sample kit including a PIC16F690 and PIC12F509 device
7. Microchip screwdriver
8. Registration Card

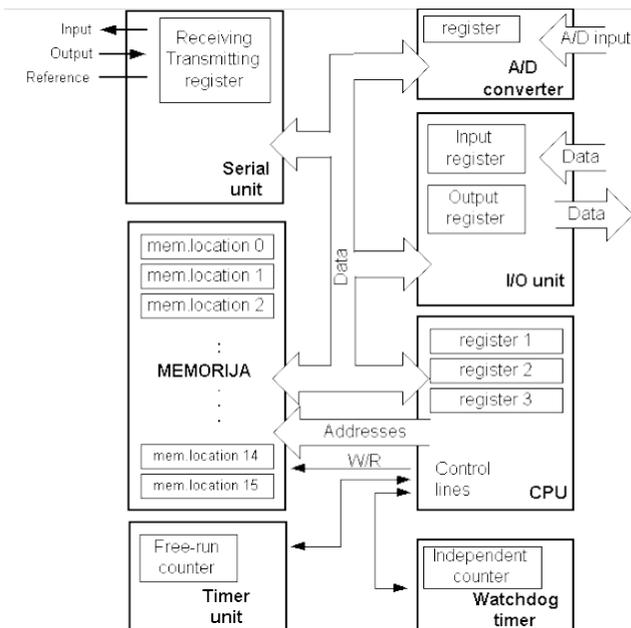


Figure 1: The Complete Set and Diagram of a Microcontroller

The MPLAB ICD 2 is an in-circuit debugger. It debugs the code by allowing the programmer to stop the part and step through the code. While stepping, the debugger uploads critical registers from the microcontroller and displays these registers in the MPLAB IDE.

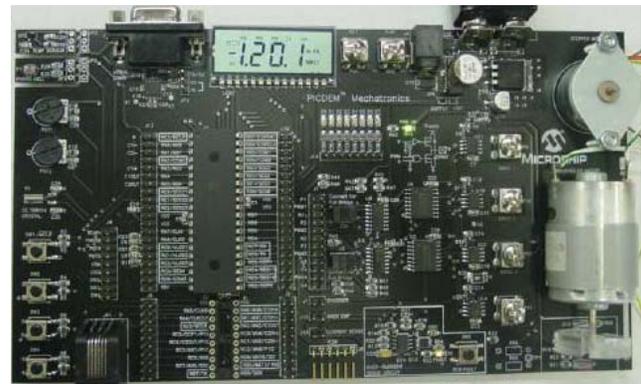


Figure 2: The PICDEM Development Board

This allows the programmer to see what the part is doing and find why it did not working as expected. Besides, MPLAB ICD 2 is used to program or to inject the assembly code into the PIC microcontroller. Figure 4a shows the ICD 2 device

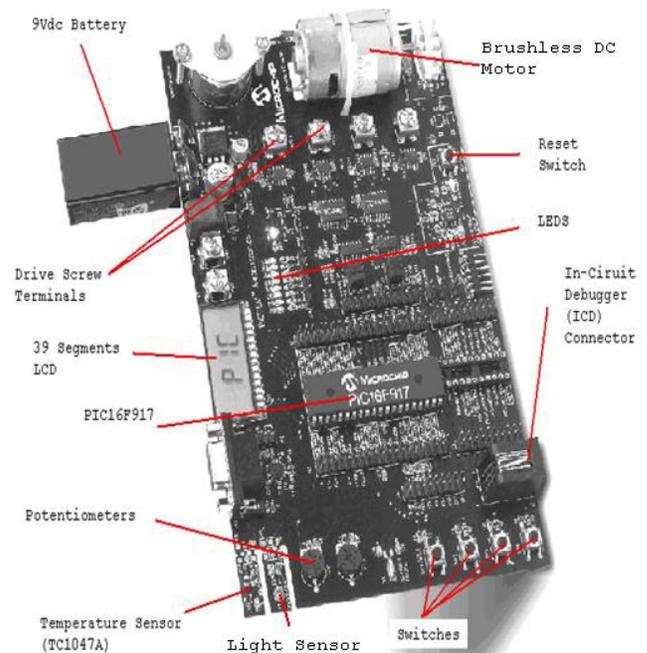


Figure 3: The PICDEM Development Board



Figure 4: a) In-Circuit Debugger (ICD 2), b) MPLAB IDE Icon

MPLAB Integrated Development Environment (IDE) of figure 4b is a free integrated compiler system based toolset for the

