SGS-THOMSON MICROELECTRONICS APPLICATION NOTE

Fuzzy Vacuum Cleaner Using ST6220 and fuzzyTECH™ ST6 Explorer

CENTRAL APPLICATION LABORATORY, SINGAPORE

INTRODUCTION

For the past 20 years, the home environment has changed drastically and with the rise of living standards, the consumers' need for home cleaning has switched from a simple mop or scrub to a more sophisticated mode. A vacuum cleaner that is able to do the cleaning based on different characteristics of floor surfaces will be very desired for today's market.

This new requirement actually represents the first and most visible group of the next generation of consumer products based on a new control-fuzzy logic. Fuzzy logic is a relatively new technology that enables machines and products to operate more efficiently and independently by processing information similar to the way people do.

This note describes a universal motor power control implemented on a standard microcontroller running software using the fuzzy logic concept. The different stages of development of the motor power control for a vacuum cleaner are described with the ST6220 microcontroller and a fuzzy logic development tool, the "fuzzyTECH™ ST6 Explorer Edition".

TABLE OF CONTENTS

1 WHY FUZZY LOGIC?

Fuzzy logic offers a number of benefits to users, especially in control applications that typically require a great deal of human supervision or intuition. Fuzzy logic also offers a control alternative for these applications that, due to system complexity, could not be controlled satisfactory using conventional control strategies.

The primary benefits of fuzzy logic are:
1) natural language programming;
2) ability to process vague or incomplete data;
3) shortened program development time;
4) ability to incorporate human subjectivity and intuition;
5) ability to process multiple inputs simultaneously.

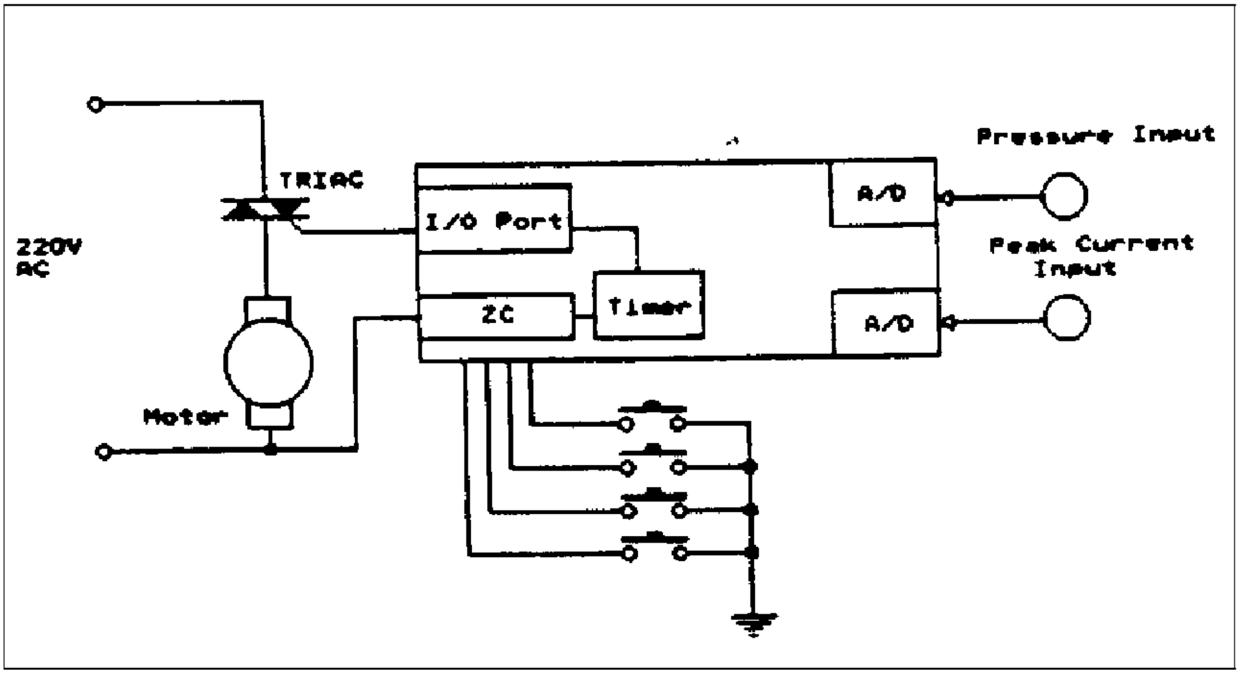
In a conventional vacuum cleaner, power control is designed based on a sample floor whereby cleaning on different floors depends largely on the user's experiences and intuition. This could be very troublesome and inefficient. However, fuzzy logic based on fuzzy set theory, allows the experts' knowledge and experiences of cleaning floors (varying from carpet cleaning to floor cleaning) to be expressed in words. This is much easier to understand and modify than mathematical equations.

With the fuzzy logic implemented in a vacuum cleaner, it will regulate its suction power to an optimum level according to the type of floor.

2 PRACTICAL APPLICATION

Figure1 describes the fuzzy logic motor power control application. The universal motor is supplied in DC mode. The AC voltage of the motor is controlled by the timing techniques through a snubberless triac. This triac can operate with low gate current and can be directly triggered by the ST6220 microcontroller with an onboard timer, while still maintaining a high switching capability.

Figure 1. Block Diagram



The output power of the motor is controlled by the phase delay of the triac drive. This delay is referred to the zero crossing of the line voltage which is detected by the means of a connection to the mains neutral.

The onboard timer is precisely tuned in order to obtain 10 msec for 50Hz (or 8.3 msec for 60Hz) delay between two gate pulses. The delay measurement is done periodically and synchronised with respect to the zero-crossing of the line voltage. As a result, the triac is driven symmetrically in both phases so that noise in the transformer is reduced.

3 APPLICATION OF FUZZY LOGIC CONCEPT

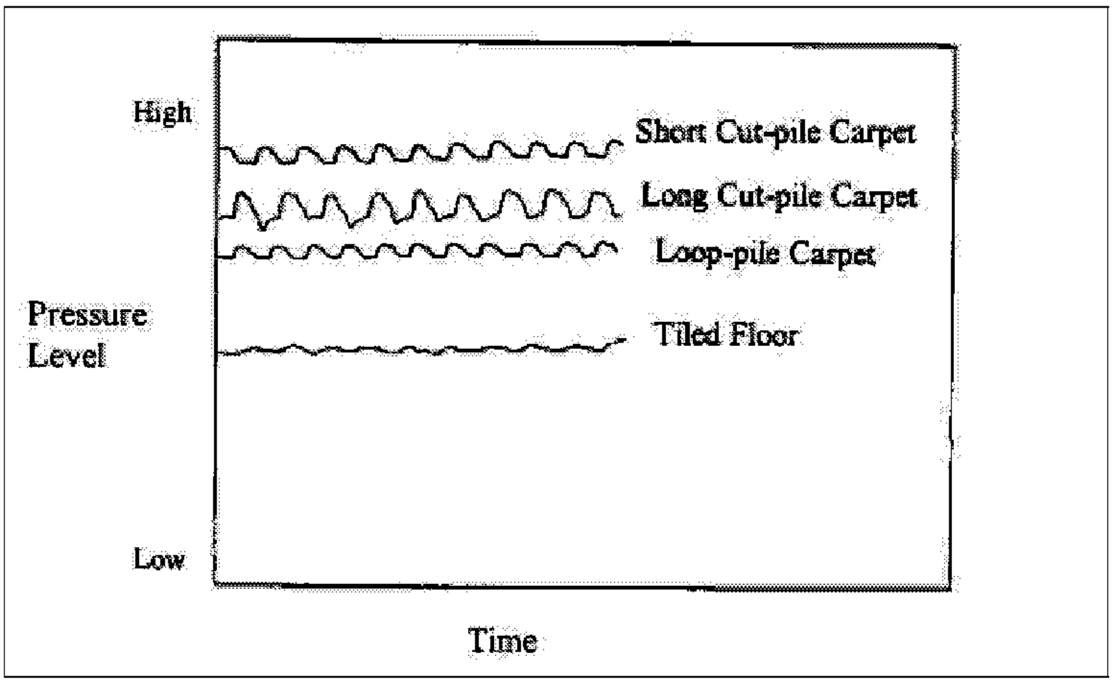
Fuzzy set theory recognises that very few crisp sets actually exist, it allows partial set membership; it allows gradual transitions between being fully a member of a set and not fully a member of the set. This characteristic allows the working path from truth to falseness to be gradual and also implicitly in such a way that something can be simultaneously partially true and partially false. This is a more powerful way of working than making assumptions and forcing our view of the world to be black or white.

3.1 Input/Output Variables Definitions

The input variables of the fuzzy logic controller used to estimate the motor power variations are the pressure level with respect to atmospheric pressure and the peak current drawn by the DC motor at the soft floor brush.

