International Conference on Instrumentation, Communication and Information Technology (ICICI) 2005 Proc.,
August 3-5, 2005, Bandung, Indonesia

Implementation of Reconfigurable Fuzzy Logic Controller on 8-bit Microcontroller Using C-Generic Code

Faizal A. Samman[1], Ida Rachmaniar[2], Yulianti Ainuddin[2], Yustinus U. Sombolayuk
Department of Electrical Engineering, The University of Hasanuddin
Jl. Perintis Kemerdekaan Km.10 Makassar 90245. Phone/Fax: +62-411-588111
E-mail: faizalas@unhas.ac.id[1], idea_09@yahoo.com[2]

Abstract – *This paper presents the design and implementation of a fuzzy logic controller (FLC) on 8-bit microcontroller using C-generic code. The FLC can be reconfigured by modifying the generic code written in C language easily. The parameter of the input and output membership functions, the fuzzy inference rules can be reconfigured by change the parameter defined on the top of C-coded program easily to suite any control application. The generic codes of the FLC are divided into: Membership function, max and min operator, and defuzzification function. The C-generic code of the FLC has been implemented on an 8-bit Microcontroller device, Cygnal C8051f020 (MCS-51 family made Cygnal Corp.). The FLC has also been tested in term of its functionality and reconfiguration. And the test results have given control surfaces that are closely similar to MATLAB results. It seems that the C-generic code could be easily implemented on Microcontroller, and provides cost-effective efforts and short design time as well.*

Keywords – *Fuzzy Logic, Digital Controller, Microcontroller, C Language, C-generic code.*

I. INTRODUCTION

The use of embedded controller is suitable to meet real-time problem in control applications. There are several options to realize controller: microcontroller, embedded processor, digital signal processor (DSP), and programmable logic device, such field programmable gate array (FPGA) devices.
Fuzzy logic controller (FLC) system is one of control methodologies that is grow fast in industrial applications. Several works have implemented the FLC in digital mode [1,2,3,4] and in analog mode [5,6].
In digital mode using FPGA technology [1,2] the FLC can be made flexible, and easy to apply testing method. The design time can also be shortened using standard hardware description language (HDL). For mass production, the digital FLC can also be implemented on application-specific integrated circuit (ASIC) [3,4]. This approach can increase design and testing time but decrease chip cost for massive manufacturing.
Another approach to implement digital FLC is by using DSP or microcontroller. Both devices are programmable and can be configured using standard Assembly language or higher-level language such as C/C++.
Implementation using DSP, FPGA, and ASIC makes system can be operated in thousand until hundreds MHz with input-output delay time is about in nanosecond time scale. While using microcontroller the system can be operated in 10-12 MHz with input-output delay time is in millisecond time scale. However the design and chip cost using microcontroller are much lower. And the system should also be operated in control system with soft-time constraint.
This works is focused on fuzzy logic controller design and implementation based on 8-bit MCS-51 microcontroller family. This FLC is dedicated to control application which has fair/immediately-short input-output delay time.

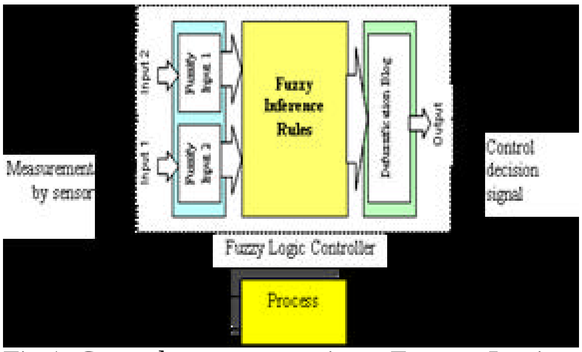


Fig.1. Control system using Fuzzy Logic Controller.

II. FUZZY MODEL

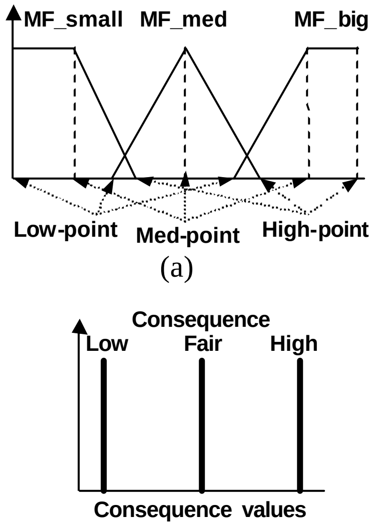
FLC on microcontroller is one of the best embedded software solutions in control problem. Design and implementation on microcontroller using C-language provides flexibility, short design-time, and cost-effective effort. Because of wide-range applications of FLC, embedded software design using C is best solution to the problem and industry challenges.