development of embedded applications employing Microchip's PIC. The MPLAB IDE runs as a 32-bit application on Microsoft Windows, and includes several free software components for application development, hardware simulation and debugging. It also could best be described as developing environment for some standard program language that is intended for programming a PC computer. Both Assembly and C programming languages can be used with MPLAB.

3 Control System Design

This section presents speed control by using ABB ACS800 variable speed drive according to change in temperature and light signals. Flowchart of this experiment is shown in figure 5. The proposed system consists of 2 inputs which are on/off switch, and the temperature/light sensor. The output of PORTD consists of 4 logic control. This output signal will control the start/stop of the ABB ACS800 speed drive, as well as the 7 constant speeds. Basically, this system works by reading the value of the temperature/light signal from the temperature/light sensor. It will then convert the temperature/light value from analog to digital value. This value will be displayed on the LCD screen. The result that the PIC gets from the conversion part will be used to send (logic output) signal to ABB ACS800 speed drive.

speed drive. When the switch is pressed, the motor should start and the whole program should ready and read the temperature from the sensor immediately. For the first stage, if the temperature is more than 46 degree Celsius, it will ask the ABB speed drive to go to speed number 1. If the temperature remain unchanged (below 46 degree Celsius), the motor will remain at zero speed and the system will keep getting the temperature reading constantly. Same goes when the temperature is more than 48 degree Celsius, it will advance to speed 2. Else, if the temperature remains lower than 48 degree Celsius but more than 46 degree Celsius, the speed will remain at speed 1. The whole system once again, will keep checking the current temperature and send appropriate logic signal to ABB speed drive. For safety purpose in real life application, this project proposed that if the temperature is more than 64 degree Celsius, the whole system will shut down. It will go back to the first state which is to check the switch whether it is on or off. If the temperature remain more than 64 degree Celsius, the whole system will not on and remain off until the temperature is lower than 64 degree. In other case, the same principle applies for running the motor accordingly to the light signals/intensity. If the light intensity is more than 46, it will ask the ABB speed drive to go to speed number 1. If the value remain unchanged (below 46), the motor will remain at zero speed and the system will keep getting the light intensity reading constantly. Same as the temperature control, if the light intensity is too high (in this case more than 64), the whole system will shut down and return to the first state again.