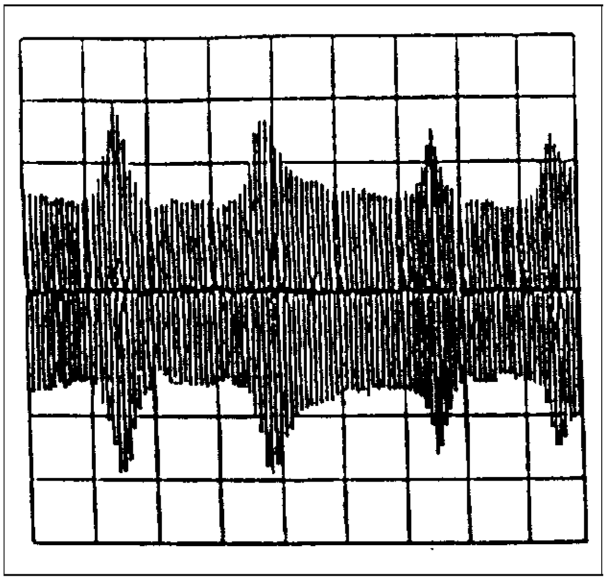
- the variable pressure is measured with reference to the atmospheric pressure. Its maximum pressure level will be obtained when the extension tube is sucked. The determination of floor type is based on the results obtained from research, as shown in Figure 2.

Figure 2: Results of Pressure Level Obtained vs different floor types

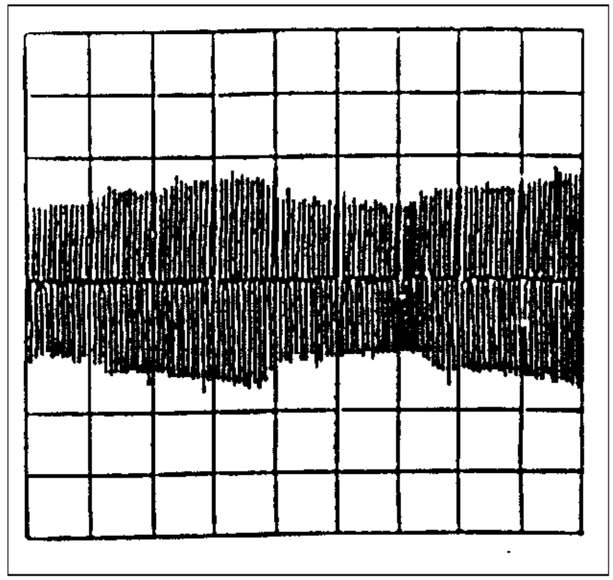


- variable peak current is used to determine the different types of carpets. The current is measured when the brush, driven by the DC motor, gets into contact with the carpet. While vacuuming different types of carpets, the following phenomenum of the current detected were obtained through some experiments:

1. Long cut-pile carpet



2. Short cut-pile carpet



3. Loop-pile carpet

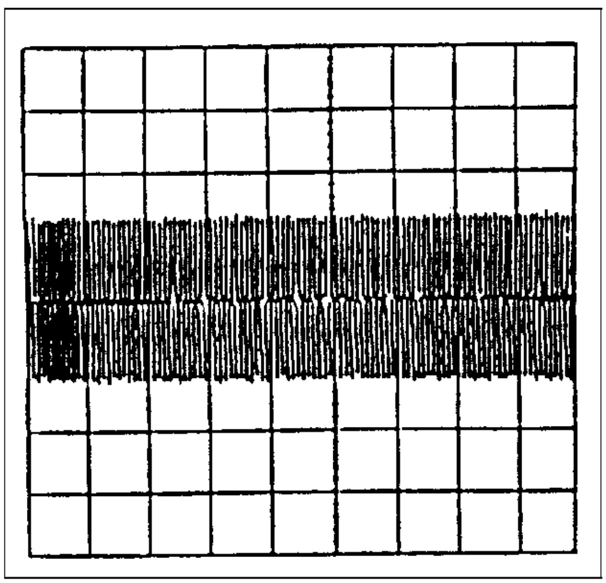


Figure 2a: Waveforms of Peak Current while cleaning on different types of carpet

The output variable is the firing angle that controls the driving of the triac, which directly determines the suction power variation of the vacuum cleaner. The minimum firing angle in our application is 80°, because any value that is less than 80° will not be sufficient to supply the minimum suction power for operating the vacuum cleaner.

3.2 Fuzzy Rules

In fuzzy logic, the different values for a given linguistic variable represent concepts (called terms), not numbers. With the 2 linguistic input variables, the fuzzy rules are defined in the following table, together with their respective linguistic terms, as to obtain the expected results for the output variable.

	Pressure Level			
Peak Current	Small	Slarge	Large	Vlarge
Off	Small*	Small*	Small*	Small*
Low	Sbig*	Sbig*	Vbig*	Small*
Shigh	Sbig*	Big*	Vbig*	Small*
High	Vbig*	Vbig*	Vbig*	Small*
Vhigh	Vbig*	Vbig*	Vbig*	Small*

* result generated is for the output variable (firing angle of triac).

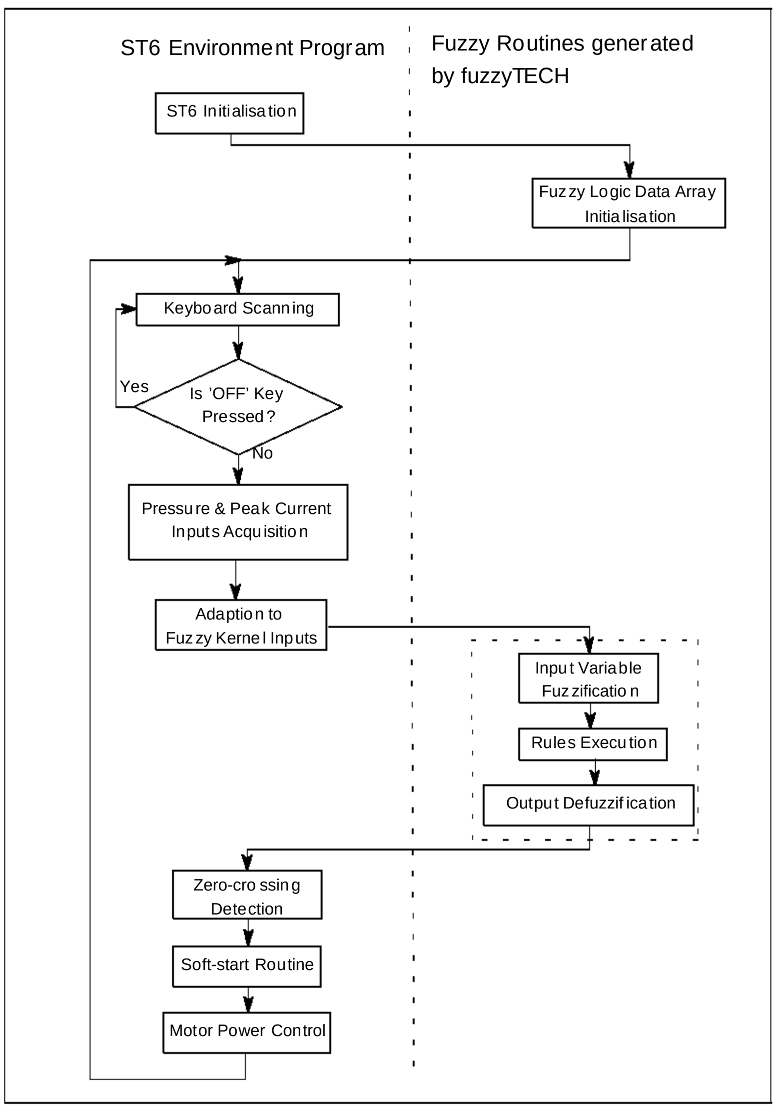
4 FUZZY LOGIC APPROACH WITH DEVELOPMENT TOOL

The program flow chart in Figure 3 shows that fuzzy logic development can be divided into two main parts: the fuzzy logic application itself and the microcontroller environment program.

- the fuzzy logic module consists of ST6 executable code generated by the development tool "fuzzyTECH ST6 Explorer Edition". This part consists of fuzzification of the input variables, execution of the activated rules and defuzzification that produces the output variable.
- the environment program consists of microcontroller initialisation, pressure and peak current acquisition, input variable adaptation to fuzzy logic kernel code values, fuzzy logic kernel calling, zero-crossing detection, soft-start routine and motor power control routine.

The " fuzzyTECH ST6 Explorer Edition" used to develop this project covers all steps of a fuzzy logic design from the project, linguistic variables, rules' definitions and off-line optimisation up to the ST6 executable code generation.

Figure 3. Block diagram for Fuzzy Vacuum Cleaner



4.1 Project Editor

The first step when using the fuzzyTECH ST6 Explorer Edition is to define the project by means of the project editor window. The project editor displays the controller structure and allows the designer to directly access definition of the linguistic variables and the formulation of the fuzzy logic rule base. Figure 4 shows the project window of the vacuum cleaner controller.

