Table of Contents

A Quick Look at *fuzzy*TECH®-MP

Fuzzy logic is an innovative technology that enables users to implement intelligent functions in microcontroller devices. With an embedded fuzzy logic control system, application engineers can use low-cost 8-bit microcontrollers to implement intelligent, complicated functions and adaptive control loops with limited resources.

Fuzzy logic is used in applications to:

- Enhance existing products currently in the market place with intelligent functions that provide better adaptation to end-user requirements.
- Utilize sensors with fuzzy logic control strategies that continuously respond to changing input conditions.

The resulting end products using fuzzy logic control strategies embedded in microcontrollers are providing greater end-user satisfaction.

Typical Fuzzy Logic Applications

This Handbook discusses two fuzzy logic applications:

- A software Simulated Crane using a fuzzy logic control strategy that you can manipulate to control the crane as it unloads a ship at a dock.
- A hardware thermostat that uses a fuzzy logic control strategy that you can manipulate to control the temperature on a demo board.

Fuzzy Logic Crane Simulation

The following crane control example illustrates a fuzzy logic application for unloading containers from a ship at dock. Because a ship is costly to keep at dock, the ship must be unloaded as quickly as possible. To reduce the unloading time, the control strategy must actively compensate sway while taking other variables like distance and speed into consideration.

The objective of this example is to transport a container from the ship to the train. The difficulty of this task lies in the fact that the container is hanging from the crane by a long cable, causing the container to sway as the crane moves toward the train. A swaying container cannot be released to the target without risking damage to the target.

This software simulation of a container crane lets you get a quick, hands-on start with fuzzy logic design techniques.

Conventional Control

Because the strategy required to control a crane varies with the position of the crane, a standard PID-type controller cannot be used. A mathematical model of the crane, however, could be used to derive a control algorithm using differential equations. Thus conventional control techniques can be used to build a controller for the crane. However, this approach would involve extensive time-consuming mathematical modeling and the resulting code could require high computational effort

Manual Control

Two basic approaches exist to control the crane manually:

1. Apply low power to the crane motor to minimize sway.
2. Apply full power and reduce power to the crane motor as the crane approaches the train.

Although both of these manual control solutions seem simple, neither are practical real-world solutions because a container ship at dock is costly and must be unloaded as quickly as possible.

Fuzzy Logic Control

A solution to the crane simulation example is easily found using fuzzy logic. With fuzzy logic, you use elements of everyday language to express the desired system behavior. You can formulate fuzzy logic control strategies using linguistic variables, and IF...THEN rules as follows:

- IF the container is Far from the target position, THEN apply full power. There is no present need to compensate sway.
- IF the container is Close to the target position, THEN compensate sway and approach the target point.

Notice that these control rules use the linguistic terms, *Far* and *Close*. To implement such rules, these terms must be defined as part of a fuzzy logic system.

Fuzzy Logic Thermostat Application

The *fuzzy*LAB™ is a simple thermostat that contains a heating resistor, a potentiometer, and a thermistor. A PWM resistor on the *fuzzy*LAB board is the heating element, and a thermistor is the temperature sensing element. A fuzzy logic application controls the temperature on the *fuzzy*LAB demo board.

Manual Control Mode

In manual control mode, you control the PWM resistor temperature by manually setting the duty cycle of the PWM resistor.

Proportional Control Mode

Proportional Control mode is the default control mode on the *fuzzy*LAB demo board. Using proportional input data leads to the simplest fuzzy logic control strategy that you can implement.

In proportional control mode, *fuzzy*LAB calculates a crisp input value, temperature error (T_E), which is the difference between the final temperature (setpoint) and the present temperature.

$$\boxed{T_E = T_{Final} - T_{Present}}$$

The *fuzzy*LAB demo board serially sends the T_E values to the PC based *fuzzy*TECH software. The *fuzzy*TECH software then uses a fuzzy logic control algorithm to calculate an output duty cycle and returns one output, Duty_Cycle. The Duty_Cycle controls the percentage of time *fuzzy*LAB applies power to the PWM heating resistor.

Proportional/Differential Control Mode

In proportional/differential control mode, *fuzzy*LAB determines the values for temperature error, T_E, and the rate of temperature error change dT_E/dt. The *fuzzy*LAB demo board sends the T_E, and dT_E/dt values to the PC based *fuzzy*TECH software. The *fuzzy*TECH software then uses a fuzzy control algorithm and sends back the Duty_Cycle value for the PWM heating resistor.

Fuzzy Logic Software Development Systems

Microchip Technology offers two fuzzy logic software development systems that enable application engineers to implement fuzzy logic control strategies. These software tools are designed to help develop control strategy firmware that you can embed into Microchip microcontroller devices. Even though fuzzy logic control strategies support a number of input and output variables and fuzzy logic rules, chip resources place practical limits on the actual number of input and output variables that you can implement, and on the number of fuzzy logic rules that you can use.

The *fuzzy*TECH-MP Edition (Part Number: DV005002) is a complete fuzzy logic software development system for all PIC16/17 microcontroller devices. Its all-graphical editor and analyzer tools provide efficient system design, optimization and verification. At the push of a button, the *fuzzy*TECH-MP Edition generates the fuzzy logic system designed as highly optimized assembly code, thus circumventing the need for specialized hardware. *fuzzy*TECH-MP Edition provides up to eight input variables, and four output variables per module.

The *fuzzy*TECH-MP Explorer (Part Number: DV005001) software is a limited version of the *fuzzy*TECH-MP Edition. It contains all functionality of the *fuzzy*TECH-MP Edition but only allows users to define fuzzy logic rules with a maximum of two input linguistic variables and one output linguistic variable. Also, each linguistic variable may only contain up to five linguistic terms. All systems designed with the *fuzzy*TECH-MP Explorer can be further developed with the *fuzzy*TECH-MP Edition.

Getting Started with *fuzzy*TECH

This chapter describes the hardware requirements for the *fuzzy*TECH system, and gives the *fuzzy*TECH software specifications.

*fuzzy*TECH System Requirements and Installation

To run *fuzzy*TECH, you need a 386-PC or higher with a minimum of two MB memory and Microsoft Windows® 3.1 or higher running in Enhanced Mode.

For the most efficient work, we recommend a 486-PC with 4 MB memory, and a color screen resolution of 800x600 or 1024x768, 256 colors, and a mouse.

A *fuzzy*TECH installation requires about 6 MB of hard disk space. A 3" disk drive is required for installation from a floppy disk.

*fuzzy*TECH can also be installed on network systems.

To install both the *fuzzy*TECH-MP Explorer and the *fuzzy*TECH-MP Edition you should have at least 8 megabytes of free disk space available on your hard disk. Installing just one system requires about 4 megabytes of free disk space on your hard disk.

Explorer Installation

To run the *fuzzy*TECH-MP Explorer, you need:

A 386 (or higher) PC with the following:

- DOS 5.0 (or higher)
- MS-Windows 3.1 (or higher) Running In Enhanced Mode
- A Color Graphic Monitor with a Resolution of 800 x 600 or 1024 x 768 is recommended, since *fuzzy*TECH makes extensive use of graphics.
- A mouse is not required but recommended since all *fuzzy*TECH functions are accessible with the keyboard.

MP Edition Installation

To run the *fuzzy*TECH-MP Edition, you need:

386 (or higher) PC with the following:

- Parallel port for a hardware key
- DOS 5.0 (or higher)]
- MS-Windows 3.1 (or higher) running in Enhanced Mode.
- Color Graphic Monitor with a Resolution of 800 x 600 or 1024 x 768 is recommended, since *fuzzy*TECH makes extensive use of graphics.
- A mouse is not required but recommended since all *fuzzy*TECH functions are accessible with the keyboard.

You may either install the *fuzzy*TECH software under MS-DOS® or MS-Windows. However, the installation procedure cannot be started in the DOS box of MS-Windows.

*fuzzy*TECH-MP Software Specifications

Operating Environment

Operates under Windows 3.1 on a 386 or higher PC compatible platform.

Input / Output Limits

- *fuzzy*TECH-MP Explorer supports up to two input variables and one output variable.
- *fuzzy*TECH-MP Edition supports up to eight input variables and four output variables per module. Larger systems can be built by using multiple modules. Chip resources will limit the actual number of variables that can be implemented.

Rule Base

- No theoretical limit exists on rules, antecedents and linguistic conjunctions. However, chip resources will place a practical limit on the number of rules.

Fuzzification, Inference, Defuzzification

- Performs input fuzzification by fast computation of a Membership Function (MBF).
- Supports MAX-MIN and MAX-DOT inference methods.
- Supports Fuzzy Associative Map (FAM) rules, to the extent possible within processor resources available.
- Supports Center-of-Maximum (CoM) and Mean-of-Maximum (MoM) user-defined defuzzification.

Data Resolution

- Supports 8-bit resolution on input and output variables.
- Supports 16-bit internal computation resolution during defuzzification.

User Interface and Debug Capabilities

- Supports full graphical user interface.
- Supports various debug modes including Interactive, Link, File and Batch.
- Supports simulation capability with various methods to display results.

Output and Code Generation

- Generates ASCII FTL (Fuzzy Technology Language) output to a file.
- Supports code generation in PIC16C5X assembly language, PIC16CXX assembly language and PIC17CXX assembly language. Also supports assembly language code complied with the Microchip MPASM assembler.
- Assembly code generator handles “Banked” RAM resources (included in some PIC16/17 processors) in an intelligent fashion by incurring “Banking” overhead only when necessary.

Copy Protection

All *fuzzy*TECH-MP versions other than the Explorer use a copy protection device. The matchbox size copy protection inserts into a parallel port (one of the printer ports LPTx) on your PC. If you are using a printer, you may connect the printer cable at the opposite side of the protection device. The copy protection device does not disturb printer operations. You may also use more than one protection device at the same time.

Compatibility

The protection device used with *fuzzy*TECH is carefully designed to not interfere with any other software or hardware products using the parallel printer port. If you are using copy protection devices from other vendors at the same time and you are experiencing problems, try changing the sequence of the devices. If this does not solve the problem, please contact Inform.

Network Installation

Each *fuzzy*TECH Edition can be installed on a network server to be used by different client PCs. Note that each PC needs its own hardware protection device for the *fuzzy*TECH-MP Edition.