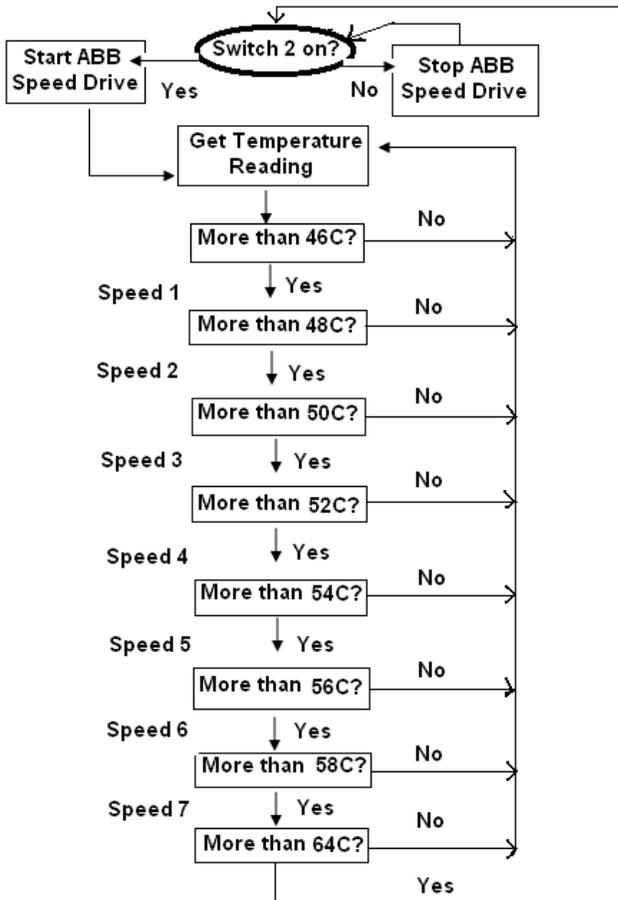


Figure 5: The Flowchart

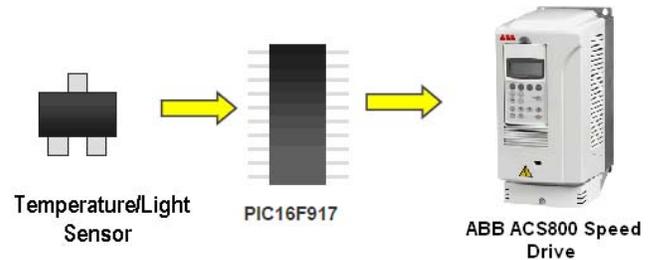


Figure 6: The Proposed System Diagram

In this work, specific equipment and apparatus are connected. Figure 3.2 shows major equipments used

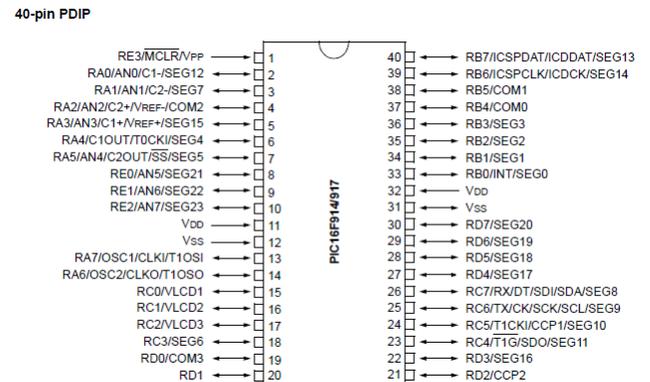


Figure 7: The PIC16F917 Diagram

As seen from figure 5, the system waits for switch 2 to be pressed. If nothing happen, no signal will be send to the ABB

PIC16F917 of figure 7 is used as the microcontroller to control both pulse width modulation and the ABB ACS800 variable

speed drive. PIC16F917 has 40 pin input and output. The advantage of using this kind of microcontroller is that it has built in 8MHz oscillator and analog to digital converter. In addition, it can drive up to 98 LCD segment, has 2 PWM modules, flash program memory and operating voltage of range from 2V to 5.5V.

On the PIC16F917, the program memory is 14 bits wide by 8192 words. The program and data memory are separated. Program memory is the memory that contains the source code. It is actually the numerical equivalence of source code. This memory is fixed once it is programmed into a PIC microcontroller. Data memory is dynamic and constantly changing. This memory contains variables and information about the current state of input/output pins on the PIC microcontroller. The temperature sensor TC1047A of figure 8a are linear Voltage output whose output voltage is directly proportional to the measured temperature. It can accurately measure temperature from -40°C to +125°C. With this kind of temperature sensor, the supply voltage can vary between 2.5V and 5.5V

The output voltage range for this device is typically 100mV at -40°C, 500mV at 0°C, 750mV at +25°C and 1.75 at +125°C. A 10 mV/°C voltage slope output response allows for a predictable temperature measurement over a wide temperature range