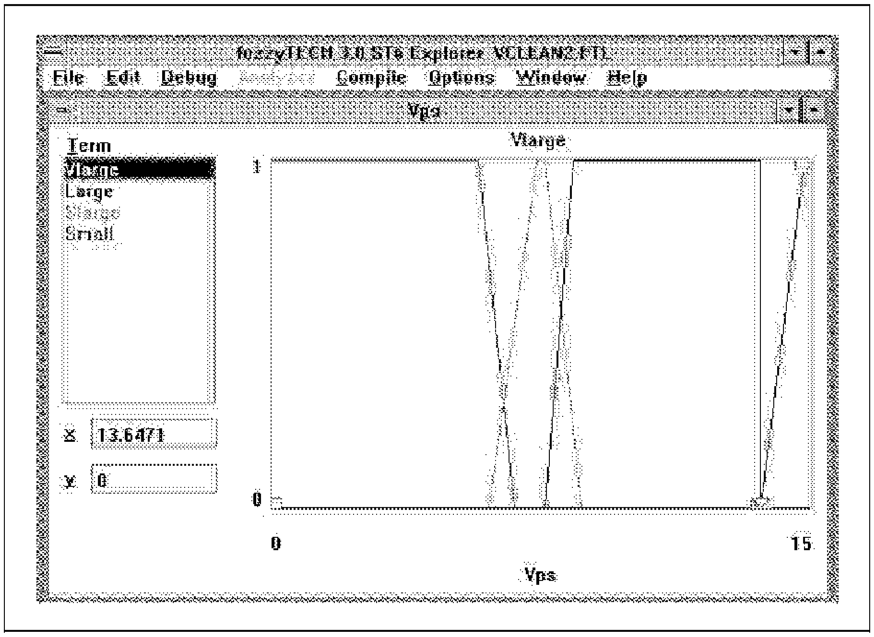
Figure 4. Project Window of Vacuum Cleaner Controller

For every linguistic variable, an interface must be defined. The interfaces for input variables of the fuzzy logic system contain the fuzzification procedure, while the interfaces for the output variables contain the defuzzification procedure. In the ST6 Explorer Edition, only Compute MBF may be used for fuzzification and only the Centre Of Maximum (COM) method can be used for defuzzification.

4.2 Linguistic Variables Definition

The next step of the controller design is the definition of linguistic variables by means of LV list. The graphic interface of the development tool allows the designer to easily create the most suitable linguistic variables and the membership function for the various terms used within a linguistic variable. During the definition of linguistic variables, the development tool allows the designer to define terms from the variable editor's pop up menu. The input/output membership functions shown in Figures 5 & 6 are defined based on the experts' knowledge.

Figure 5. Membership Functions for Pressure(Vps) & Peak



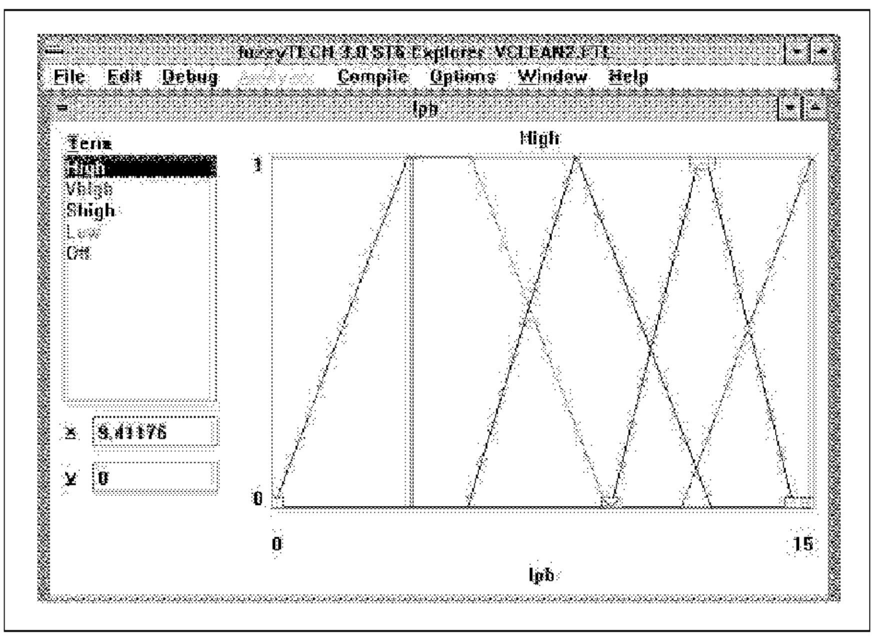
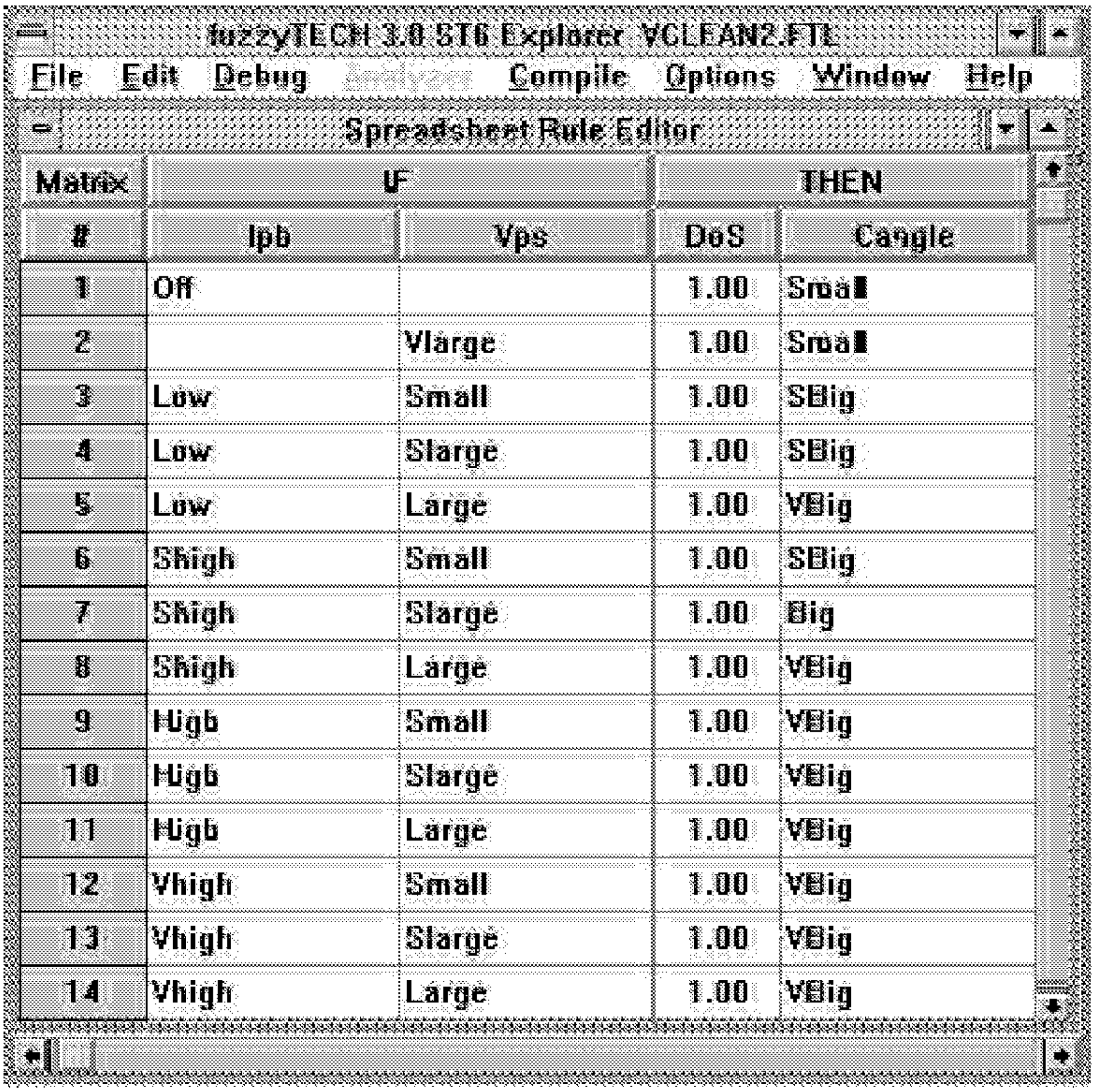


Figure 6. Membership Function for the Fuzzy Output (Cangle)

4.3 Formulation of Fuzzy Rules

Figure 7 shows the fuzzy controller rules. The fuzzy development tool with its spreadsheet rule editor provides easy rules acquisition and also permits the definition of the rule aggregation (MIN and MAX operator are available) and rule composition (PROD operator with degree of support of 0 or 1). In the ST6 Explorer Edition, only the MIN-MAX operator with a parameter of either 0 (MIN) or 1 (MAX) is supported, representing the logical AND or the OR operation. In this application of a vacuum cleaner, the rules aggregation is done using the MIN operator and the degree of support for all rules has been fixed to 1.