Project files and generated C-code can be stored on a local drive of each client PC by changing the path in the File dialog or in the Directories dialog in *fuzzy*TECH.

Fuzzy Logic Primer

This primer guides you through easy-to-visualize fuzzy logic principles. The text focuses on the fuzzy logic features implemented in the *fuzzy*TECH software.

Fuzzy logic is an innovative technology that enables you to describe a desired system behavior using everyday spoken language. Applications range from consumer electronic goods and household appliances to auto electronics, process control, and industrial automation.

Fuzzy logic uses membership functions to define the degree to which crisp physical values belong to terms in a linguistic variable set. For a linguistic variable such as *Power*, typical terms in the linguistic variable set might be *Small, Medium,* or *High*.

The linguistic variable set, *Power*, can be expressed as:
Power (*Small, Medium, High*).

In general, a linguistic variable set can be expressed as:
Linguistic_Variable (Term_1, Term_2,... Term_n)
where n typically equals 3, 5, or 7.

After the *fuzzy*TECH software assigns degrees of membership to the linguistic terms, the software processes the terms by making fuzzy logic inferences.

The *fuzzy*TECH software produces crisp output values through a process called defuzzification.

Fuzzy Logic Definitions

The definitions of fuzzy logic terminology used in this User's Guide are given here for your quick reference.

Base Variable

The base variable is plotted over the horizontal axis of the membership function graph. For an input membership function, the base variable defines the range of crisp physical values that the input linguistic variable represents. For an output membership function, the base variable defines the range of crisp physical values that the output linguistic variable represents.

Crisp Value

A crisp value is a specific input or output physical value that can be measured with instruments. Crisp input values to the *fuzzy*TECH software typically come from sensors or instruments such as the thermistor on the *fuzzy*LAB demo board. Crisp output values are defuzzified values from the *fuzzy*TECH software that provide control.

The *fuzzy*TECH software uses membership functions to determine the degree to which a crisp value belongs to a linguistic variable set.

Degree of Membership, μ

The degree of membership is the degree to which a crisp value belongs to a linguistic variable set, or the degree to which a linguistic variable is true. The Degree of Membership has a value ranging from zero to one.

0 = The variable does not belong to the set — totally untrue
1 = Total membership in the set — totally true

The Greek letter, μ, represents degree of membership.

Degree of Support

Degree of Support (DoS) is the degree to which the *fuzzy*TECH software supports a specific rule in a rule base when calculating an inference from the rule. The degree of support allows you to attach individual weights to each rule in a rule base. Degrees of Support range from 0.00 to 1.00. A rule with a Degree of Support of 1.00 is fully supported. A rule with a Degree of Support of zero is equivalent to a nonexistent rule.

Fuzzification

Fuzzification is a conversion process performed by the *fuzzy*TECH software that converts real world crisp data measurements to fuzzy logic linguistic values. A membership function (a graphical plot of linguistic terms) defines the degree of membership for each specific crisp value. A fuzzified crisp value has a degree of association ranging from zero to one inclusive.

Linguistic Term

A linguistic term is a member of a linguistic variable set. A linguistic variable set typically contains three or more linguistic terms.

Each linguistic term associated with a linguistic variable set has a degree of membership that ranges from zero to one inclusive. A degree of membership of one represents total membership in the linguistic variable set; a degree of membership of zero represents total exclusion from the linguistic variable set.

Partial membership in a linguistic variable set allows for computations which are partly true and partly false. Thus, an infinite number of degrees of membership are possible.

Linguistic Variable

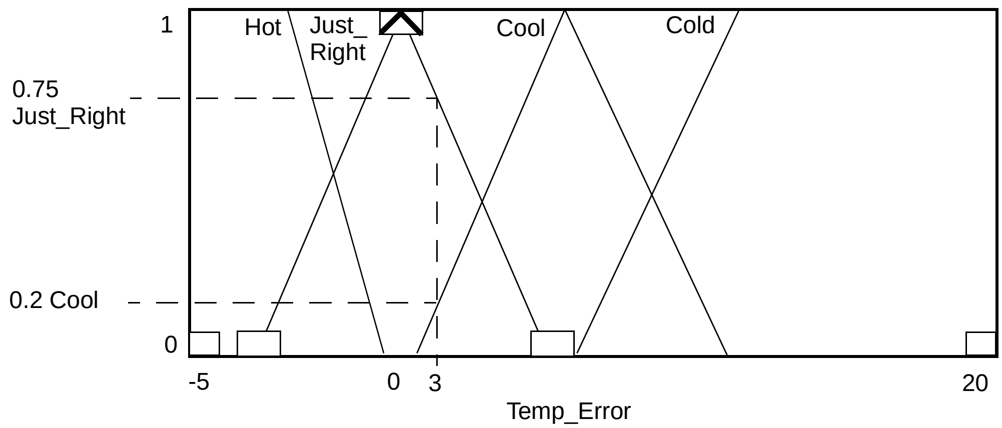
A linguistic variable is a data type (like string, character, real...) that represents a measurable quantity such as temperature, temperature error, rate of change, distance, speed, angle, weight, etc. A linguistic variable can also represent a non-tangible value such as process criticality. A linguistic variable contains associated linguistic terms that represent everyday concepts. For example, a linguistic variable, Temperature, may have the associated linguistic terms Cold, Cool, Just_Right, and Hot.

Membership Function

The membership function (MBF) graphically represents the degree to which crisp values belong to a given fuzzy set. A membership function is a plot of all linguistic terms that belong to a linguistic variable. For example, the membership function for Temp_Error is a plot of the degree to which the linguistic terms Cold, Cool, Just_Right, and Hot belong to the linguistic variable, Temp_Error.

The *fuzzy*TECH software uses the membership function plots to calculate the degree of membership to assigned to a linguistic term.

See Figure 1 below for an example of a membership function for Temp_Error. In this example, the temperature error is 3°C. For a temperature error of 3°C, Just_Right (J.R.) has a degree of membership of 0.75, and Cool has a degree of membership of 0.2.



3°C Fuzzified to 0.75 Just_Right and 0.2 Cool

Figure 1 - Membership Function for TEMP_ERROR

Rule Based Reasoning Versus Classical Logic

The following paragraphs contrast the differences between fuzzy logic and Boolean logic:

- Fuzzy Logic — A natural, continuous logic that expresses degrees of membership.
- Boolean Logic — An exact logic that expresses precise physical values of zero and one.

Fuzzy logic is a natural, continuous logic patterned after the approximate reasoning of human beings. As a theoretical mathematical discipline, fuzzy logic reacts to constantly changing variables. It challenges traditional logic by not being restricted to the conventional binary computer values of zero and one. Instead, fuzzy logic allows for *partial* truths and *multivalued* truths.

Fuzzy logic is especially advantageous for problems that cannot be easily represented by mathematical modeling because data is either unavailable, incomplete, or the process is too complex. The real-world language used in fuzzy control enables engineers to incorporate ambiguous, approximate human logic into computers. Using linguistic modeling, as opposed to mathematical modeling, greatly simplifies the design and modification of a fuzzy logic system. Linguistic modeling also leads to quick development cycles, easy programming, and accurate control.

Boolean Logic

Science, mathematics, and the foundation of computers are based upon the precise principles of Boolean logic. Despite the advantages of its accuracy, classical Boolean logic has a major drawback: it cannot reproduce human thought patterns. This is because the thought patterns of humans are largely intuitive. In both human thought and linguistics, references to vaguely defined concepts are frequent.

Example:	Consider an exact Boolean logic set of *Tall* people, a concept that can be understood without establishing at what height a person is considered *Tall*. In traditional logic, the ideal tall person correlates with an exact height. Without a cut-off limit, it would be impossible to establish whether a premise is true or false. If 6 ft. is established as the threshold for *Tall,* then it would be unreasonable that a person with a height of 6 ft. 1 in. would be considered *Tall* and a person with a height of 5 ft. 11 in. would not be considered *Tall*. In such a case, the human mind would think about the strength or weakness of the relationship between the example data and the established threshold. In fuzzy logic, this is called degree of membership.

In contrast to the exactness of Boolean logic, fuzzy logic recognizes the advantages of approximate logic. In most real-world situations, a precise answer does not necessarily provide the optimal solution.

Designing a Fuzzy Logic Rule Base

In control applications, fuzzy logic uses everyday spoken language to define a rule base. To define a fuzzy logic control strategy such as a strategy to control a crane motor, the goal of the control strategy should be to obtain a desired output. With the crane motor, output power applied to the crane motor is based on the input values for the crane position and load angle.

A crane moving a container from point A to point B as quickly as possible without damage provides an excellent example for describing how engineers are using fuzzy logic. This example sounds simple but it has many variables. The figure below shows
a rule base for using a crane to move a container from a starting point to a destination. If a container is FAR from the destination, THEN apply Positive_High power etc.

Fuzzy logic systems use the following conditions to define and implement a rule base:

- Input and output linguistic variables contain terms that may be true to a degree ranging from zero to one.

- Zero represents totally untrue, and one represents totally true. A linguistic term that is true to the degree 0.5 would be half true and half false.
- A fuzzification step links crisp input values to the fuzzy logic control strategy.
- The linguistic operators AND, and OR express conditions in the rule base.
- IF...THEN operators express conditions in the fuzzy logic rule base. The *fuzzy*TECH software uses these IF... THEN conditions to make fuzzy logic inferences that link input linguistic terms to output linguistic terms. See Figure 2
- A defuzzification step links output linguistic terms to crisp output values.

Typical Rule Base				
Rule		**Input Linguistic Terms for Distance**		**Output Linguistic Terms for Power**
1	IF	Far	THEN	Positive_High
2	IF	Medium	THEN	Positive_Medium
3	IF	Close	THEN	Positive_Medium
4	IF	Zero	THEN	Zero

Figure 2 - Typical Fuzzy Logic Rule Base

Implementing a fuzzy control strategy for controlling a crane, requires two-way translations between crisp values and linguistic terms. The *fuzzy*TECH software uses the following three steps to implement a fuzzy logic control strategy:

1. Fuzzification

The *fuzzy*TECH software translates crisp input values into linguistic terms and assigns a degree of membership to each crisp input value. Linguistic variables are fuzzy sets which contain linguistic terms as members of the set.