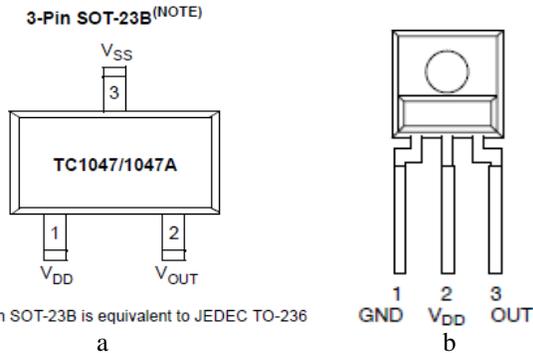


Figure 8: a) TC1047A temperature sensor
b) TSL12S Light Sensor

The Light Sensor TSL12S are linear Voltage output whose output voltage is directly proportional to the measured light intensity. With this kind of light sensor, the supply voltage can vary between 2.7V and 5.5V. The sensor responds to light in the range of 320nm to 1050nm wavelength

ABB ACS800 speed drive is an AC drive technology developed by Asea Brown Boveri (ABB), in Switzerland. ACS800 consists of direct torque control (DTC), a revolutionary motor control method that allows direct torque control to motor magnetizing flux and motor torque. There is no modulator and no requirement for tachometer or position encoder to feed back the speed or position of the motor shaft, down to zero speed. The ACS800 features built-in preprogrammed application macros for configuration of parameter such as inputs, outputs and signal processing. There are five standard macros and two user macros are available for user to choose. For convenient, the start-up assistant will guide

the user through the start-up procedure to help the user to feed the requested data (parameter values) to the drive.

In this project, the ABB ACS800 setting is simple. This motor only takes the signal given by the output PORTD of PIC16F917 and instructs the ABB motor to follow the logic input signal. It means that only the sequential control is needed and necessary at this time.

This speed drive handles 7 constant speeds which are speed 1 (200rpm), speed 2 (400rpm), speed 3 (600rpm), speed 4 (800rpm), speed 5 (1000rpm), speed 6 (1200rpm) and final speed (1300rpm). This ABB ACS800 has only maximum speed of 1350rpm and that is why this particular project chose 1300rpm as the maximum speed (speed 7). Table 1 illustrates the 7 speeds

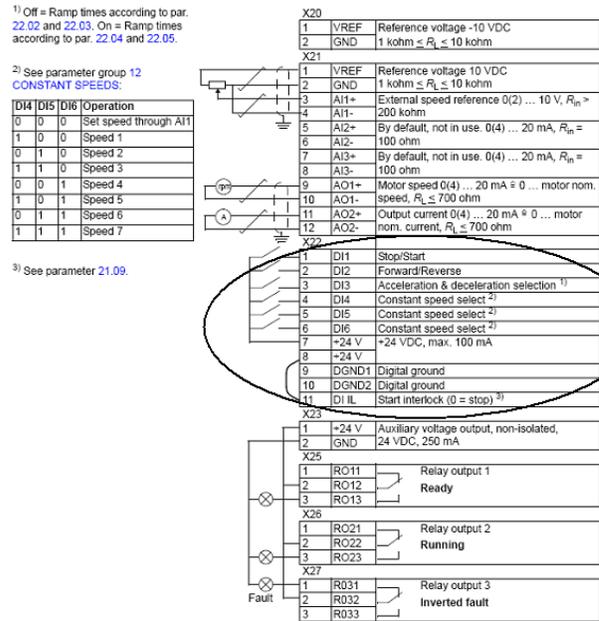


Figure 9: The Sequential Control of ABB ACS800

As shown in figure 9, these are the sequential control in ABB ACS800 speed drive. It only controls start/stop of the motor, the motor direction, and the seven constant speeds. This project only proposed to control start/stop of the motor, and the 7 constant speeds. To explain the sequential control on ABB speed drive, this part here is the logic inputs which take signal (zeros and ones) from any logic control device such as PIC16F917. It only reads zeros (0) and ones (1) signal. In addition, every single terminal port of this ABB ACS800 speed drive panel only accepts 24volts as on (1) otherwise below than that is consider to be off (0).