Figure 7. Spreadsheet rule editor



Matrix	IF		THEN	
#	Ipb	Vps	DoS	Cangle
1	Off		1.00	Small
2		Vlarge	1.00	Small
3	Low	Small	1.00	SBig
4	Low	Slarge	1.00	SBig
5	Low	Large	1.00	VBig
6	Shigh	Small	1.00	SBig
7	Shigh	Slarge	1.00	Big
8	Shigh	Large	1.00	VBig
9	High	Small	1.00	VBig
10	High	Slarge	1.00	VBig
11	High	Large	1.00	VBig
12	Vhigh	Small	1.00	VBig
13	Vhigh	Slarge	1.00	VBig
14	Vhigh	Large	1.00	VBig

4.4 Adaptation of Real Variables to Fuzzy Kernel Input

Working with variations of input variables, pressure sensing circuitry is needed to convert the pressure input, with respect to the atmospheric pressure, to its corresponding voltage (for linguistic variable Vps). A current-to-voltage conversion circuit is also needed to provide the peak current input for the fuzzy linguistic variable Ipb. All these inputs are fed into the A/D converter input ports of ST6220.

However, a translation has to be done to code the whole range of the input variables within the range [0, 15] while the range of A/D converter can vary from 0 (for voltage level 0V) to 255 (corresponding to 5V).

4.5 Offline Optimisation

In order to achieve the desired system performances, usually a few system optimisation steps are required. Because the optimisation strategy varies with each type of application, the fuzzyTECH ST6 Explorer Edition development tool offers two ways to test and optimise the rules and membership function definitions:

- the interactive debug mode provides the designer with a graphical verification of every step while the system is still being performed.

- the batch mode associates with an input file, generated by the pattern generator, to an output file without displaying the inference process.

All debug modes are mutually exclusive. If in any debug mode, no rule for the current combination of input values fires for an output variable, the value of this linguistic output variable will be set to its default value. In any debug mode, this situation is indicated by a question mark (?) displayed at the right side of the output variable for which no rule is fired.

4.6 Implementation of Assembly Code for ST6

After the system has been optimised and verified, it can be implemented on the hardware platform. Optimised ST6 assembly code is automatically generated by the fuzzyTECH ST6 Explorer Edition. The real time test of the application can be done using an ST6 emulator or an EPROM version of ST6 device.

4.7 Generated Results From fuzzyTECH

The following table shows the variations of the output power of the fuzzy vacuum cleaner with respect to the 2 fuzzy inputs (Vps and Ipb):

Figure 8. An Expert's knowledge

		Ipb														
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	15
Vps	0	86	86	86	86	100	100	100	100	100	100	115	134	147	150	150
	1	86	86	86	86	100	100	100	100	100	100	115	134	147	150	150
	2	86	86	86	86	100	100	100	100	100	100	115	134	147	150	150
	3	86	86	86	86	100	100	100	100	100	100	115	134	147	150	150
	4	86	86	86	86	100	100	100	100	100	100	115	134	147	150	150
	5	86	86	86	86	100	100	100	100	100	100	115	134	147	150	150
	6	86	86	86	86	100	100	100	100	100	100	115	134	147	150	150
	7	86	86	86	86	100	100	104	108	112	116	127	139	148	150	150
	8	86	86	86	86	120	120	120	120	120	129	132	139	148	150	150
	9	86	86	86	86	150	150	150	150	150	150	150	150	150	150	150
	10	86	86	86	86	150	150	150	150	150	150	150	150	150	150	150
	11	86	86	86	86	150	150	150	150	150	150	150	150	150	150	150
	12	86	86	86	86	150	150	150	150	150	150	150	150	150	150	150
	13	86	86	86	86	150	150	150	150	150	150	150	150	150	150	150
	14	86	86	86	86	86	86	86	86	86	86	86	86	86	86	86
	15	86	86	86	86	86	86	86	86	86	86	86	86	86	86	86

5 APPLICATION APPROACH WITH ST6220

5.1 Circuit Description (Hardware)

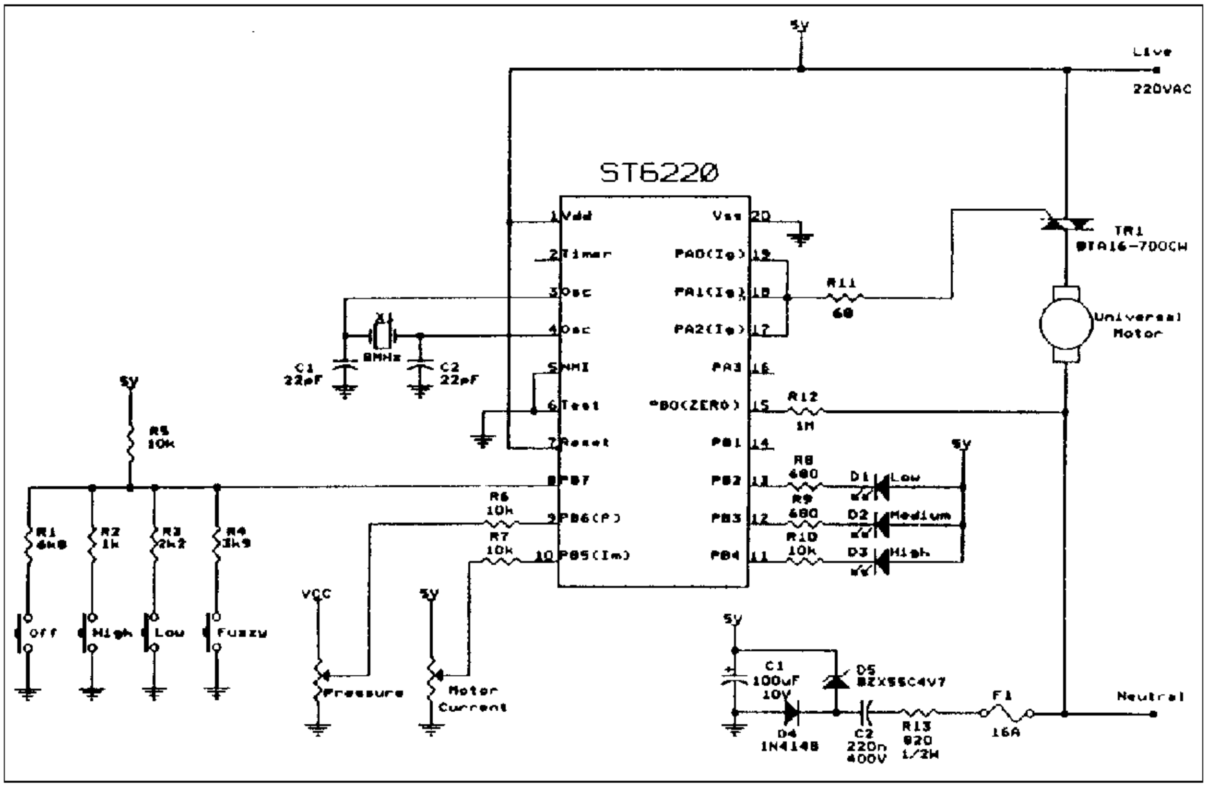
Figure 9 shows the schematic diagram of the motor control board used in the fuzzy vacuum cleaner. A universal motor drive circuit, supplied directly from the 220V mains has been realised using a microcontroller ST6220 (the pin configuration can be found in APPENDIX) and a snubberless triac. The user interface is a series of 4 push-buttons and 2 potentiometers. These 2 potentiometers are used to simulate the inputs of the 2 fuzzy variables. The 3 onboard LED lights are to indicate the power level operated by the vacuum cleaner. The board includes a minimum of components in order to save cost and size. The auxiliary supply is derived from the mains voltage.

The output power and therefore the motor speed are controlled by the phase delay of the triac drive. This delay is referred to the zero-crossing of the line voltage which could be detected through a connection to the mains neutral.

The pulse driving the triac should be short in order to minimise the +5V supply circuit size. The snubberless triac is driven in quadrants QII and QIII with 60 mA gate current provided by three I/O ports of the ST6220 in parallel. This pulse should be sufficiently long to insure that the triac is latched at the end of the pulse. Pulse length can be modified if another triac or motor is used.

In this application, two modes of user interfaces are allowed to operate with the control of the 4 onboard push-buttons. The 'Fuzzy' push-button allows the user to select the option of applying the fuzzy logic concept, as to optimise the power level depending on the 2 fuzzy inputs. While the 'High' and 'Low' buttons provide the user with the choice of overruling the fuzzy application, thus to regulate the suction power with the predefined power level. In addition, the 'Off' button allows the system to be switched off and the power supply could be reassumed by any of the other 3 keys (High. Low and Fuzzy)

Figure 9. Schematic Diagram for the Fuzzy Vacuum Cleaner



5.2 SOFTWARE DEVELOPMENT

For this application of a vacuum cleaner, the system application software is written in two modules: Fuzzy logic application routines which are generated by the fuzzyTECH ST6 Explorer Edition and the ST6 application routines. The source files are assembled separately by AST6 MACRO ASSEMBLER before their generated ST6 object files are combined by LST6 Linkage Editor.