This section will provide an example of the FLC model using C-generic code. The designed FLC in this research can be reconfigured in term of:

1. Number of inputs and outputs
2. Number of membership function (MF) for inputs, and its membership shape and form
3. Number of fuzzy singleton consequences for outputs and its consequence values
4. Number of implication rules (IR), and its rules configuration.

Figure 1 shows one of the FLC model with two-input and one output. And the membership function of the FLC can be configured with three membership function as shown in Figure 2(a) for both inputs, and Figure 2(b) for its single output.

After FLC has been modeled, and the next works is modeling it using C-generic code.



(b)

Fig.2. Reconfigurable (a) the Inputs' MF, and (b) the Outputs' Consequence values.

The FLC specification is as follow:
Number of inputs is 2, number of outputs is 1, number of MF for input 1 is 3, number of MF for input 2 is 3, number of MF for output is 3, and number of IR is 9. Figure 2 shows membership form for input and output. The MF parameters are in the following table.

		Low-point	**Med-point**	**High-point**
MF for Input 1	**MF_small1**	2	24	32
	MF_med1	24	32	40
	MF_big1	32	40	50
MF for Input 2	**MF_small2**	0	25	50
	MF_med2	25	50	75
	MF_big2	50	75	100

(a)

Output consequence	**Low**	**Fair**	**High**
Consequence value	5	10	15

(b)

Table.1. (a) Membership function parameter. (b) The consequence values of the output.

If _&_ Then consequence		**Input 2**		
		small2	**med2**	**big2**
Input 1	**small1**	Low	Low	Fair
	med1	Fair	Fair	High

	big1	Fair	High	High

(a)

If _&_ Then consequence		**Input 2**		
		small2	**med2**	**big2**
Input 1	**small1**	High	High	High
	med1	Fair	Fair	Low
	big1	Low	Low	Low

(b)

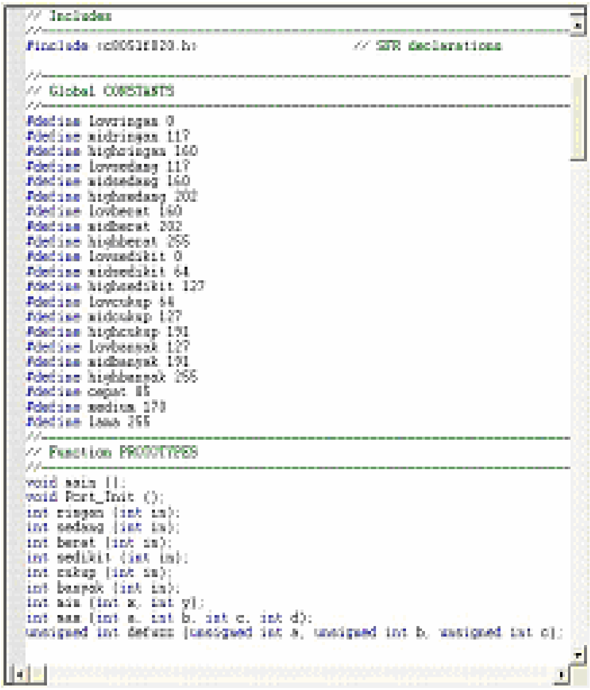
Table 2. (a) Implication rules for configuration 1. (b) Implication rules for configuration 2.

III. FUZZY C-GENERIC CODE

By understanding the structure of fuzzy system and the syntax of C, the user could derive model of the fuzzy system using the C-generic codes.

Specifications described in Table 1 and Table 2 is then translated in C-code as shown in Figure 3. All specification can be reconfigured by changing information or data defined in Figure 3 where they are written in top of the C-program.

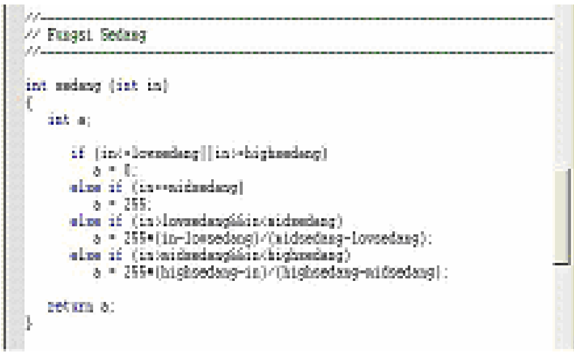
The membership function of the inputs and fuzzy singleton consequences of the output are also translated in C-code and segmented on a single C-function for every membership function. Change data specification in Figure 3 will also change C-function membership form interpretation. Figure 4 shows an example of a function of one membership function.



```
// Includes
//-----------------------------------------------------------------------------
#include <c8051f020.h>                    // SFR declarations

//-----------------------------------------------------------------------------
// Global CONSTANTS
//-----------------------------------------------------------------------------
#define lowringan 0
#define midringan 117
#define highringan 160
#define lowsedang 117
#define midsedang 160
#define highsedang 202
#define lowberat 160
#define midberat 202
#define highberat 255
#define lowsedikit 0
#define midsedikit 64
#define highsedikit 127
#define lowcukup 64
#define midcukup 127
#define highcukup 191
#define lowbanyak 127
#define midbanyak 191
#define highbanyak 255
#define cepat 85
#define medium 170
#define lama 255
//-----------------------------------------------------------------------------
// Function PROTOTYPES
//-----------------------------------------------------------------------------
void main ();
void Port_Init ();
int ringan (int in);
int sedang (int in);
int berat (int in);
int sedikit (int in);
int cukup (int in);
int banyak (int in);
int min (int x, int y);
int max (int a, int b, int c, int d);
unsigned int defuzz (unsigned int a, unsigned int b, unsigned int c);
```