Table 1: The logic 7 Constant Speeds Input of ABB ACS800

DI4	DI5	DI6	operation
1	0	0	Speed 1 = 200 rpm
0	1	0	Speed 2 = 400 rpm
1	1	0	Speed 3 = 600 rpm
0	0	1	Speed 4 = 800 rpm
1	0	1	Speed 5 = 1000 rpm
0	1	1	Speed 6 = 1200 rpm
1	1	1	Speed 7 = 1300 rpm

4. PICDEM Circuit Connection

Figure 10 shows the schematic connection of temperature/ light sensor and switch as inputs and PORTD 1, 4, 5, 6 as an output to start/stop the ABB speed drive and also to vary the motor speed according to the temperature/light levels. The complete connection of the PICDEM board is shown in the figure 11. Jumper configuration:

Temp to RA0 (for light sensor experiment, just simply move the TEMP jumper to LIGHT jumper as shown in green color cable in figure 11)

SW2 to RA4

RD1 to DI1

RD6 to DI4

RD5 to DI5

RD4 to DI6

The Supply Voltage for this board is a 9V battery.

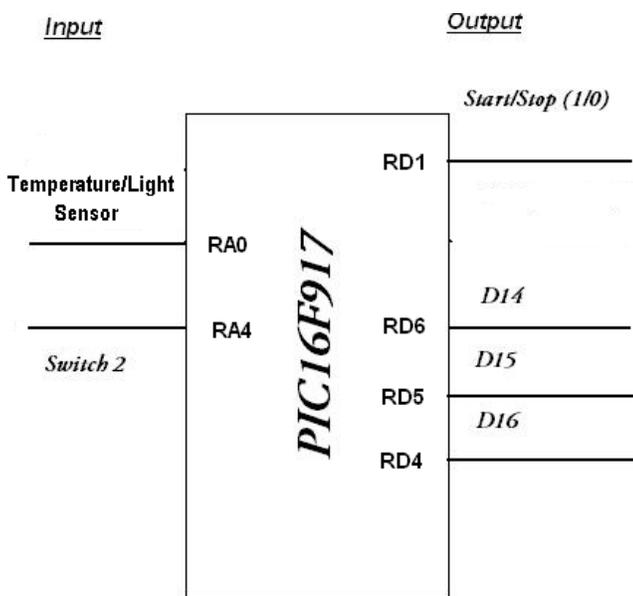


Figure 10: The Schematic diagram of the PIC Connection

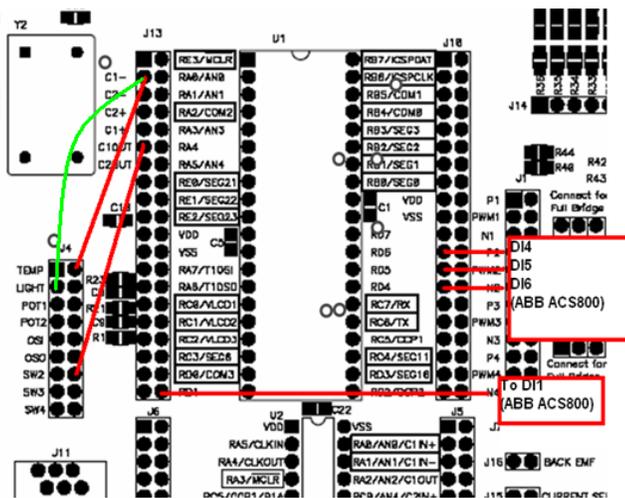


Figure 11: The Schematic diagram of the PICDEM Connection

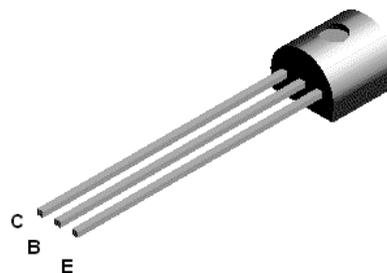


Figure 12: The 2N3904 Transistor

Since the PICDEM (as well as the output from the PIC16F917) only produce 5V output, it will not send enough signals to ABB input panel. To settle this problem, transistor is needed to amplify and boost up the voltage from 5V to 24V figure 12. The connection of the transistor is shown in figure 13. In that particular circuit connection, the transistor (NPN) acts as a switch and inverter (NOT GATE) at the same time. The output signal (voltage) is the inverse of the input signal. When the input is high (+Vs) the output is low (0V). When the input is low (0V) the output is high (+Vs). This explain why the ABB ACS800 speed drive reads the ones (1) from the PIC16F917 as zeros (0) and vice versa.