Note, before using the linker facility, the software should be made with LST6 version 3.00 or higher and AST6 version 4.00 or higher. This is because the object file format has been modified. Before linking different object files, all the sources files must be reassembled.

All operating features are contained in a 1600 bytes program (not optimised version). The architecture of the ST6 application routines is modular in order to provide maximum flexibility. The flowcharts for the major routines in the software are shown in the following pages:

- Main routine
- Key scan routine
- Soft-start routine
- ADC average routine
- Timer interrupt service routine
- Input service routine.

Figure 10. Flowchart for MAIN routine

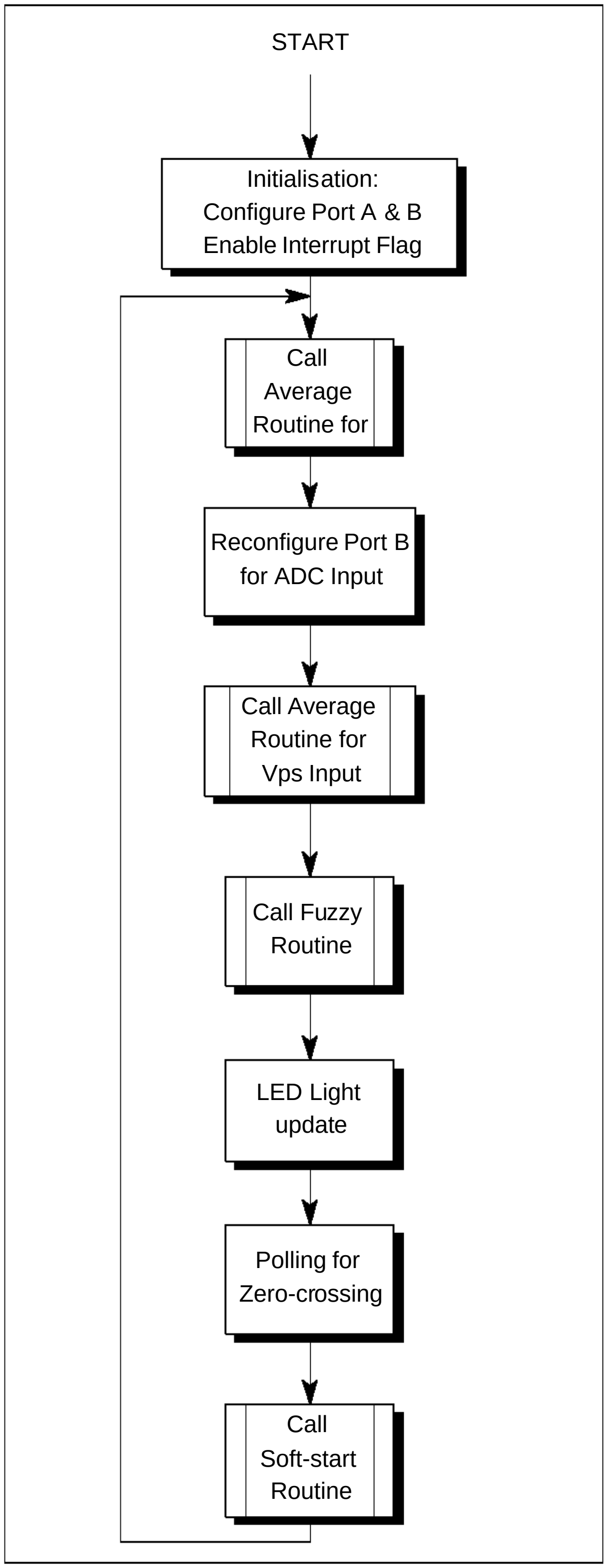


Figure 11. Flowchart for KEY SCAN routine

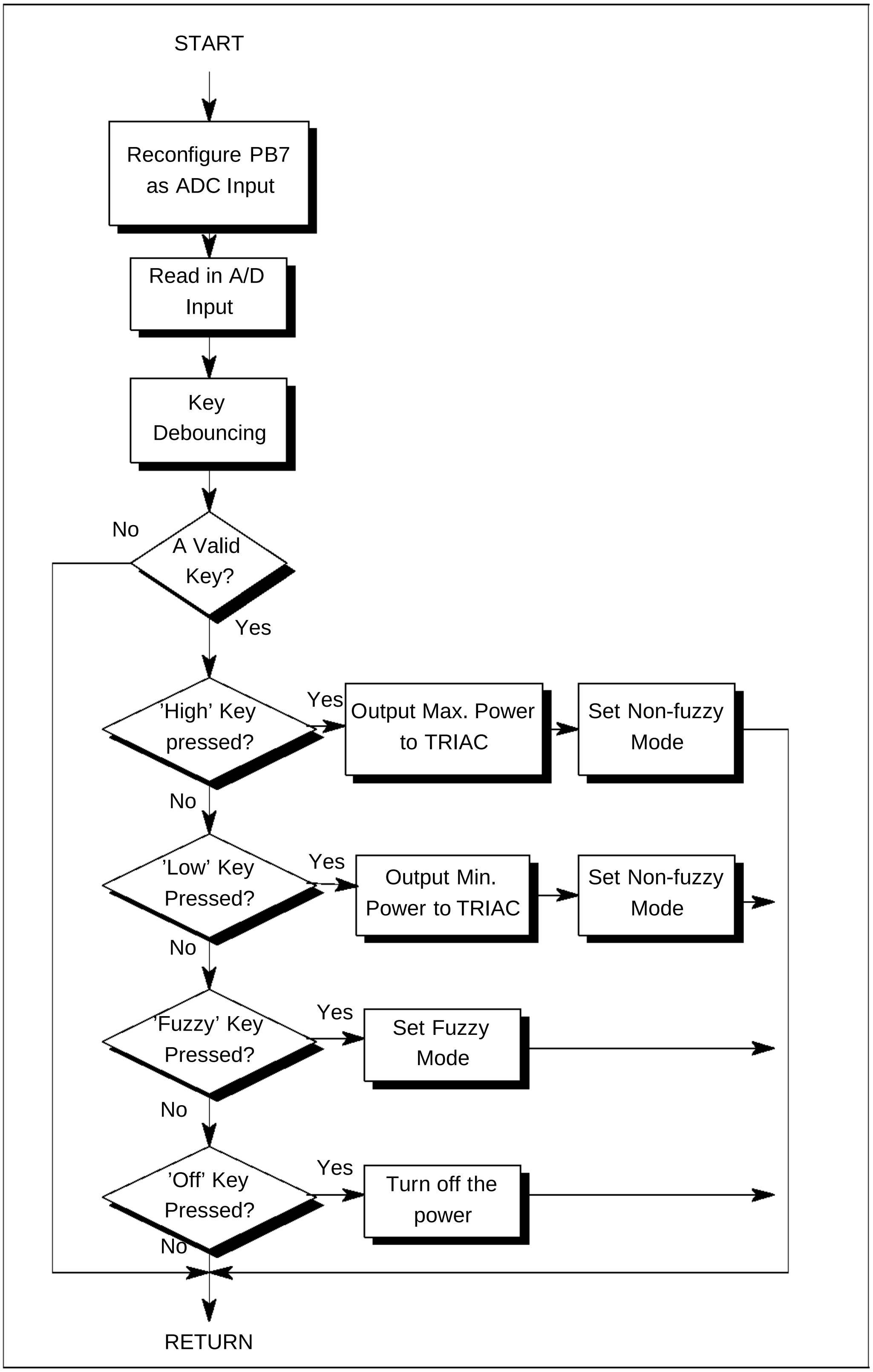


Figure 12. Flowchart for SOFT-START routine

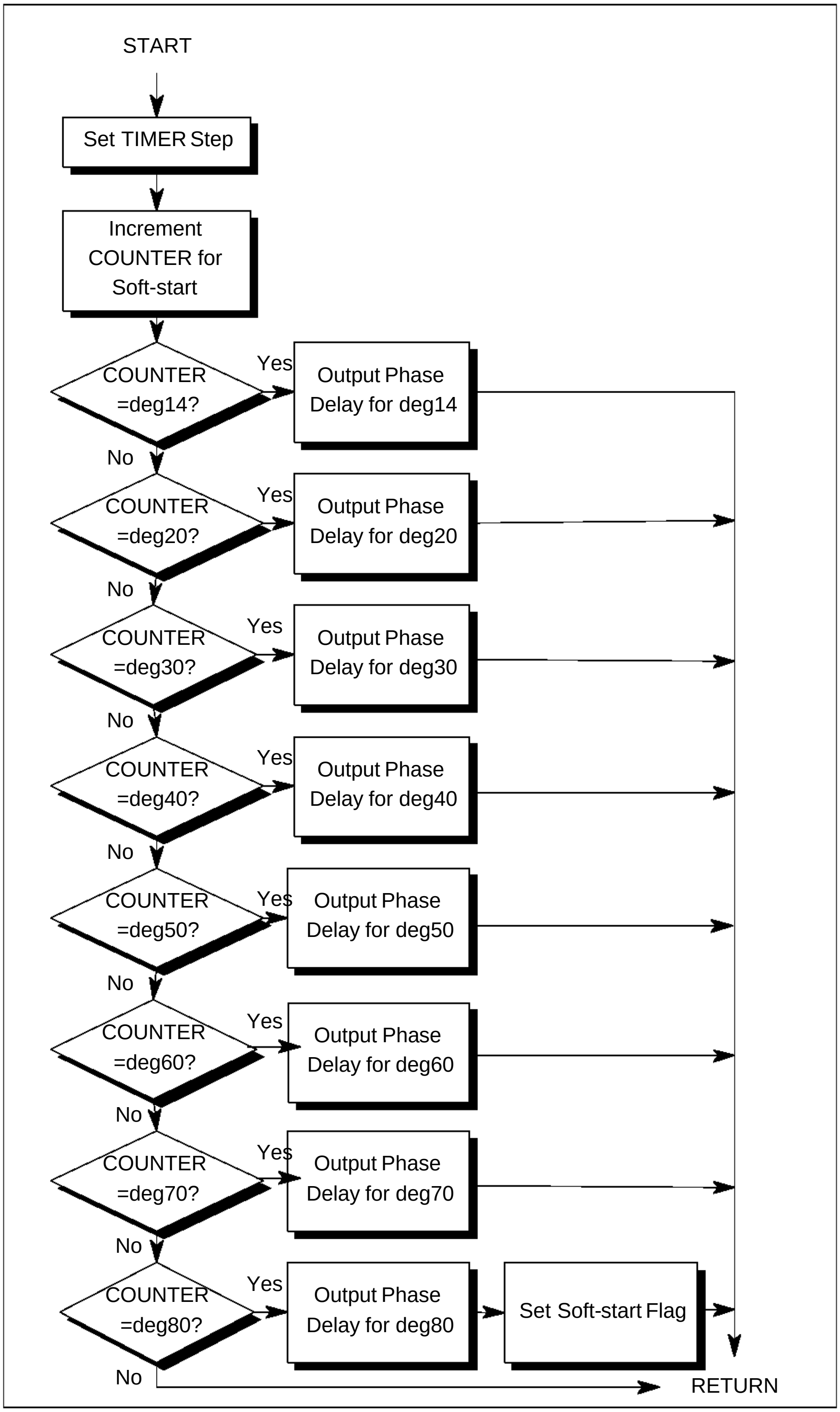


Figure 13. Flowchart for ADC AVERAGE routine

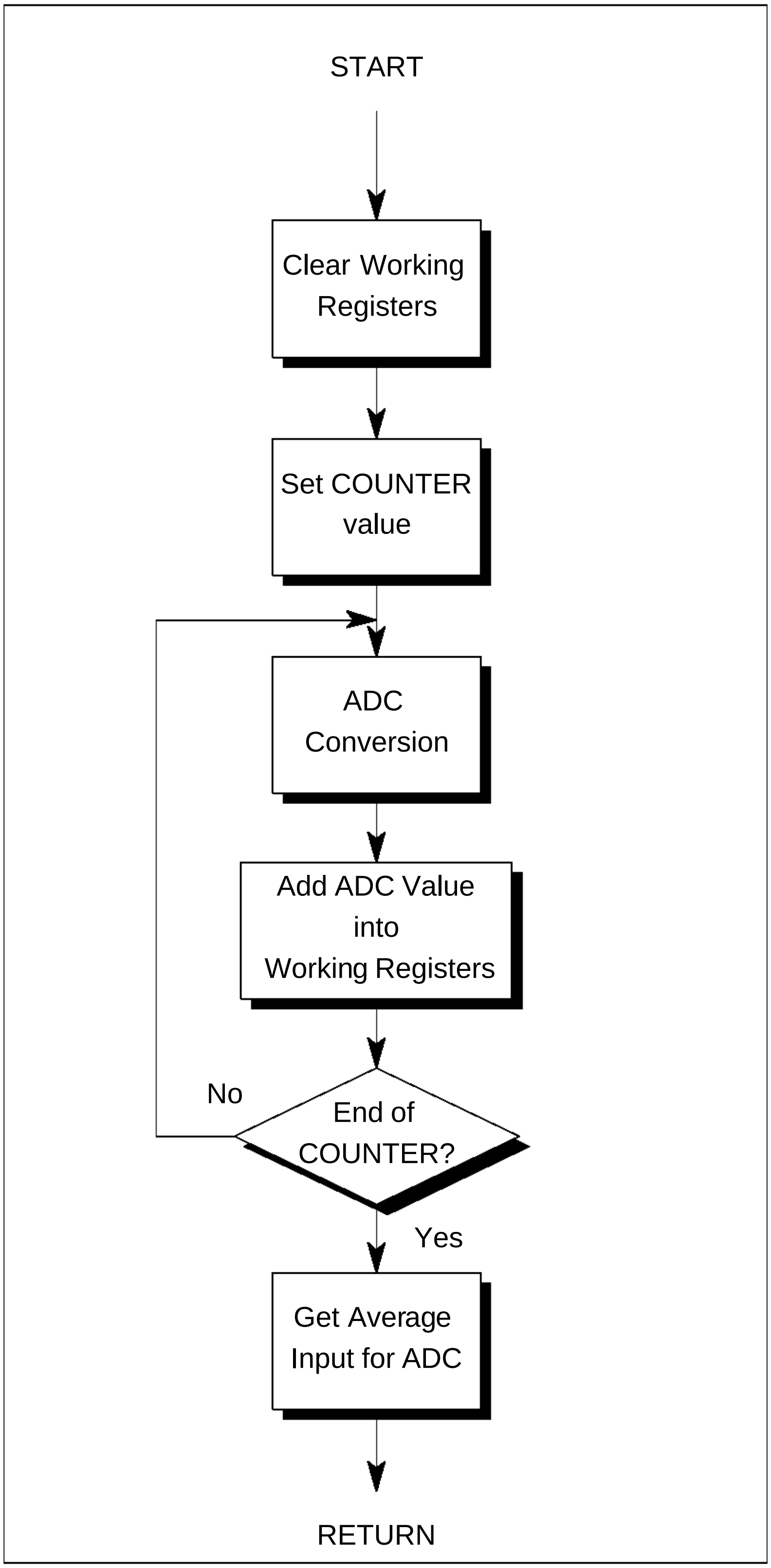


Figure 14. Flowchart for TIMER INTERRUPT routine

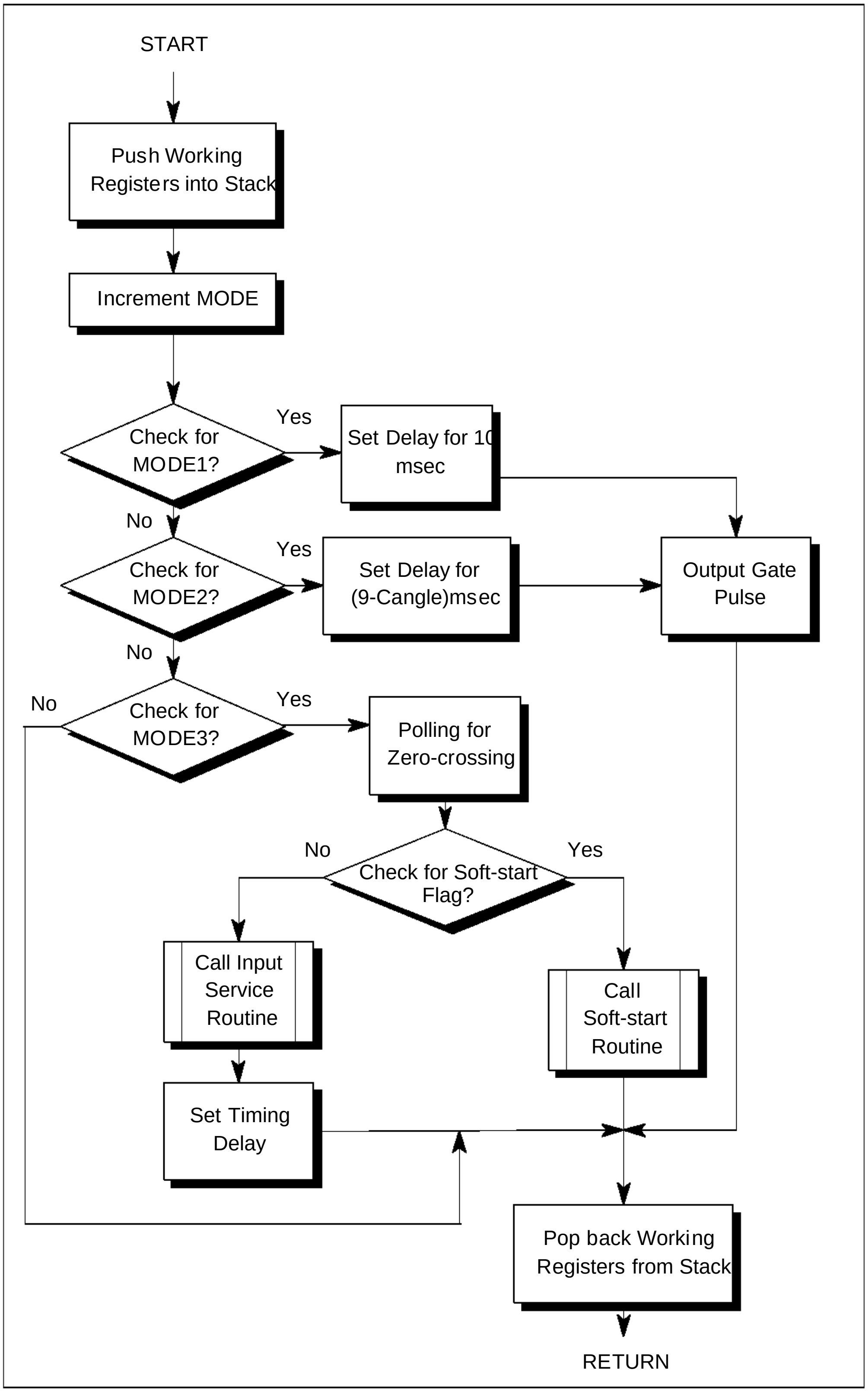
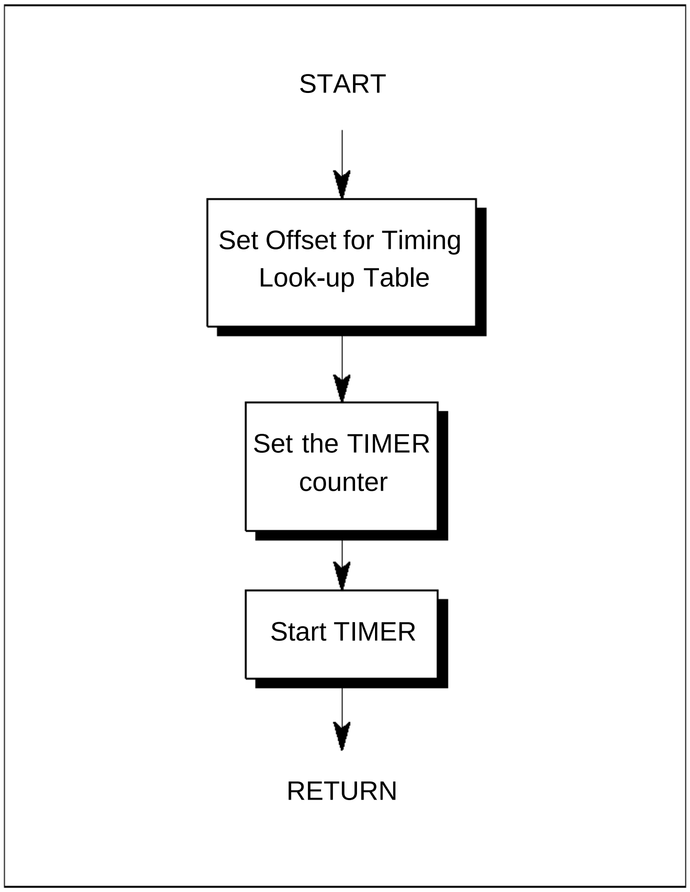


Figure 15. Flowchart for INPUT SERVICE routine

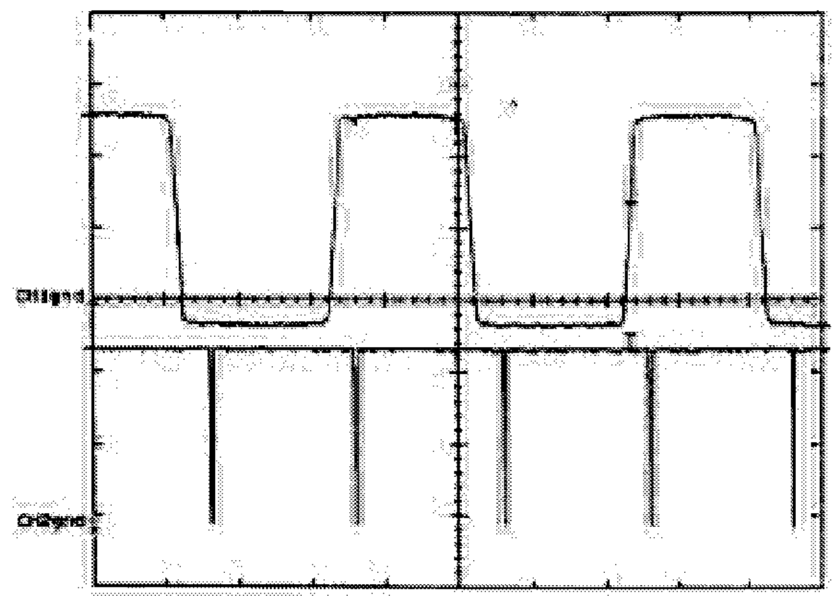