Example: For a linguistic term named Far, the fuzzyTECH software calculates the degree to which 60 yards is considered Far and assigns a degree of membership to the crisp input value of 60 yards.

2. Fuzzy Rule Inference

The fuzzyTECH software uses IF...THEN rules to define the relationship between linguistic variables. These rules determine the course of action the controller must follow.

Example: If Distance is Far THEN Power is Positive_High.

3. **Defuzzification**

   The fuzzyTECH software translates the result of the fuzzy inference from a linguistic concept to a crisp output value.

   Example: "Power" Positive_High translates to a physical value of 8 amps.

Linguistic Variables

In fuzzy logic, the different values for a given linguistic variable represent concepts, not numbers.

Example:	Consider a linguistic variable, Distance, representing the distance from a target point of a container carried by a crane. The linguistic variable, Distance, has associated linguistic values, called linguistic terms that describe Distance. Possible associated linguistic terms describing Distance could be Far, Medium, Close, Zero, or Neg_Close. Since each linguistic variable is represented by a specific fuzzy set, even approximate linguistic terms like Extremely_Close and Very_Far are possible.

Using Membership Functions

Membership functions (MBF) represent the degree to which crisp values belong to a given fuzzy set. Typically, the horizontal axis (base variable) of the membership function represents crisp physical quantities, or the universe of discourse. The vertical axis represents degrees of membership. The degree of membership is represented by m and has a value ranging from zero to one inclusive. See Figure 3.

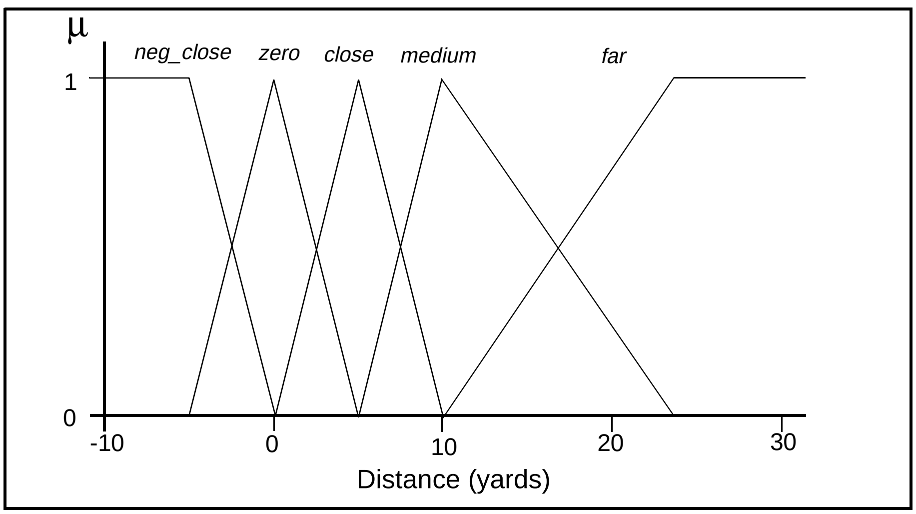


Figure 3 - Membership Function for Distance

Standard Membership Functions

Although scientific publications have suggested many different types of membership functions for fuzzy logic, fuzzyTECH uses standard membership functions in most practical applications. Figure 4 shows the standard membership function types:

- Z-Type (Also called Trapezoidal)
- Lambda Type (Also called Triangular)
- Pi-Type
- S-Type

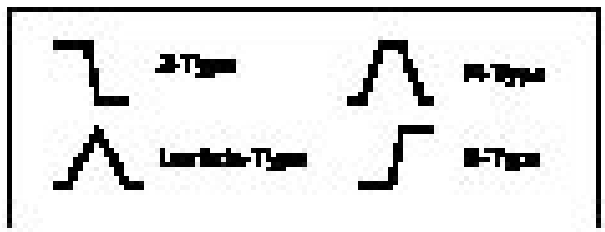


Figure 4 - Standard Membership Function Types

Defining a Membership Function

Each of these membership functions can be mathematically represented as a piecewise linear function with up to four defining points. When you define a membership function, place the degree of membership on the vertical axis, and place the (crisp) value (representing physical data) on the horizontal axis. Use the following steps to guide you as you define membership functions for fuzzyTECH software:

Steps for Defining a Membership Function

Step 1: For each term of a given linguistic variable, find the crisp numerical value or range of crisp values that best characterizes the term. Since this is a “prototype” value, assign a degree of membership of one in the fuzzy set describing the term.

Step 2: After you define the crisp values with a degree of membership of one, find the crisp numerical value or range of crisp values with a degree of membership of zero. (These crisp values are found as the “prototype” of the next higher term and the “prototype” of the next lower term determined in Step 1.)

Step 3: After all crisp values with a degree of membership, m, of either zero or one are defined, define the intermediate crisp values. With standard membership functions, use lines to connect the crisp values. Except for the left and right-most linguistic terms, this results in standard Pi- and Lambda-type membership functions.

Step 4: Any crisp value that is less than the lowest-defined value for the left most term or any crisp value that is greater than the highest-defined value for the right most term is considered an absolute member of the respective term's fuzzy set (μ=1). These are represented by Z- and S-type membership functions

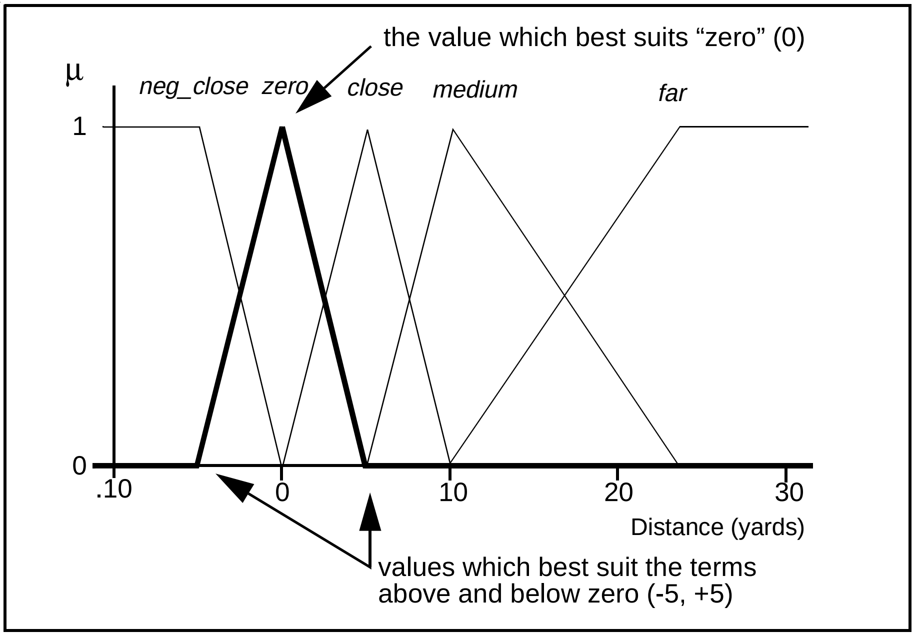


Figure 5 - Defining a Standard MBF for the Term Close

In some applications, the value that best characterizes a term will be an interval, not a number.

Example:	Any distance within ±2 yards from the target point could be considered zero and would be best represented as a Pi-shaped MBF. Figure 6 shows an example of a Pi type MBF

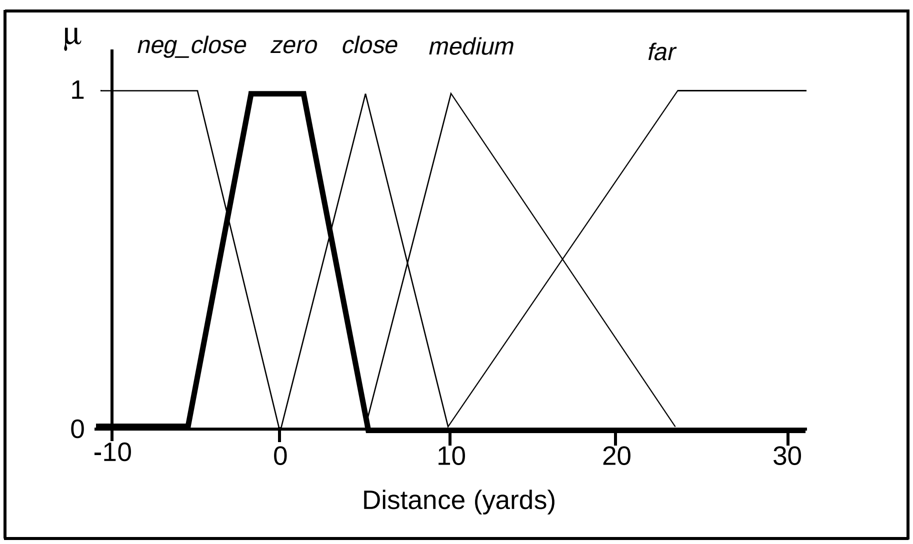


Figure 6 - Definition of a Pi-Type MBF for the Term Close

Fuzzification

Fuzzification refers to the task of assigning degrees of membership to crisp input values. Crisp input values are often measured with instruments such as a volt meter, or a thermistor. Crisp input values could also originate from data calculated by a software application, or crisp input values could be entered by an operator.

Fuzzification means to use membership functions to compute the degree of validity or degree of membership of each linguistic term at a specific operation point of the process.

Example:	Consider the linguistic variable “Distance” as shown in Figure 5. Let the distance measured by the sensor be 12 yards. The result of fuzzification would be:
Far	degree of validity = 0.9
Medium	degree of validity = 0.1
Close	degree of validity = 0.0
Zero	degree of validity = 0.0
Neg_Close	degree of validity = 0.0

Linguistically, a distance of 12 yards could be expressed as *Nearly_Far_Just_Slightly_Medium*. Fuzzification is the first step in the computation of a fuzzy system and must be performed for each input variable. Figure 7 shows the membership function definition for *Angle*.

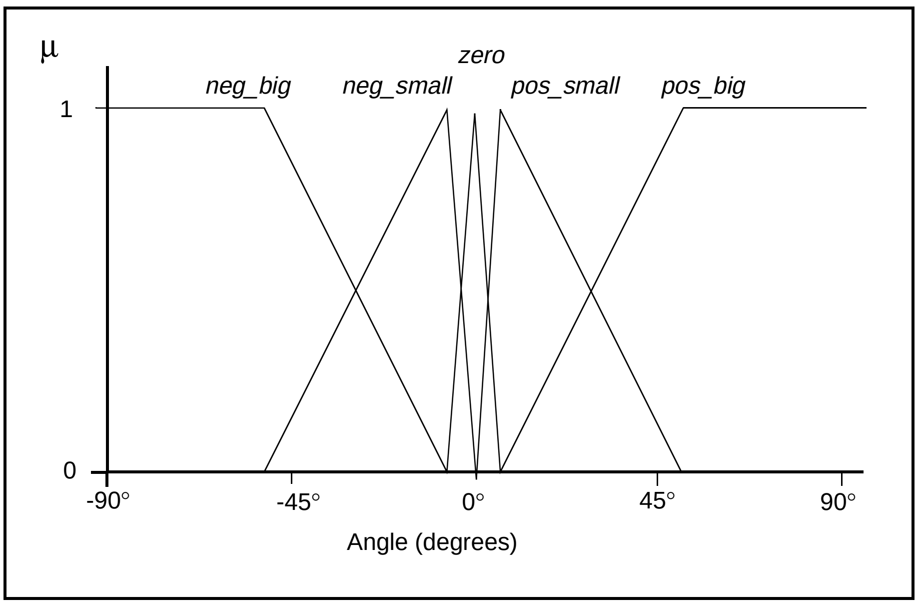


Figure 7 - Membership Functions for Angle

Example:	In Figure 7, if Angle = 4°, fuzzification would calculate:
Neg_Big	degree of validity = 0.0
Neg_Small	degree of validity = 0.0
Zero	degree of validity = 0.2
Pos_Small	degree of validity = 0.8
Pos_Big	degree of validity = 0.0

Once the membership functions of all terms have been defined, the control behavior can be defined in the system rules.

Fuzzy Rule Inference

The computation of fuzzy rules is called fuzzy rule inference. The *fuzzy*TECH software uses production rules when calculating inferences. The *fuzzy*TECH software calculates inferences in two steps: aggregation and composition.

Production Rules

Fuzzy logic based application solutions use production rules to represent the relationship between the linguistic variables and to derive output actions from sensor inputs. Production rules consist of a precondition (IF-part) and a consequence (THEN-part). The IF-part may consist of more than one condition linked together by the linguistic conjunctions AND or OR.

Aggregation–First Step of Fuzzy Rule Inference

Aggregation the first step of the fuzzy inference, determines the degree to which the complete IF-part of the rule is fulfilled. Special fuzzy operators are used to aggregate the degrees of validity of the various preconditions.

AND, OR Fuzzy Operators
Aggregation calculations use fuzzy logic operators to calculate the result of the IF part of a production rule when the rule consists of more than one input condition. The linguistic conjunctions, AND or OR, links multiple input conditions. Using the conjunction AND for the minimum and OR for the maximum is often appropriate in small control applications.

Example: Consider the following rules:

Rule 1: IF Distance = Medium AND Angle = Pos_Small THEN Power = Pos_Medium

Rule 2: IF Distance = Far AND Angle = Zero THEN Power = Pos_Medium

Rule 3: IF Distance = Medium AND Angle = Zero THEN Power = Zero

In these fuzzy logic production rules, an AND links the two input conditions.

MIN, MAX Fuzzy Operators

Sometimes other kinds of operators are needed to signify the relationship of the different parts of the condition. All *fuzzy*TECH Editions support MIN and MAX operators. (Some *fuzzy*TECH editions also support compensatory operator families such as GAMMA or MIN-AVG.)

Aggregation Operators	MIN	Minimum operator. Selects the minimum member of a set.
	MAX	Maximum operator. Selects the maximum member of a set.

In Rule 1 in the previous example, the minimum operator, MIN, represents the conjunction AND. The validity of the entire condition is calculated as:

In Rule 1 in the previous example, the minimum operator, MIN, represents the conjunction AND. The validity of the entire condition is calculated as:

Validity of the Condition =

MIN {Degree of Validity (Distance = Medium),
Degree of Validity (Angle = Pos_Small)}

Consider a crane in which the distance of the load from the target point is 12 yards and the load angle is 4 degrees.

Fuzzification of this data would establish for rule 1 that the first condition, Distance = *Medium*, is fulfilled to the degree of 0.9 and the second condition, Angle = *Pos_Small* is fulfilled to the degree of 0.8.

Therefore, the Validity of the Condition = MIN {0.9, 0.8} = 0.8

In words:
The MIN operator selects the minimum member of the set,
MIN {0.9, 0.8}, where
0.9 is the degree of validity of Distance = Medium, AND
0.8 is the degree of validity of Angle = Pos_Small.

Composition–Second Step of Fuzzy Rule Inference

Composition links the validity of the entire condition with the Degree of Support (DoS). Thus, composition, the second calculation step of each production rule, uses the validity of the condition to determine the validity of the consequence.

In standard MAX-MIN or MAX-PROD (sometimes called MAX-DOT) inference methods, the consequence of a rule is considered equally as true as the condition.

Fuzzy Associative Map (FAM) Inference

A Fuzzy Associative Map is a matrix that represents fuzzy rules. White fields represent completely plausible rules (DoS = 1); black fields represent completely implausible rules (DoS = 0); gray fields represent partially plausible rules.

Each rule in a FAM has an assigned Degree of Support that represents the importance of the rule. The Degree of Support represents the degree to which a given rule is plausible. Thus, rules themselves can be “fuzzy” – meaning they can have a Degree of Support ranging between zero and one.

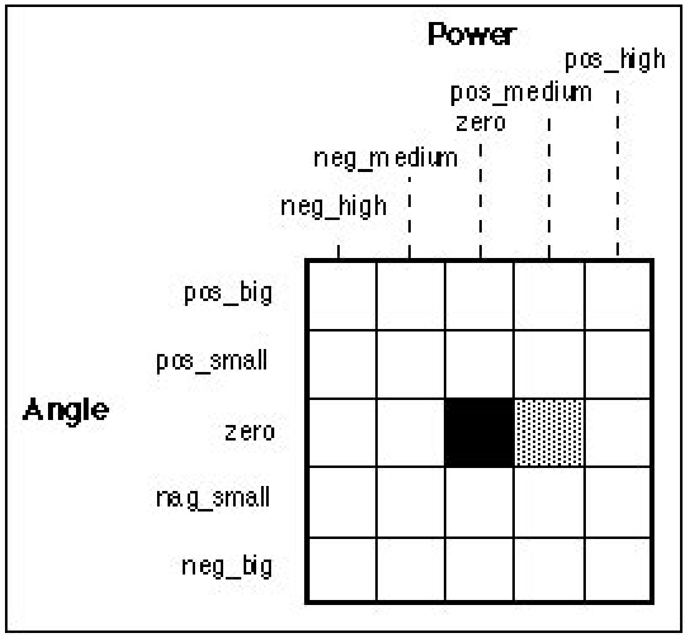


Figure 8 - Expressing Ambiguities using FAM

Standard MAX-MIN or MAX-PROD inference methods produce rules where the individual importance of each rule is either zero or one. Thus, MAX-MIN or MAX-PROD inference methods often result in arbitrary rule addition/deletion, with a clumsy a trial and error approach to optimizing the rules. For this reason, most *fuzzy*TECH Editions support an advanced inference method, the Fuzzy Associative Map inference, or FAM. With FAM inference, each rule is assigned a Degree of Support (DoS) representing the individual importance of the rule.

Fuzzy Associative Map Rules

If a unique consequence for a given combination of input variables cannot be found, use FAM rules to express ambiguities.

Example:	Consider the following rule to be represented in a fuzzy logic system:
	IF Distance = Close AND Angle = Zero
	THEN Power = Mostly_Zero_but_Somewhat_Pos_Medium

Define a new term for Mostly_Zero_but_Somewhat_Pos_Medium as one approach to expressing ambiguities. This approach, however, would result in an excessive amount of terms and membership functions. Moreover, system structure would become unnecessarily complex and difficult to survey.

Using FAM, this ambiguity can be expressed in rules. Figure 8 shows a possible representation of the preceding rule.

Composition Product Operator, PROD

The *fuzzy*TECH software uses the composition product operator, PROD, to calculate the Result of the Rule. In calculating the Result of the Rule, the composition product operator links the Degree of Validity of the Input Condition to the Degree of Support of the Rule.

Composition Operator PROD		
Composition uses the fuzzy logic product operator, PROD, to link the input condition to the output condition.		
Composition Operator	PROD	Product operator. Multiplies the members of a set.

Example: Given the previous example rule 1 (see above), let the Degree of Support be 0.8. Using the product operator for composition, the result of the rule would be:

Result of the Rule (Validity of the Consequence) =

PROD {Degree of Validity of the Input Condition,

Degree of Support of the Rule}

Therefore, the Result of the Rule = PROD {0.8, 0.8} = 0.64

This means that the result for Power *Pos_Medium* has a degree of validity of 0.64.

If more than one rule produces the same consequence (e.g. example rule 2 also produces Power = *Pos_Medium*), the maximum degree of validity of the consequences is selected for all further processing.

Example: Example rules 1 and 2 yield the same result for Power but with different degrees of validity:

Pos_MediumDegree of Validity = 0.64 (Rule 1)

Pos_MediumDegree of Validity = 0.1 (Rule 2)

The degree of validity 0.64 would be adopted for all subsequent computations.

Using Rule Base to Define System Behavior

The individual rules of a fuzzy system define the actual behavior of the system. To produce a prototype of an appropriate set of rules for a fuzzy system:

1. First create rules which represent unambiguous controller strategies at specific operation points.
2. From there, construct a rule set by working backwards, step-by-step, until you have defined the most detailed rule.

Example:	"If the crane has reached the target point and the load is not oscillating, no power should be applied to the motor." This statement can be formulated as a fuzzy logic production rule as follows: IF Distance = Zero AND Angle = Zero THEN Power = Zero.

Once you establish an unequivocal rule set, you can easily define more detailed crane behavior. For example, you could define the following rules if the load is oscillating slightly at the target point:

Example:	If Distance = Zero AND Angle = Neg_Small THEN "Power" = Pos_Med If Distance = Zero AND Angle = Pos_Small THEN Power = Neg_Med

The following steps describe how to use a FAM (refer to Figure 8) to establish the rules of a fuzzy system. Step 3 describes when to use FAM to express ambiguities:

Step 1: Select combination of one output variable and one input variable as follows from the Fuzzy Associative Map:

a. Select the first output variable on the upper horizontal axis of the FAM rule matrix, and

b. Select the input variable with the most influence on the system on the left vertical axis of the FAM rule matrix.

Step 2: For each remaining output variable on the horizontal axis of the Fuzzy Associative Map, select the input variable on the left axis that best suits the output variable.

Define only those rules with a degree of truth of zero or one.

Step 3: For some combinations of terms for input variables, there is no one exact term which expresses the desired output value. If this is the case, do not change the membership function definitions. Instead, use FAM to express the ambiguities.

Initial Rule Base Design

When first getting started with fuzzy logic, use rules with a Degree of Support of only zero and one. If you need to weight individual rules during optimization, use degrees of support between zero and one.

At the end of the total inference process, all system output variables are associated with a fuzzy value. To demonstrate this, the following degrees of validity of the input condition (Angle) are assigned to the Power:

Example:	The linguistic result for Power:	
	Neg_Big	degree of validity = 0.0
	Neg_Medium	degree of validity = 0.0
	Zero	degree of validity = 0.2
	Pos_Medium	degree of validity = 0.64
	Pos_Big	degree of validity = 0.0

Defuzzification

The result produced from the evaluation of fuzzy rules is, of course, fuzzy. In the previous example, the result could be linguistically expressed as Mostly_*Pos_Small_but_Also_Slightly_Zero*. Obviously, a crane motor cannot interpret such a linguistic command.

Output membership functions translate fuzzy output linguistic terms into crisp output values. This translation of output linguistic terms into crisp output values is known as defuzzification. Defuzzification can be performed by using a number of methods. The *fuzzy*TECH software supports the CoM and the MoM defuzzification methods.

CoM Defuzzification

Center-of-Maximum evaluates more than one output term as valid, and compromises between the different results. The Center-of-Maximum (CoM) defuzzification method computes a crisp output value as a weighted mean of the term membership maxima, weighted by the inference results.

Figure 9 defines the membership function for the linguistic variable, Power, and shows a CoM defuzzification snap shot for the linguistic variable, Power. The locations of the individual term membership maxima are indicated by the gray arrows and the inference result is shown by the height of the blackened portion of the arrows. The defuzzification process produces the following crisp output value: $\frac{(0.2 * 0\ \text{KW} + (0.64 * 8\ \text{KW}}{0.2 + 0.64} = 6.1\ \text{KW}.$

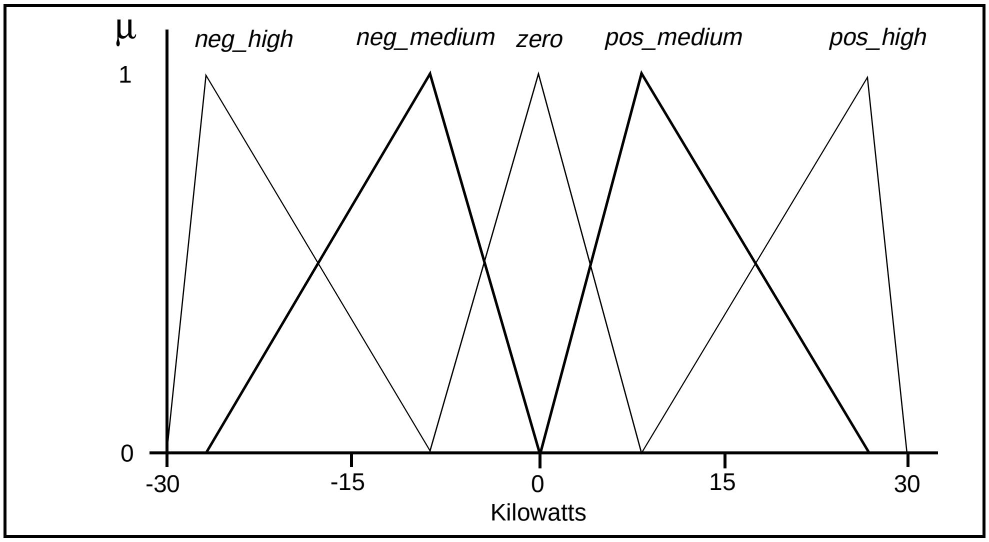


Figure 9 - Definition of the Linguistic Variable Power

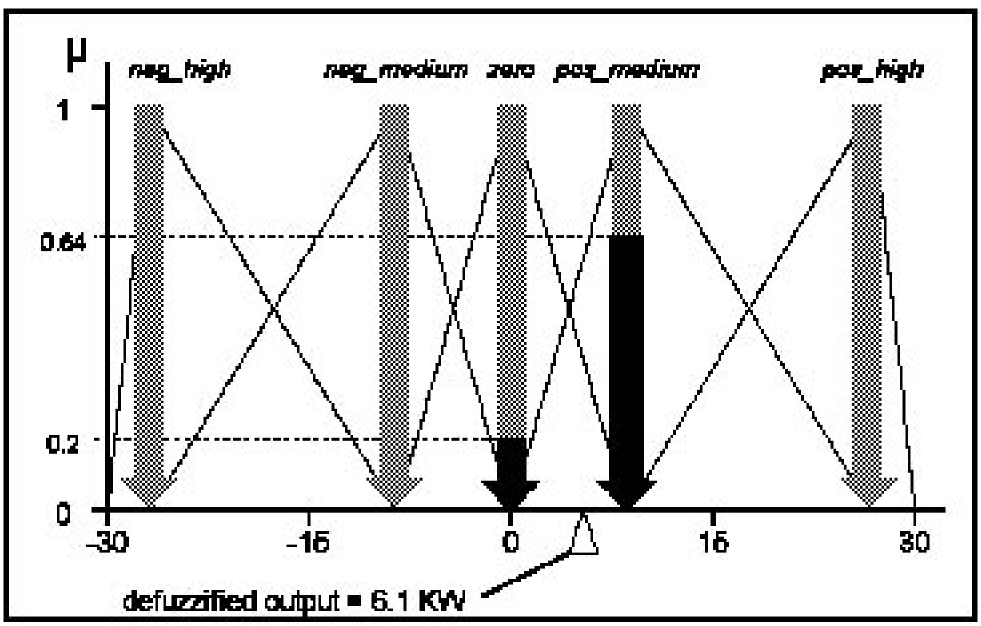


Figure 10 - Defuzzification with Center-of-Maximum

MoM Defuzzification

The Mean-of-Maximum(MoM) defuzzification method computes a system output only for the term with the highest resulting degree of validity. If the maximum is not unique (like in a Pi-shaped MBF), the mean of the maximizing interval is computed. Figure 11 shows the MoM defuzzification process.

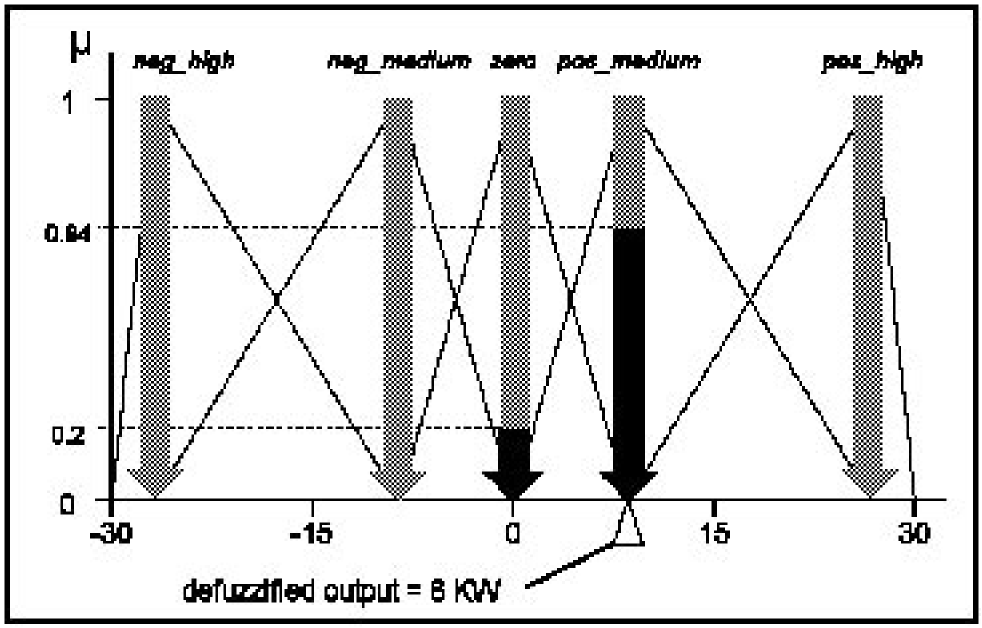


Figure 11 - Mean-of-Maximum Defuzzification

Comparison of Defuzzification Methods

While the CoM defuzzification method results in the *best compromise solution*, MoM results in the *most plausible solution.*

CoM is most commonly used in control applications because the output value represents the best compromise of all inferred results.

MoM is often used in pattern recognition and classification applications, and as a plausible solution MoM is most appropriate.

Scientific literature has suggested many other defuzzification strategies that, until now, have rarely been used in industrial applications. Since *fuzzy*TECH also allows for the omission of defuzzification, a user-defined defuzzification method is also possible.

Controlling *fuzzy*LAB Temperatures with Fuzzy Logic

This tutorial shows you how to use the *fuzzy*TECH software to implement a fuzzy logic control strategy to control the temperature loop on the *fuzzy*LAB™ demo board.

The *fuzzy*LAB is a great training tool to guide you in learning and applying fuzzy logic principles. The *fuzzy*LAB demo board demonstrates the ease with which you can control temperature by using a fuzzy logic control scheme.

Description of the *fuzzy*LAB Demo Board

The *fuzzy*LAB is a simple thermostat that contains a heater resistor, a potentiometer, and a thermistor. The resistor on the *fuzzy*LAB board is the temperature controlling element, and the thermistor is the temperature sensing element.

The resistor, activated by a pulse width modulated (PWM) source, heats an adjacent thermistor. A potentiometer sets the desired final temperature (setpoint), and the thermistor senses the temperature.

***fuzzy*LAB** controls the temperature of an on-board resistor in manual mode, proportional mode, or proportional/differential mode. (These modes are also described in the chapter, “A Quick Look at *fuzzy*TECH-MP.”

- **Manual Control Mode**

  In manual control mode, you control the resistor temperature by manually setting the duty cycle of the PWM resistor.

- **Proportional Control Mode**

  In proportional control mode, *fuzzy*LAB determines the temperature error, T_E, which is the difference between the desired temperature (setpoint) and the current temperature. *fuzzy*LAB serially sends the T_E values to the PC based *fuzzy*TECH software. The *fuzzy*TECH software then uses a fuzzy control algorithm and sends back the duty cycle value for the PWM resistor.

- **Proportional/Differential Control Mode**

  In proportional/differential control mode, *fuzzy*LAB determines the values for temperature error, T_E, and the rate of temperature error change dT_E/dt. *fuzzy*LAB sends the T_E, and dT_E/dt values to the PC based *fuzzy*TECH software. The *fuzzy*TECH software then uses a fuzzy control algorithm and sends back the duty cycle value for the PWM resistor.

Designing a Fuzzy Logic Temperature Control Scheme

The firmware developed by using *fuzzy*TECH software provides fuzzy logic control of the *fuzzy*LAB demo board. When you use fuzzy logic firmware to control the *fuzzy*LAB demo board, you will get the typical benefits that fuzzy logic control provides:

- Fuzzy logic simplifies the design of a control strategy by providing an easy to understand and intuitive approach to solving control problems.
- Fuzzy logic reduces development time, and allows designers to produce control prototypes in less time than would be possible by using conventional control methods.
- Fuzzy logic allows control strategies to use sensors with less precision in many cases while maintaining the desired level of control.

Fuzzification of Crisp Temperature Values

In the fuzzification process, the *fuzzy*TECH software converts crisp input temperature error values, T_E, to fuzzified linguistic terms based on the membership function. The membership function associates a degree of membership ranging from zero to one, inclusive, with respect to temperature. In the figure below, the membership function shows the linguistic terms Hot, Just_Right, Cool, and Cold.

Hot has a degree of membership of zero at $T_E = -2$. Hot has a degree of membership of one at $T_E = -20$ to $T_E = -6$.

Just_Right has a degree of membership of zero at $T_E = -4$ and at $T_E = +4$. Just_Right has a degree of membership of one at $T_E = 0$.

Cool has a degree of membership of zero at $T_E = 2$ and at $T_E = 12$. Cool has a degree of membership of one at $T_E = 8$.

Cold has a degree of membership of zero at $T_E = 10$. Cold has a degree of membership of one at $T_E = 14$.

Figure 12 shows the degree of membership for the linguistic terms associated with $T_E = 3°$ C. To identify the linguistic terms associated with $T_E = 3°$ C, extend a vertical line from 3° C. The vertical line intersects Just_Right at a degree of membership of 0.3, and intersects Cool at a degree of membership of 0.125. Thus, $T_E = 3°$ C corresponds to Just_Right to the degree of 0.3 and corresponds to Cool to the degree of 0.125.

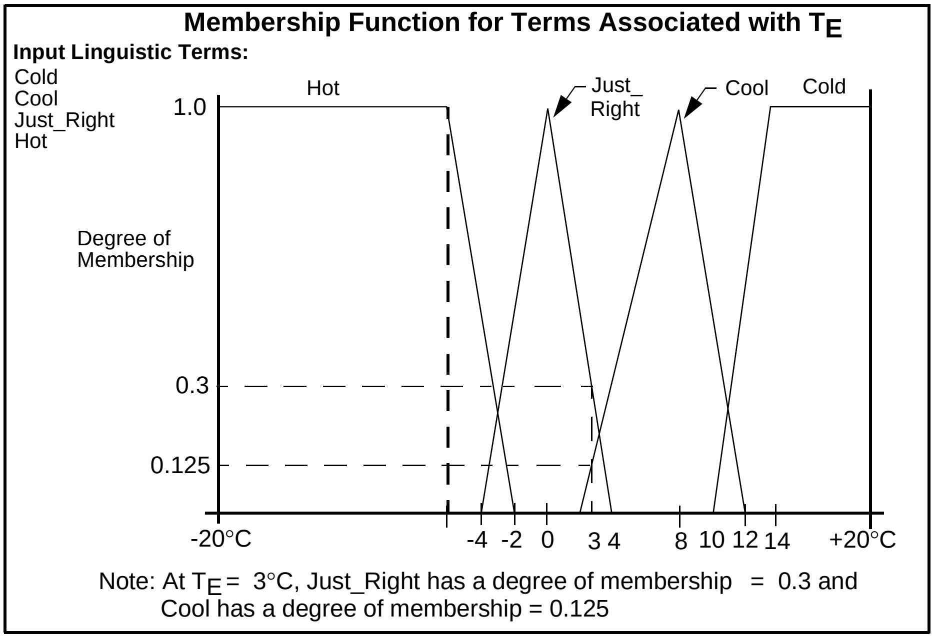


Figure 12 - Fuzzification Membership Function

Fuzzy Rule Inference for Temperature Control

Figure 13 shows the fuzzy logic rule base for our example. The rule base uses fuzzy rule inferences to represent the relationship between the linguistic terms. Inferences link the precondition (IF-part) of the rule base to the consequence (THEN-part) of the rule base. (The IF-part could also consist of more than one condition linked together by the linguistic conjunction, AND, or OR.

This example rule base links input linguistic terms to output linguistic terms in a simple one to one correlation between input linguistic terms and output linguistic terms. The input fuzzy variable T_E is linked to the output fuzzy variable Duty_Cycle. In our example, a simple one-to-one correlation between input linguistic terms and output linguistic terms has been selected as:

Rule One: If the linguistic input term is Cold, THEN the Duty_Cycle is Large.

Rule Two: If the linguistic input term is Cool, THEN the Duty_Cycle is Medium.

Rule Three: If the linguistic input term is Just_Right, THEN the Duty_Cycle is Small.

Rule Four: If the linguistic input term is Hot, THEN the Duty_Cycle is Zero

Rule Base				
Rule		**T_E**		**Duty_Cycle**
1	IF	Cold	THEN	Large
2	IF	Cool	THEN	Medium
3	IF	Just_Right	THEN	Small
4	IF	Hot	THEN	Zero
To simplify our demo, we selected a simple one-to-one rule base where each input linguistic term corresponds to one output linguistic term.				

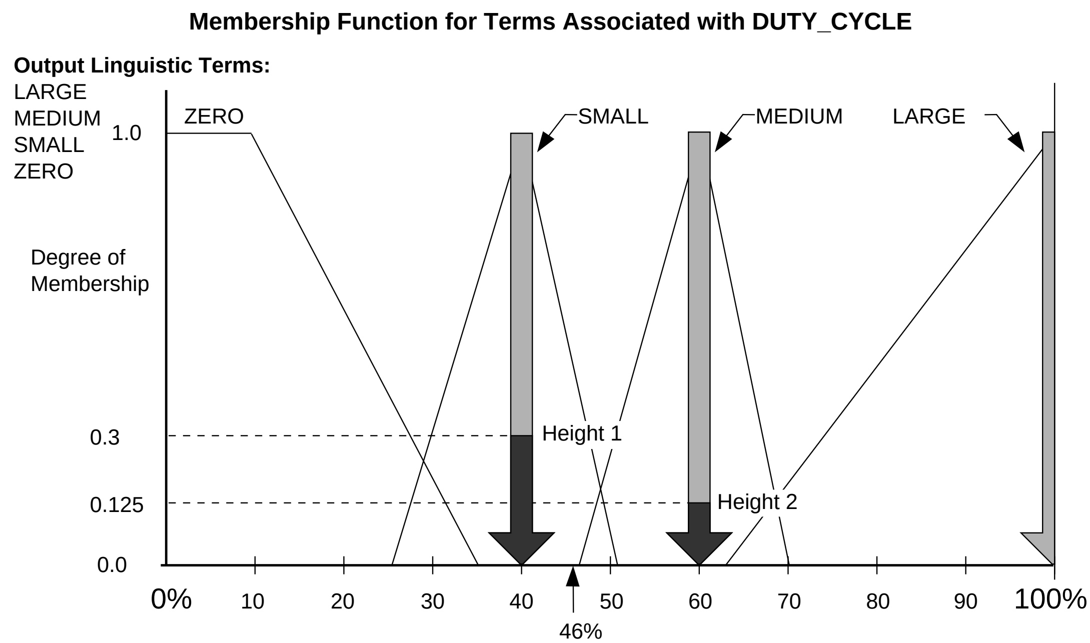
Figure 13 - Fuzzy Logic Rule Base

In the case when T_E = 3°C, Rule #3 and Rule #2 will fire. Rule #2 will fire to a degree of 0.125 and Rule #3 will fire to a degree of 0.3.

Defuzzification to Produce Duty_Cycle Values

The fuzzyTECH software uses the output membership function to defuzzify linguistic terms. In the Defuzzification Membership Function Figure 14, the fuzzyTECH software associates output linguistic terms with crisp output Duty_Cycle values which vary from zero to 100 percent.

In the case where T_E = 3°C, Rule #2 will fire to a degree of 0.125 and Rule #3 will fire to a degree of 0.3. Since Rule #2 is linked to Medium Duty_Cycle, a horizontal line at 0.125 drawn to intersect the functions representing Medium will denote the degree to which Medium will fire. Medium is denoted by Height 2. Similarly Height 1 denotes the degree to which Small Duty_Cycle will fire.



A Center-of-Maximum method is used to determine the crisp value of the DUTY_CYCLE. In the case where $T_E = 3^{\circ}C$, the crisp DUTY_CYCLE value is 46%.

Rule #3 fired and the inference result is shown by the height of the blackened portion of the arrow for SMALL.

Rule #2 fired and the inference result is shown by the height of the blackened portion of the arrow for MEDIUM.

A Center-of-Maximum method is used to defuzzify the fuzzy inputs and obtain a crisp output value for the DUTY_CYCLE. In the case of our example the DUTY_CYCLE equals 46%.

The Center-of-Maximum defuzzification process produces the following crisp output value:

$$\frac{(0.3 \cdot 40\%) + (0.125 \cdot 60\%)}{0.3 \cdot 0.125} \approx 46\%$$

Figure 14 - Defuzzification Membership Function

Building a Rule Base for Controlling Temperature

The following paragraphs describe the steps for using *fuzzy*TECH software to implement a fuzzy logic control application.

Generating a New Fuzzy Logic Control Project

From the main menu, select File > New to generate a new control project.

Specify Number of Input and Output Linguistic Variables

When Generate Project dialog box displays, enter the number of input and output linguistic variables that you want to use. (See Figure 15.)

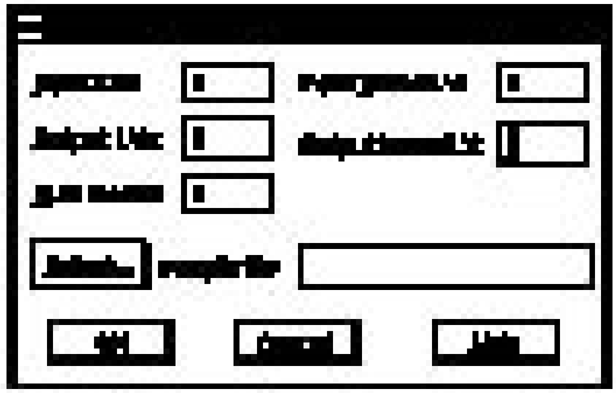


Figure 15 - Generate Project

Specify Integer Size

Select the integer size supported by the microcontroller that you will be using.

Fuzzification

The following paragraphs show how to define input linguistic variables, and input linguistic terms.

Name Your Input Linguistic Variable

Select in1 from the LV window. When the Rename Variable dialog box displays, type your own variable name (Temp_Error over the default name of in1.

Name Input Linguistic Terms

Double click on Temp_Error in the LV window to open the Membership Function Dialog Box for input linguistic terms. (Figure 16.)

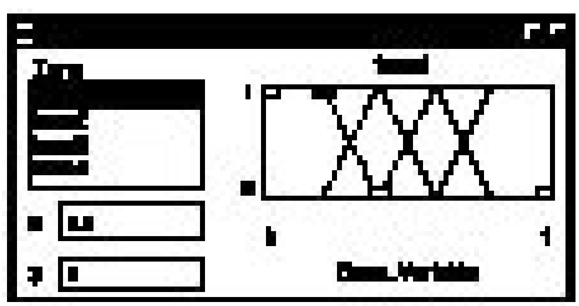


Figure 16 - Membership Function Dialog Box

Double click Base_Variable and enter the name of your base variable (Celsius). Enter the temperature range: Min = -5, Max = 20.

Now double click on term1 through term4 and type your name for each input linguistic term.

Use the rule base established previously to name the input linguistic terms:

term1 represents Hot.
term2 represents Just_Right
term3 represents Cool.
term4 represents Cold.

Fuzzy Rule Inference

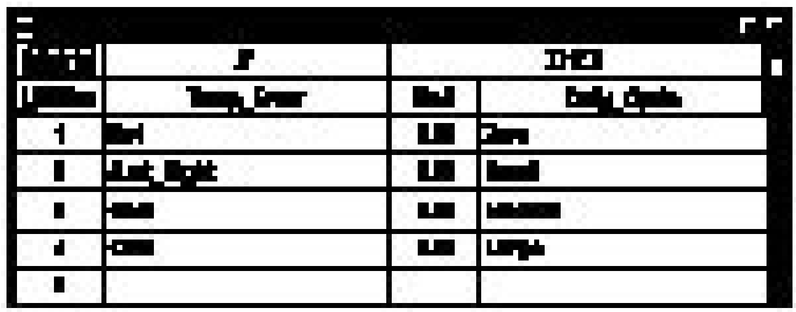


Figure 17 - Fully Configured Rule Base

The fully configured rule base illustrated in Figure 17 above shows a degree of support of 1.00 for a Zero, Small, Medium, or Large Duty_Cycle. Rule three in this illustration reads: If the Temp_Error is Cool, Then the Duty_Cycle is Medium with a degree of support of 1.00.

Aggregation

The first step in calculating the fuzzy rule inference, aggregation, determines the degree to which the complete IF-part of the rule is fulfilled. In this example, the input conditions, (the IF part) of fuzzy rule inference, contains simple statements not linked by the linguistic conjunctions, AND, or OR. Thus, the result of the aggregation calculation is either Hot, Just_Right, Cool, or Cold.

Composition

The second step in calculating the fuzzy rule inference, composition, links the validity of the entire condition with the Degree of Support (DoS). Thus, if the input Temp_Error is Cool, the output Duty_Cycle is Medium with a Degree of Support (DoS) of 1.00. The Degree of Support represents the importance of the rule, or the degree to which a given rule is plausible. Degree of Support values range between zero and one.

The *fuzzy*TECH software uses the composition product operator, PROD, to calculate the Result of the Rule. In calculating the Result of the Rule, the composition operator links the Degree of Validity of the Input Condition to the Degree of Support of the Rule.

Example:	In this example, let the Degree of Validity of the Input Condition = 0.8. From the Spreadsheet Rule Editor in Figure 3.10, observe that the Degree of Support is 1.00.
	Using the product operator for composition, the result of the rule is: Result of the Rule (Validity of the Consequence) = PROD {Degree of Validity of the Input Condition, Degree of Support of the Rule} Therefore, the Result of the Rule = PROD {0.8, 1.00} = 0.80

Defuzzification

Name Your Output Linguistic Variable

Select **out1** from the LV window. When the message, Rename Variable Dialog Box, displays, type your own output variable name (Duty_Cycle) over the default name of out1. (See Figure 18 .)

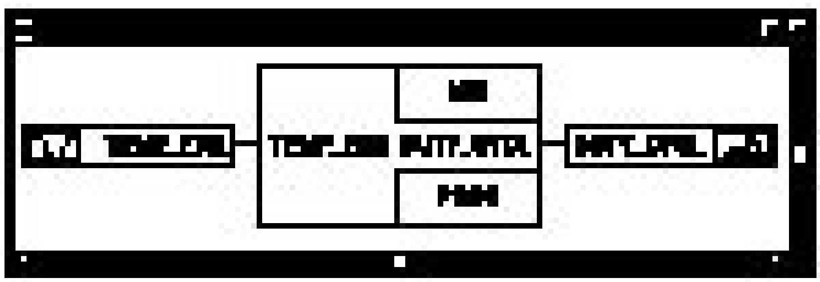


Figure 18 - Project Editor Dialog Box

Name Output Linguistic Terms

Double click on Duty_Cycle in the LV window to open the Membership Function Dialog Box for input linguistic terms. (See Figure 19.)

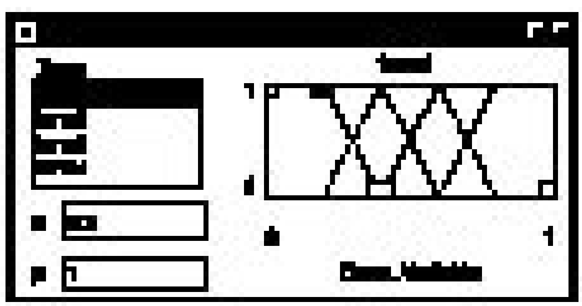


Figure 19 - Membership Function Dialog Box

Double click Base_Variable and enter the name of your base variable (Duty_Cycle). Enter the Min and Max values for Duty_Cycle: Min = 0, Max = 100.

Now double click on term1 through term4 and type your name for each output linguistic term.

Use the rule base established previously to name the output linguistic terms:

term1 represents Zero
term2 represents Small
term3 represents Medium
term4 represents Large

Editing the Rule Base

Double-click the Rule Block in the Project Editor window. See Figure 20.

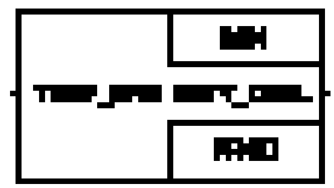


Figure 20 - Rule Block

Use your rule base established in the previous section as you select entries in the Spreadsheet Rule Editor. Delete rules that do not apply. See Figure 21.

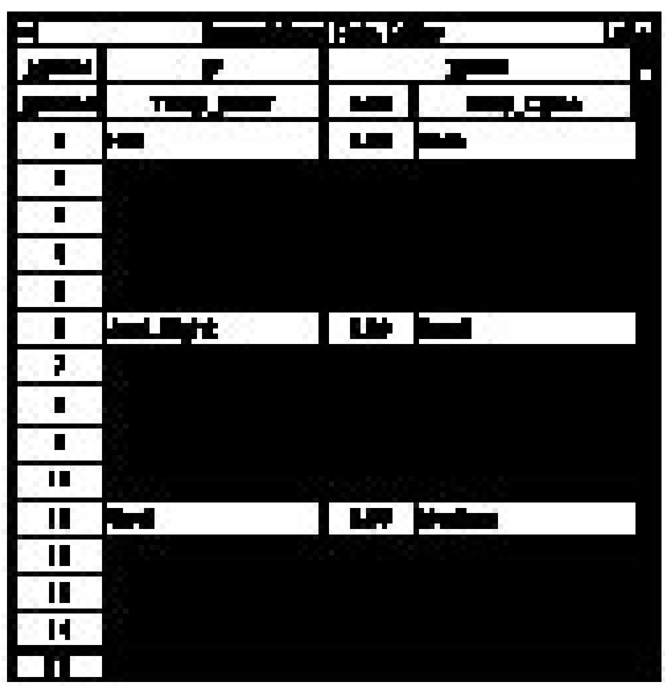


Figure 21 - Spreadsheet Rule Editor

The four remaining rules are the ones that define your control strategy. See Figure 22.

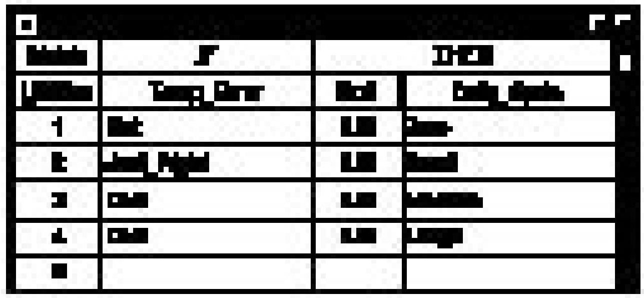


Figure 22 - Fully Configured Rule Base

Testing Your Temperature Control Strategy

After building a fuzzy logic control strategy, you can use *fuzzy*TECH to interactively test and fine tune your control strategy. The following paragraphs illustrate how to use *fuzzy*TECH to optimize temperature control on the *fuzzy*LAB demo board.

Select a Terminal Connection

From the main menu, select *Options > Terminal* to display the Terminal dialog box.

Select the Com port to which you will connect the demo board. (Com 1 through Com 4).

If you get a message "Hardware not Present" try a different port.

Select the 9600 Baud transfer rate.

Save Your Control Strategy Firmware

From the main menu, select *Options > Directories* to display the **Directories Dialog** box.

In the projects box, type C:FTMPEXPL\SAMPLE*FUZZY*LAB as the directory where you will save your control strategy.

Click **OK**, or press **Enter**.

From the main menu, select *File > Save As*. The listed directory will be the one you just entered.

In the File Name box, type TEMPCTRL.FTL as the name of your proportional control strategy for controlling the temperature on the demo board.

Running Your Fuzzy Logic Control Strategy

Perform the following steps to run your fuzzy logic temperature control strategy:

1. Press RESET on the demo board to initiate Fuzzy Proportional Control mode.
2. Press **START** on the demo board to begin running the fuzzy logic control strategy.

Available Fuzzy Logic Control Options

In FUZZY CONTROL mode, the *fuzzy*LAB demo board provides the following control options:

- Set a target temperature with the potentiometer marked **TEMP**.
- Start a control operation by pressing the **START** key.
- Interactively change control parameters using *fuzzy*TECH software.

Fine Tuning Your Control Strategy Interactively

The *fuzzy*LAB Demo Board allows you to run your control strategy while fine tuning your linguistic terms for optimum performance.

To open the input membership function (the Variable Editor), double-click **Temp_Error** in the LV window. (See Figure 23.)

The input membership function, also referred to as the **Variable Editor**, displays on the screen.

Select the linguistic term, **Just_Right**.

Observe that **Just_Right** contains small rectangular boxes that mark abrupt changes in the plot. Click on each box and compare the displayed x,y values with the x,y values on the print of your control scheme. They should match identically.

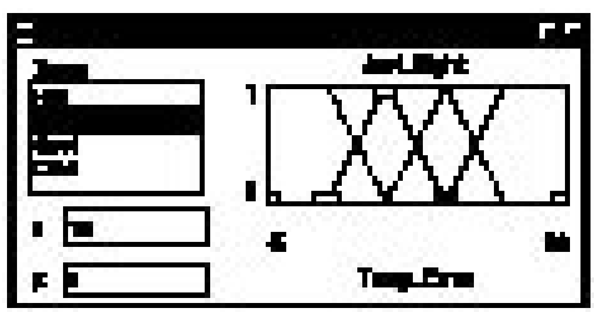


Figure 23 - Variable Editor (Membership Function)

Interactively fine tune your control strategy as the control proceeds as follows:

1. Select the linguistic term, in the Variable Editor.
2. Click one of the small rectangular boxes that define the transition points for Just_Right and drag the box to a new location on the horizontal axis (Temp_Error).

Loading Fuzzy Logic Firmware on a Microcontroller

The following paragraphs describe how to load your fuzzy logic temperature control firmware on a microcontroller. This example uses the PIC16C71 microcontroller device.

Compiling a Fuzzy Control Strategy

When you are pleased with your control strategy, select *Compile > to 16CXX* from the *fuzzy*TECH Main Menu to automatically generate fuzzy control code. You can then load this fuzzy control code into a PIC16C71 device. After loading the fuzzy control code into the PIC16C71, you can run your control strategy independently of the serial link to *fuzzy*TECH.

Preparing Fuzzy Control Source File

1. Add the generated fuzrules.asm file and the fuzrules.var file to an existing file named STNDALON.ASM.
2. After adding these two fuzzy generated files to the existing STNDALON.ASM file, assemble the new file with the MPASM assembler to create a file called STNDALON.HEX.

The resulting HEX file is the one you will download to the PIC16C71 device. Once you download a HEX control file to a PIC16C71 device and plug that device into your *fuzzy*LAB, the *fuzzy*LAB will operate independently of the *fuzzy*TECH software.

Downloading Your Fuzzy Control Firmware to a Chip

After assembling the STNDALON.ASM file to create the STNDALON.HEX file, you are ready to download the assembled HEX file to a PIC16C71 device.

1. Download the STNDALON.HEX file to either a PICSTART™ or a PRO MATE™ device programmer to embed your control strategy in a PIC16C71 microcontroller device.

Installing a Fuzzy Programmed Chip on the Demo Board

Plug the programmed PIC16C71 device into the *fuzzy*LAB board and the fuzzy control should operate as a stand alone controller.

1. Plug the programmed chip into the *fuzzy*LAB demo board.

Customer Support

Systems Information and Upgrade Hot Line

The Systems Information And Upgrade Line provides system users a listing of the latest versions of all of Microchip's development systems software products. Plus, this line provides information on how customers can receive any currently available upgrade kits. The Hot Line Numbers are: 1-800-755-2345 for U.S. and most of Canada, and 1-602-786-7302 for the rest of the world.

Connecting to Microchip BBS

You may connect worldwide to the Microchip BBS using the CompuServe™ communications network. You do not need CompuServe membership.

The procedure to connect will vary slightly from country to country. Please check with your local CompuServe agent for details if you have a problem. CompuServe service allow multiple users at baud rates up to 14400 bps.

The following connect procedure applies in most locations.

2. Set your modem to 8-bit, No parity, and One stop (8N1). This is not the normal CompuServe setting which is 7E1.
3. Dial your local CompuServe access number.
4. Depress **<Enter↵>** and a garbage string will appear because CompuServe is expecting a 7E1 setting.
5. Type +, depress **<Enter↵>** and `Host Name:` will appear.
6. Type **MCHIPBBS**, depress **<Enter↵>** and you will be connected to the Microchip BBS.
7. In the United States, to find CompuServe's phone number closest to you, set your modem to 7E1 and dial (800) 848-4480 for 300-2400 baud or (800) 331-7166 for 9600-14400 baud connection. After the system responds with `Host Name:`, type

   NETWORK, depress **<Enter↵>** and follow CompuServe's directions.

For voice information (or calling from overseas), you may call (614) 457-1550 for your local CompuServe number.

SALES AND SUPPORT

To order or to obtain information, e.g., on the pricing or delivery, please use the listed part numbers, and refer to the listed sales offices.

PART NUMBER	DESCRIPTION
DV005001	*fuzzy*TECH-MP Explorer
DV005002	*fuzzy*TECH-MP full featured Edition

WORLDWIDE SALES & SERVICE

AMERICAS

Corporate Office
Microchip Technology Inc.
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 602 786-7200 Fax: 602 786-7277

Atlanta
Microchip Technology Inc.
500 Sugar Mill Road, Suite 200B
Atlanta, GA 30350
Tel: 404 640-0034 Fax: 404 640-0307

Boston
Microchip Technology Inc.
Five The Mountain Road, Suite 120
Framingham, MA 01701
Tel: 508 820-3334 Fax: 508 820-4326

Chicago
Microchip Technology Inc.
333 Pierce Road, Suite 180
Itasca, IL 60143
Tel: 708 285-0071 Fax: 708 285-0075

Dallas
Microchip Technology Inc.
14651 Dallas Parkway, Suite 816
Dallas, TX 75240-8809
Tel: 214 991-7177 Fax: 214 991-8588

Dayton
Microchip Technology Inc.
35 Rockridge Road
Englewood, OH 45322
Tel: 513 832-2543 Fax: 513 832-2841

Los Angeles
Microchip Technology Inc.
18201 Von Karman, Suite 455
Irvine, CA 92715
Tel: 714 263-1888 Fax: 714 263-1338

AMERICAS (CONTINUED)

New York
Microchip Technology Inc.
150 Motor Parkway, Suite 416
Hauppauge, NY 11788
Tel: 516 273-5305 Fax: 516 273-5335

San Jose
Microchip Technology Inc.
2107 North First Street, Suite 590
San Jose, CA 95131
Tel: 408 436-7950 Fax: 408 436-7955

ASIA/PACIFIC

Hong Kong
Microchip Technology Inc.
Unit No. 3002-3004, Tower 1
Metroplaza
223 Hing Fong Road
Kwai Fong, N.T. Hong Kong
Tel: 852 2 401 1200Fax: 852 2 401 3431

Korea
Microchip Technology Korea
168-1, Youngbo Bldg. 3 Floor
Samsung-Dong, Kangnam-Ku,
Seoul, Korea
Tel: 82 2 554 7200 Fax: 82 2 558 5934

Singapore
Microchip Technology Inc.
200 Middle Road
#10-03 Prime Centre
Singapore 0718
Tel (mobile): 65 634 2305

Taiwan
Microchip Technology Taiwan
10F-1C 207
Tung Hua North Road
Taipei, Taiwan, ROC
Tel: 886 2 717 7175
Fax: 886 2 545 0139

EUROPE

United Kingdom
Arizona Microchip Technology Ltd.
Unit 6, The Courtyard
Meadow Bank, Furlong Road
Bourne End, Buckinghamshire SL8 5AJ
Tel: 44 0 1628 851077Fax:44 0 1628 850259

France
Arizona Microchip Technology SARL
2 Rue du Buisson aux Fraises
91300 Massy - France
Tel: 33 1 69 53 63 20Fax: 33 1 69 30 90 79

Germany
Arizona MicrochipTechnology GmbH
Gustav-Heinemann-Ring 125
D-81739 Muenchen, Germany
Tel: 49 89 627 144 0 Fax: 49 89 627 144 44

Italy
Arizona Microchip Technology SRL
Centro Direzionale Colleoni
Palazzo Pegaso Ingresso No. 2
Via Paracelso 23, 20041 Agrate Brianza (MI)
Italy
Tel: 39 039 689 9939Fax: 39 039 689 9883

JAPAN
Microchip Technology Intl. Inc.
Benex S-1 6F
3-18-20, Shin Yokohama
Kohoku-Ku, Yokohama
Kanagawa 222 Japan
Tel: 81 45 471 6166 Fax: 81 45 471 6122

MICROCHIP

 5/01/95