Note that the four transistors are connected in parallel. The first transistor will receive a signal from the PORTD 1 (RD1) from the PIC. The second transistor will receive a signal from PORTD 6 (RD6), third transistor receives signal from PORTD 5 (RD5) and fourth transistor receives signal from PORTD 4 (RD4). This circuit will be powered up by using the channel number 7 in the ABB ACS800 sequential control and will be grounded at the channel number 10

Since the transistor acts as switch to amplify the input voltage to 24V, the new inverted logic shown in table 2 is used in programming the PIC16F917 logic output.

Table 2: The New Inverted logic input to ABB ACS800 7 Constant Speeds

DI4	DI5	DI6	operation
0	1	1	Speed 1 = 200 rpm
1	0	1	Speed 2 = 400 rpm
0	0	1	Speed 3 = 600 rpm
1	1	0	Speed 4 = 800 rpm
0	1	0	Speed 5 = 1000 rpm
1	0	0	Speed 6 = 1200 rpm
0	0	0	Speed 7 = 1300 rpm

Certain simple procedure need to be followed in order to set the desire 7 constant speeds. The first step is to on the ABB ACS800 press the parameter button (“PAR” on the ABB remote). Find and select a parameter group called constant speed (12 Constant Speed). Then, find the parameter number 01 which says about the constant speed selection. Proceed by pressing enter button on the remote. In this mode, find parameter of [DI4, 5, 6] then press enter. Here, the 7 constant speed setting can be done