6 EVALUATION RESULTS

Figure 16 includes the two gate pulses generated by the ST6220 for both minimum and maxium output power for the motor. The delay between these gate pulse and their corresponding zero crossing will be the phase delay used to control the output power of the motor. The variations of the motor speed is directly controlled by the changes in these phase delays.

Figure 16. Timing diagram of gate pulse generated by ST6220

A . When the Gate pulse is Maximum



B . When the Gates pulse is Minimum

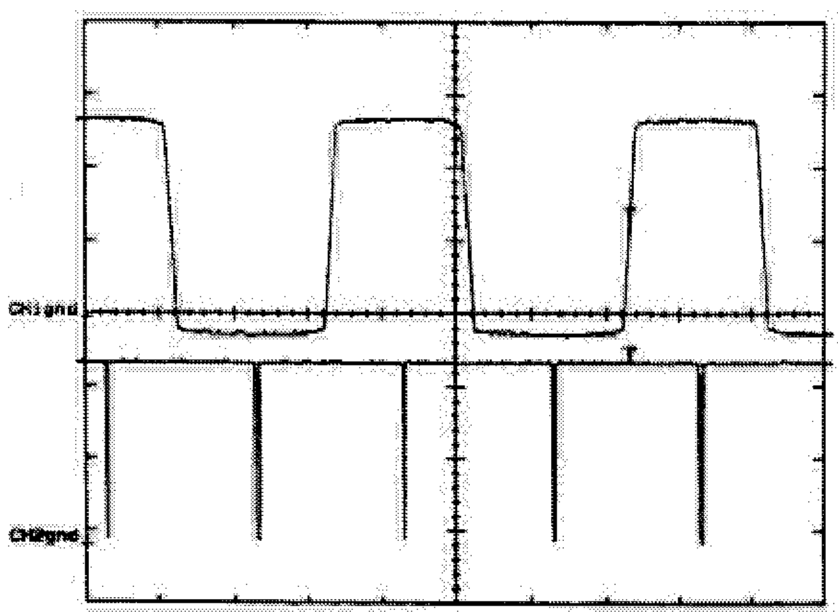
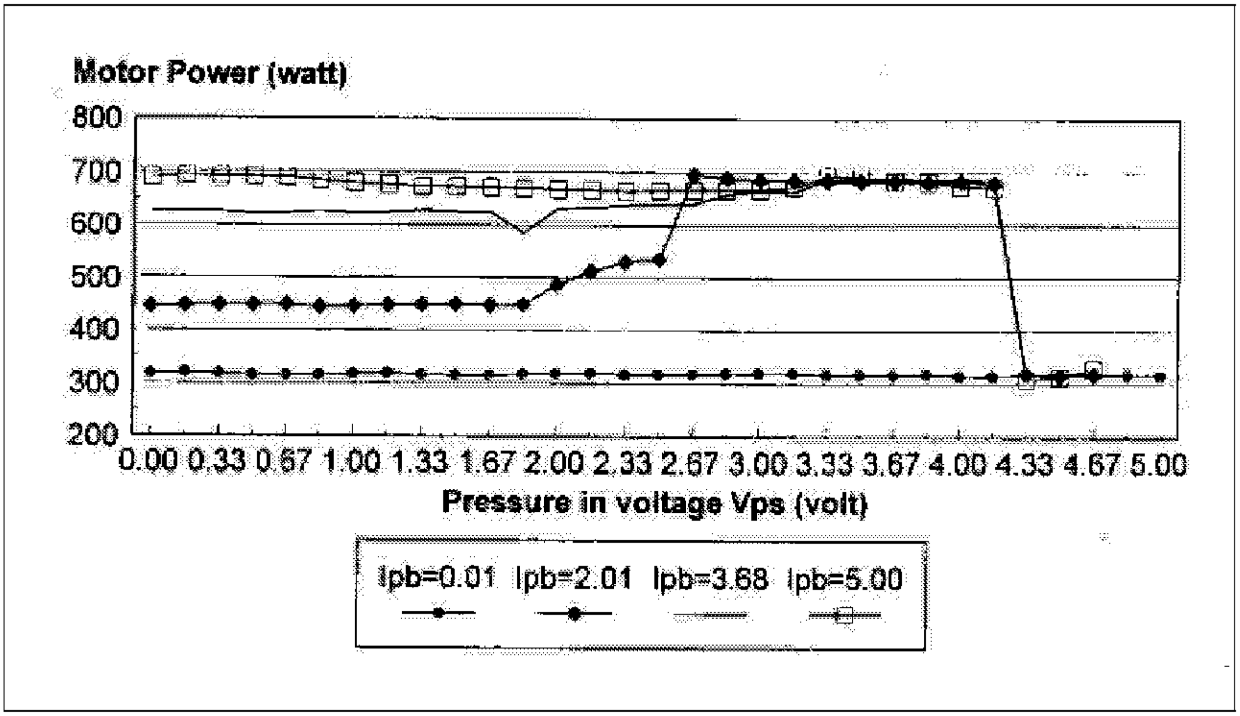