Fig.3. The top list presenting C-code for fuzzy parameters.



```
//-----------------------------------------------------------------------------
// Fungsi Sedang
//-----------------------------------------------------------------------------

int sedang (int in)
{
  int a;

    if (in<=lowsedang||in>=highsedang)
      a = 0;
    else if (in==midsedang)
      a = 255;
    else if (in>lowsedang&&in<midsedang)
      a = 255*(in-lowsedang)/(midsedang-lowsedang);
    else if (in>midsedang&&in<highsedang)
      a = 255*(highsedang-in)/(highsedang-midsedang);

  return a;
}
```

Fig.4. The C-code presenting the list of a fuzzification function.

There are six generic function implemented in C-code to make this FLC can be reconfigured with flexible effort. They are: Trapezoidal membership function with positive slope, its counterpart with negative slope, Triangular membership function, Max function, Min function and Defuzzification function. The designer could use all the generic function to perform number of membership terms of the input, number of rules, configuration of rules, and number consequence values of the output. The designer could also modify FLC by adding more inputs and outputs. In this case

C8051f020 has 8 ports that could be assigned as I/O ports.
After designing FLC C-code by combining all C-function then it is compiled to produce configuration file in binary or hex format. Designing and compiling program use Cygnal IDE (Integrated Development Environment) software.

IV. SIMULATION AND TESTING RESULTS

Binary or hex format file produced from no-error compilation process is then downloaded into flash memory in the C8051f020 microcontroller. Figure 5 shows an ISP board containing C8051f020 microcontroller. This board is used to download the program and test its functional behaviors.

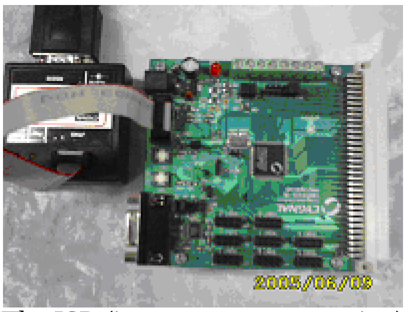
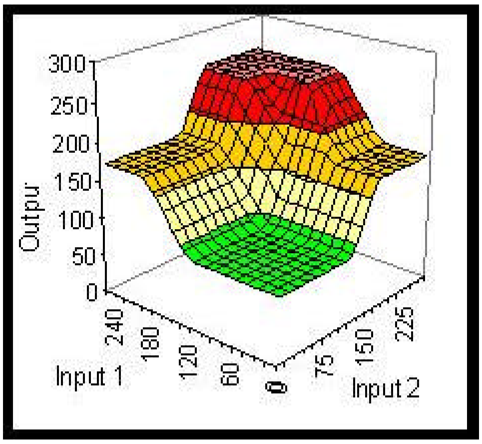
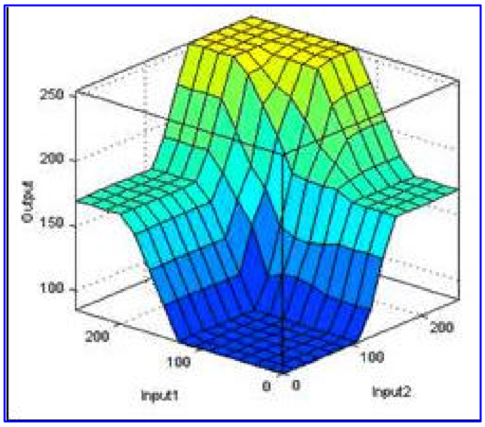


Fig.5. The ISP (in-system programming) board containing microcontroller Cygnal C8051f020.



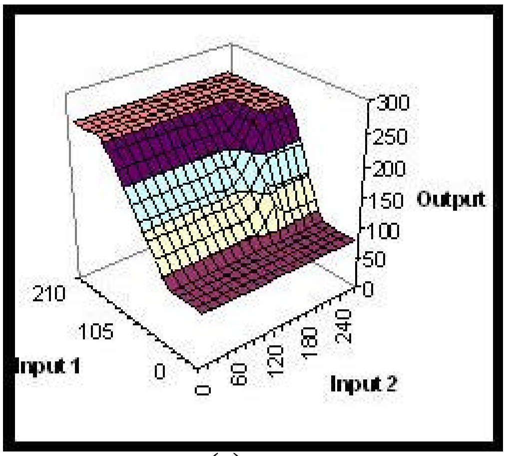
(a)



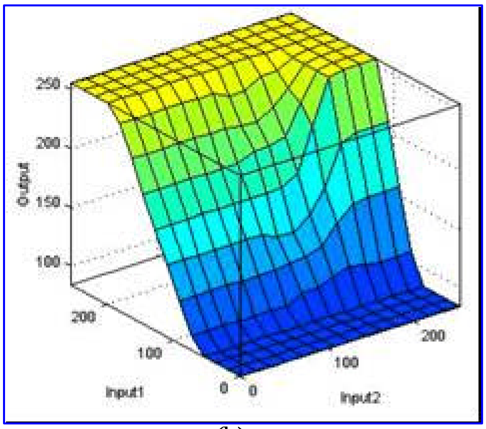
(b)

Fig.6. Control surface for configuration 1 (see Table 2(a), which is obtained from (a) microcontroller C8051f020 (actual), and (b) desired.

Figure 6 shows control surface obtained from testing the FLC rule specification example mentioned in Table 2(a). It looks that the actual surface output obtained from testing microcontroller could follow the desired control surface.
Another rule configuration as shown in Table 2(b) is tested on microcontroller. Before downloading new binary file, the FLC C-code is modified with less effort, because the C-code designed is made flexible enough to configure into another configurations.
Figure 7 shows the testing result using rule configuration 2 as exhibited in Table 2(b). It looks that control surface obtaining from testing microcontroller has similar form to desired control surface.



(a)



(b)

Fig.7. Control surface for configuration 2 (see Table 2(b), which is obtained from (a) microcontroller C8051f020 (actual), and (b) desired.

Both testing results as presented in Figure 6 and Figure 7 use 225 test vectors respectively. The functional test is done by applying binary signals into both input ports and observing the binary output via 8 LED connected into output port.

V. CONCLUSIONS

The fuzzy logic controller (FLC) has been successfully implemented on microcontroller C8051f020 using C-based programming language. The FLC can be reconfigured in term of its input's membership form, output's consequences values, and implication rules. Configuration of the FLC C-code to follow desired control surface can be done easily and less effort. Two examples of the rule configuration has been verified, and both has given good functional to follow desired control surface.

For future works, the FLC should be applied to control any specific plant with any specification. And the C-code of the FLC would be modified to suite control specification of the fuzzy system.

And also for future works, timing testing would be performed to observe input-output delay time and performance of the FLC. This works would help to determine weather the FLC can be applied into particular real-time control application with time-constraint specification.

VI.REFERENCES

[1] F.A. Samman, E.Y. Syamsuddin." Programmable Fuzzy Logic Controller Circuit Design Using VHDL-Based Synthesis". Majalah Ilmiah Teknik Elektro, ITB. Vol.7, No.3, Dec.2001.

[2] T. Holstein, S.K. Halgamuge, M. Glesner. "Computer-Aided Design of Fuzzy Systems Based on Generic VHDL Specifications". IEEE Trans. On Fuzzy Systems, Vol. 4 No. 4, Nov. 1996.

[3] H. Eichfeld, T. Kunemund, M. Menke: A 12b General-Purpose Fuzzy Logic Controller Chip. IEEE Trans. On Fuzzy Systems, Vol. 4 No. 4, Nov. 1996.

[4] M.J.Patyra, J.R. Grantner, K. Koster: Digital Fuzzy Logic Controller: Design and Implementation. IEEE Trans. On Fuzzy Systems, Vol. 4 No. 4, Nov. 1996.

[5] F.A. Samman, R.S. Sadjad. "Reconfigurable Analog Fuzzy Logic Controller Design Using CMOS Integrated Circuit Technology". Majalah Ilmiah Teknik Elektro, ITB, Vol.9, No.1, April 2003.

[6] F.A. Samman, R.S. Sadjad. "Analog Fuzzy Controller Circuit Design for Control Application". *The 4th Asian Control Conference*, Singapore. Sept.25-27, 2002.

[7] T. J. Ross: "Fuzzy Logic with Engineering Applications". McGraw-Hill, Singapore, 1995.