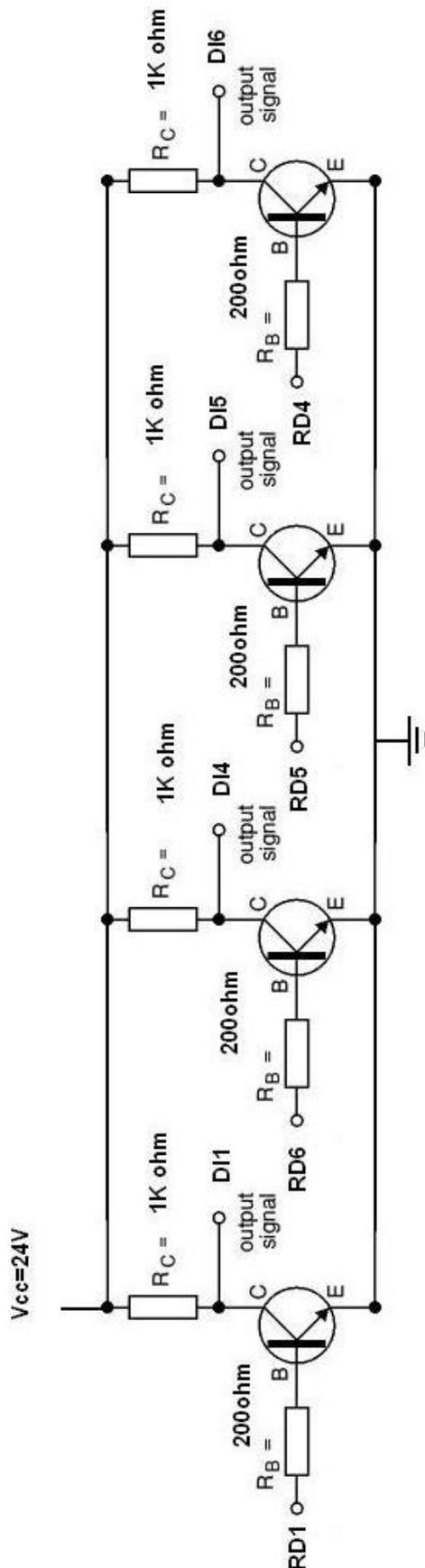


Figure 13: The Circuit Connection of the Transistor

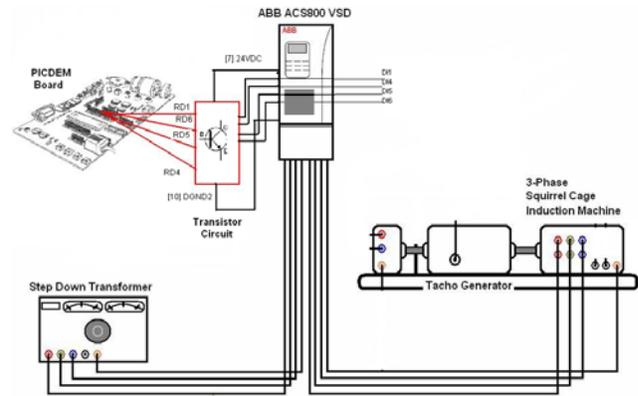


Figure 14: Complete Circuit Connection of the PICDEM Board, Transistor circuit and ABB ACS800 Speed Drive

The connection of the overall circuit is shown in figure 14. The step down transformer consists of 4 outputs which are connected to the ABB ACS800 channels. These are L1, L2, L3 and PE. Connect L1 to U1 (ABB ACS800), L2 to V1, L3 to W1 and finally PE to ground

For the 3-phase Squirrel Cage Induction Machine, delta connection needs to be done. On the motor, connect W2 with U1, U2 with V1, and V2 with W1. Once the connection is done, connect W2 to U2 (ABB ACS800), U2 to V2 (ABB ACS800), V2 to W2 (ABB ACS800) and ground to ABB ACS800 ground.

5. Programming the PIC

The program language for the PIC Microcontroller uses 35 words (instructions) [7]. In this project, PIC has been programmed by using the assembly language. Assembly languages are a family of low-level languages for programming computers. It implements a symbolic representation of the numeric machine codes and other constants which are needed to program a particular CPU architecture. This representation is usually defined by the hardware manufacturer, and is based on abbreviations (called mnemonics) that help the programmer remember individual instructions, and registers.

Programs written in assembly language must be translated into a "language of zeros and ones" in order to understand it for a PIC microcontroller. "Assembly language" and "assembler" are two different notions. The first represents a set of rules used in writing a program for a microcontroller, and the other is a program on the personal computer which translates assembly language into a language of zeros and ones. A program that is translated into "zeros" and "ones" is also called "machine language". Physically, "Program" represents a file on the computer disc (or in the memory if it is read in a microcontroller), and is written according to the rules of assembler or some other language for microcontroller programming. When writing a program, certain rules must be followed in order to reach a desired effect. A Translator interprets each instruction written in assembly language as a series of zeros and ones which have a meaning for the internal logic of the microcontroller. The place where this translation of assembly language is found called an "execution" file. Apart from that, "HEX" file comes from a hexadecimal representation of that file. Once it is generated, the execution file is read in a microcontroller through a programmer.

6. Outcome graphs

Once the project is implemented, a test is made to find the performance of the system. Input temperatures from 40 °C to 70 °C (by using hair dryer) have been carried out for practical investigation on temperature sequential speed control test. A shadow of the finger covers the light sensor to change the light intensity is used, and consequently the speed changes. Tables 3 and the corresponding figure 15, show the change in speed according to the temperature and light. A reverse operation of the induction motor is also employed in the program of PIC.

Table 3: Motor Speed vary with the temperature or light intensity

Temperature °C & Light intensity cd	Speed rpm
0 – 46	0
46 – 48	200
48 – 50	400
50 – 52	600
52 – 54	800
54 – 56	1000
56 - 58	1200
58 – 64	1300
> 64	0

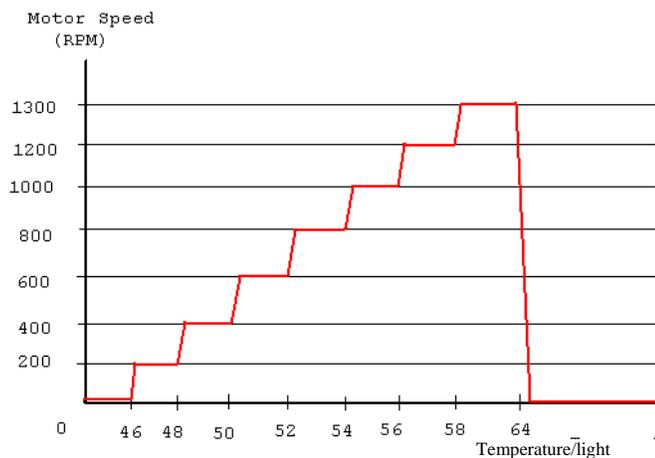


Figure 15: Motor Speed Vs Temperature/light

7. Conclusion

This work has devoted a lot of practical experience and valuable theoretical knowledge on motor speed control, temperature and light measurement as well as PIC programming skill. It has been proven that motor can be controlled by using various techniques such as using the PWM mode as well as injecting logic signal from the PIC to the ABB ACS800 variable speed drive.

The use of PWM to control logic outputs (0 and 1) to the ABB ACS800 speed drive using PIC is important. The PIC programs which give outputs of zeros and ones can easily control the channel DI1, DI4, DI5 and DI6 of ABB ACS800 speed drive. In addition, there are no difficulties in changing the parameter of the temperature value in the program code as well as the 7 constant speed of the ABB speed drive.

Similarly, in temperature sequential control system, PIC16F917 is able to convert the analog signals given by the temperature sensor and controls the motor rotational speed correspond to the temperature reading. In similar way, a same principle applies for converting the analog signals given by the light sensor. The Analog to Digital conversion is easily done in PIC programming since it is not hard to understand the Assembly language.

PIC is a friendly user system and a better choice of device in designing an automation control system and controlling industrial machineries. Overall, the objectives and requirements of this project have been fulfilled.

Acknowledgment

Author wishes to express his deepest gratitude to Malaysian government for funding this project through the science fund of MOSGTI organization of research and development. Special thank goes to the R&D dean of Curtin Sarawak for his help on the proposal and constant encouragement to obtain the grant for developing this work

References

- [1] Adeline Voon, J.Y. (2006) „Sequential control by speed drive for three phase induction motor , Degree Thesis, Curtin University of Technology, Sarawak.
- [2] Barsoum, N.N. (2005). Induction machine – machine construction and concept of operation, Retrieved April 7, 2009, from <http://www.engineeredu.org/courses/units/em/em2/21.pdf>
- [3] Cheong, Z.X. (2007). „Implementation of temperature sequential controller on motor speed drives , Degree Thesis, Curtin University of Technology, Sarawak.
- [4] Ibrahim, D. (2008). Advanced PIC Microcontroller Projects in C, 1st edition. Newnes
- [5] Jasio, L.D. (2008). Programming 32-bit Microcontrollers in C, 1st edition Newnes.
- [6] Katzen, S. (2000). The Quintessential PIC Microcontroller, 1st edition. Springer-Verlag.
- [7] Matic, N. (2003). PIC Microcontroller, 1st edition. Prentice Hall.
- [8] Morton, J. (1998). The PIC Microcontroller, Your Personal Introductory Course, 1st edition. Newnes. 104
- [9] Predko, M. (2008). Programming and Customizing the PIC Microcontroller, 3rd edition. Tab Electronics.
- [10] Smith, D.W. (2006). PIC in Practice, 1st edition. Newnes
- [11] Spasov, P. (1999). Microcontroller Technology, 3rd edition. Prentice Hall