Figure 17 shows the variation of the suction power of the unloaded motor used in the fuzzy vacuum cleaner with respect to the different levels of pressure input while keeping the peak current input at a constant value.

Figure 17. Graph of power versus pressure level with fixed peak current



While the peak current is relatively low (almost at zero level), the suction power is at its minimum value (about 320 watt). This situation is expected to happen only while cleaning is done on a tiled floor.

As the peak current goes higher (refer to the graph with Ipb=2.01 level) and the pressure level still remains in the range of Vps=0V to Vps=2.00V, the suction power of this vacuum cleaner will stay at the high level (about 450 watt). These combinations of pressure and peak current will explain that a loop-pile carpet is being cleaned. If the peak current grows much higher (Ipb=3.68 level) and the pressure levels are still relatively low (in the range of Vps=0V to Vps=2.00V), the power level will be high (about 630 watt) as the cleaning is done on a short cut-pile carpet.

While cleaning on a long cut-pile carpet, the suction power of the vacuum cleaner is required to be very high (about 690 watt). This will occur when the peak current is very high (Ipb=5.00 level) or when the pressure level is relatively high (in the range of Vps=2.6V to Vps=4.2V).

If the vacuum cleaner suddenly sticks to something or sucks in a lump of dust, this will correspond to a very high pressure level (for Vps more than 4.25V) and the vacuum cleaner will automatically decrease its suction power to the minimum value (around 320 watt). After a while, the vacuum cleaner will recover its normal operating condition by itself.

7 CONCLUSION

This note presents a fuzzy logic based system using a vacuum cleaner as an application example. This system is implemented by using a universal motor controlled by a standard ST6220 microcontroller running software using the fuzzy logic concept. The major stages of development of the power control for this vacuum cleaner using the fuzzyTECH ST6 Explorer Edition are also included.

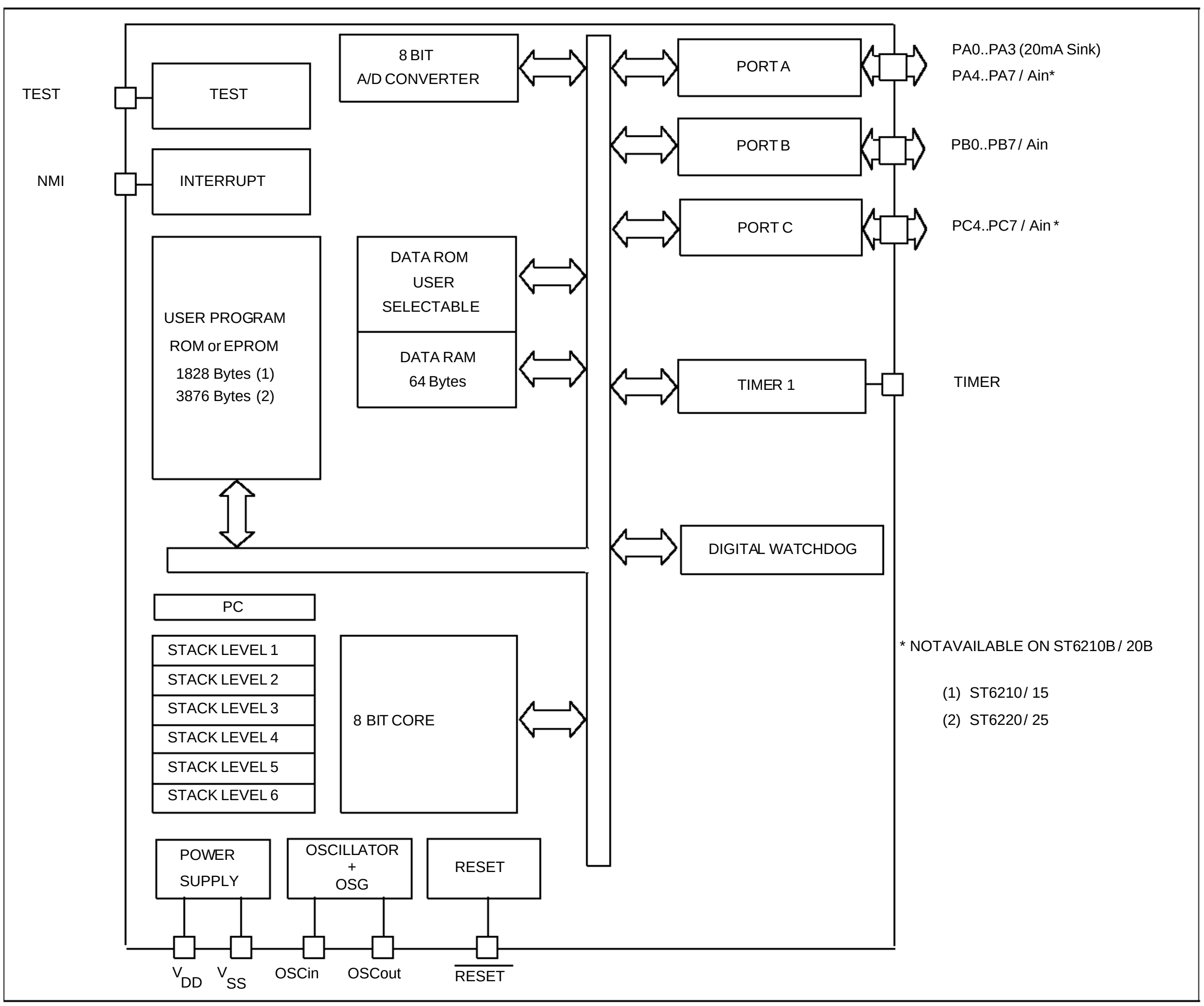
The hardware board implemented is just an emulation board whereby the 2 fuzzy inputs are simulated by using 2 potentiometers. Although the results obtained are effective in regulating the motor power, there exists an opportunity to further maximise this application. An optimised result can be achieved by further fine-tuning the inference rules and membership functions. These optimisation trials may be reduced in number if the effect of the fuzzy logic variable modifications on the system behaviour is well understood.

8 APPENDIX

Summary of Built-in Features

- register oriented 8 bit core
- 8 MHz crystal
- 3876 bytes of ROM and 64 bytes of data RAM
- 12 fully software programmable I/O ports
- 4 I/O lines can sink up to 20 mA for TRIAC driving
- 8 bit A/D converter with up to 8 analog inputs
- digital watchdog
- power-on reset
- on-chip clock oscillator
- PDIP20 and PSO20 package.

Figure 18. ST6220 Block diagram



Pin Description

Pin	Designation	Function	Configuration
***	Vdd	Vdd	Input
***	TIMER	Not Used	Input
***	OSCIN	Oscin	I/O
***	OSCOUT	Oscout	Input
***	NMI	Not Used	Output
***	TEST	Not Used	Input
7	RESET	Reset	Input
***	PB7	Keyboard Input	Input
***	PB6	Pressure Input	ADC Input*
***	PB5	Peak Current Input	ADC Input*
***	PB4	’High’ LED Light	ADC Input*
***	PB3	’Medium’ LED Light	Open Drain Output
***	PB2	’Low’ LED Light	Open Drain Output
***	PB1	Not Used	Open Drain Output
***	PB0	Zero-crossing Input	Input
***	PA3	Not Used	I/O
***	PA2	TRIAC Driver	High Open Drain Output
***	PA1	TRIAC Driver	High Open Drain Output
***	PA0	TRIAC Driver	High Open Drain Output
***	Vss	Vss	Input

* only **ONE** I/O line must be configured as analog input at a time.

NOTES: