Fuzzy Logic Based Intelligent Control of RGB Colour Classification System for Undergraduate Artificial Intelligence Laboratory

M. F. Abu Hassan, Y. Yusof, M.A. Azmi, and M.N. Mazli

Abstract— **Fuzzy logic is one of the topic traditionally taught in artificial intelligence course. Teaching Artificial Intelligence to undergraduate engineering students can be a time consuming and expensive task. Students receive the necessary mathematical and theoretical foundation in lecture format. The final learning experience may require that students create and code their own fuzzy logic application that solves a real world problem. Typical AI lab session, the students experience the use of fuzzy logic through computer-based simulation software such as MATLAB, LABVIEW etc. or using expensive hardware that comes with pre-packaged software. Lecture sessions cover the theoretical aspect of Artificial Intelligence (AI) and laboratory exercises allow the students to visualize, experience and appreciate the application aspects of AI. This paper presents a comprehensive AI laboratory using fuzzy logic for colour sensing from applying the fuzzy inference system using MATLAB to implementation on a low-cost educational microcontroller-based system. Hardware details of the intelligent sensor and the software implementing the fuzzy logic algorithm are given in the paper.**

Index Terms— **Artificial Intelligence Course, embedded controller, fuzzy logic**

I. INTRODUCTION

Initiated in the 1960s by Lofti Zadeh, a graduate student at Columbia University [1], fuzzy logic proven to be reliable and widely accepted for industrial usages. One of the best known industrial fuzzy logic applications is the control system of the Sendai underground railway in Japan, utilized by Hitachi Company [2]. In relation to these, teaching fuzzy logic must not only cover the theoretical but also the applied aspect.

Typical AI lab session which utilizes only numerical computing software such as MATLAB, SCILAB etc. enable students to rapidly perform simulation, tune and visualize the implemented fuzzy logic algorithm. This software does not expose the students to hands-on technical skills of developing the real application such as writing fuzzy logic program using high-level language such as C, Java, and C++ etc. for embedded microcontroller.

Therefore, in this paper we propose a fuzzy logic based colour sensing application utilizing Red, Green and Blue (RGB) colours classification as part of the training system to motivate and help undergraduate students in the university to understand the underlying concepts of fuzzy logic by using MATLAB. Then, write embedded C program to be implemented on low cost microcontroller based embedded system. Fig. 1 summarizes the process of implementing the Fuzzy Logic Colour Sensing System.

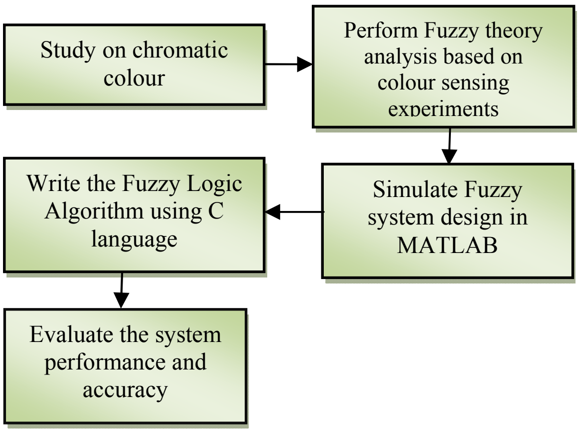


Fig. 1. Fuzzy Logic based colour sensing development process.

This paper will: a) briefly discuss the theoretical aspects of RGB colour classification and sensor design; b) demonstrate fuzzy logic simulation using MATLAB; c) show the embedded microcontroller colour sensing system hardware/software design; d) illustrate a complete working example and explain the conducted experiments; e) draw the conclusion.

II. RGB COLOUR CLASSIFICATION

The RGB colour model is an additive colour model in which the three primary colours; red, green, and blue light are combined in various ways to reproduce a wide array of colours. Fig. 2 shows the effect of these colour combination where two overlap primary colours produces secondary colours such as yellow, cyan and magenta; the combination of all three primary colours in appropriate intensities makes white.

Manuscript received March 23, 2011.

M. F. Abu Hassan is a lecturer with the Universiti Kuala Lumpur Malaysia France Institute, 43650 Bandar Baru Bangi, Selangor, Malaysia. (email: fadzil@mfi.unikl,edu.my).

Y. Yusof is a lecturer with the Universiti Kuala Lumpur Malaysia France Institute, 43650 Bandar Baru Bangi, Selangor, Malaysia. (phone: +603-8926-2022; fax: +603-8925-8845; e-mail: yusman@mfi.unikl,edu.my).

M.A. Azmi is a BET Industrial Automation and Robotics student at Department of Industrial Automation, Universiti Kuala Lumpur Malaysia France Institute.(e-mail: asyahmi87@yahoo.com).

M.N. Mazli is a BET Industrial Automation and Robotics student at Department of Industrial Automation, Universiti Kuala Lumpur Malaysia France Institute. (e-mail: cymophanex2@hotmail.com).

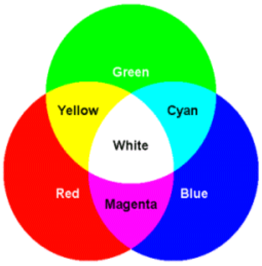


Fig. 2. A representation of additive colour mixing.

In general, a colour can be described by certain quantities, called the tristimulus values, *r* for the red component, *g* for the green component, and *b* for the blue component known as the RGB colour model is shown as follows [3]:

$$\text{Colour} = r + g + b \qquad (1)$$

Every colour in the RGB spectrum is composed of different levels for each of their red, green and blue components. The combination of these primary colour elements will affect colour result.

A. *Colour from Light*

Technically, the accurate definition of colour is: "Colour is the visual effect that is caused by the spectral composition of the light emitted, transmitted, or reflected by objects"[8]. The colour of an object depends on the coloured light rays sent to our eyes; light is necessary if we are to have any perception of colour at all. Therefore, an object is appeared to have colour because of the coloured light rays emitted and captured by the human eyes, thus sending a message to brain to analyse. An object appears red to our eye because it absorbs all other colours light and only reflects visible red colours spectrum. Fig. 3 illustrates the entire colours spectrum present in white light will affect the object colour see by the human eye.

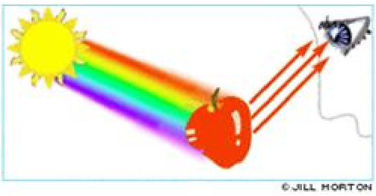


Fig. 3. Colour originates in light [8].

B. *Basic Colour Sensor*

Adopting the concept of red objects reflect red light but absorb green light; green objects reflect green light but absorb red light, a simple colour sensor can be constructed by using three different colour LED (red, green, and blue) and Light-Dependent Resistors (LDR) [5]. When a red light is shine (i.e. from a red LED) on red and green object, the red object will reflect much more light than the green object. As such, the object that appears the brightest to the sensor will be the red object.

In Fig. 4, an LDR is used to determine the LEDs' lux levels reflected from an object. The LDR is basically a resistor that changes its resistive value in ohms, Ω depending on how much value of lux received from reflected LEDs light on the object surface.

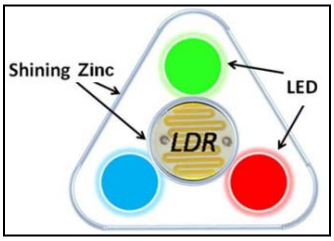


Fig. 4. Colour sensor construction.

Referring the sensor design in Fig. 5, the three coloured LEDs and LDR are enclosed using reflective material which is used to reflect as much light as possible to the LDR. Fig. 6 shows the LDR sensor circuit.

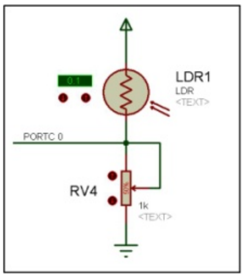


Fig. 5. LDR sensor circuit design.

The three RGB colour LEDs will be used and the object whose colour is required to be detected will be perpendicularly placed in front of the sensor system, hence the light rays reflected from the object will fall on the single LDR as shown in Fig.6.

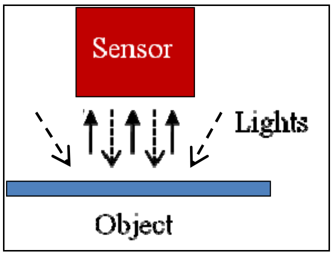


Fig. 6. Sensor and object placement.

High luminous intensity LEDs are selected as the light source and to give a better brightness. Referring to Fig. 7, the LEDs are connected to motor driver (L293D) which acts as a switching device. With the aid of the motor driver, the voltage and current are constant at the same level of output across all LEDs. Variable resistors is also added to the circuit for LED brightness calibration in order to gets the desired luminous output level.

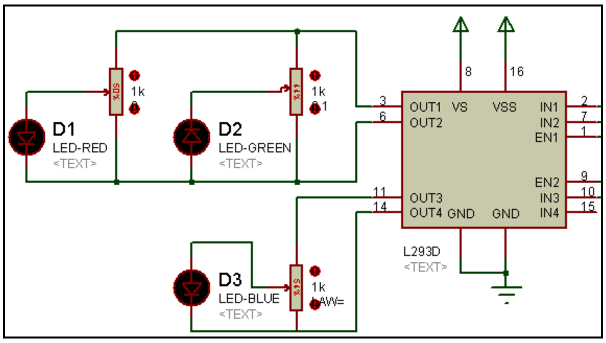


Fig. 7. LED's circuit.

Table 1: LED light source calibration on white surface

LED Colour	Voltage	Reading Data	Luminosity
Red	2.07V	424	1890 lux
Green	2.08V	426	1789 lux
Blue	2.07V	423	1892 lux

C. *Colour Sensing*

Fig. 8 shows the pseudo code to collect data from LDR [6]. Delay of 50ms is needed to make sure the amount of light from the LED is at acceptable point to be captured by LDR and stored in Microcontroller registers. Using the RGB colours shown in Fig. 9 and the pseudo code in Fig. 8, the sensor was tested and the voltage reading from the LDR circuit were recorded in Table 2-4.

```
Turn on red LED
delay 50ms
record sensor reading, R
turn off red LED

Turn on green LED
delay 50ms
record sensor reading, G
turn off green LED

Turn on blue LED
delay 50ms
record sensor reading, B
turn off blue LED
```

Fig. 8. Pseudo code for colour sensing.

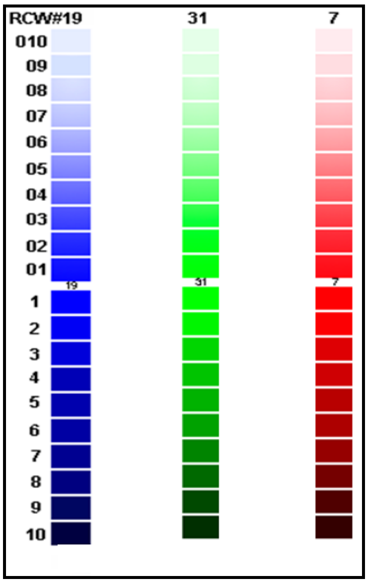


Fig. 9. RGB colour space [7].

The analogue voltage readings from the LDR sensor were then converted into 10 bit digital value using the equation (2):

$$\text{Digital Value (Data)} = (\text{Voltage Reading}/1024) \times 5\text{V} \quad (2)$$

Table 2, 3 and 4 summarizes the experiment result tested on various red, green and blue colour respectively based on RGB colour space in Fig. 9. Column RED, GREEN and BLUE represents when the coloured LEDs is turned 'ON', column 'VOLT' shows the output measured voltage and column 'DATA' represents the output converted digital value. From the results, it can be seen that the voltage reading from LDR is the highest when the colour of turned 'ON' LED is similar with the detected object. These confirms with the statement that coloured objects absorb other coloured light but reflect same coloured light.

Table 2: Sensor Data for Red Colour.

COLOUR CODE	BRIGHTNESS LEVEL	RED		GREEN		BLUE	
		VOLT	DATA	VOLT	DATA	VOLT	DATA
7/10	1	0.54	110	0.69	142	0.66	136
7/9	2	0.57	116	0.71	146	0.62	126
7/8	3	0.69	142	0.72	148	0.69	142
7/7	4	0.94	192	0.75	153	0.70	144
7/6	5	1.18	241	0.75	154	0.70	143
7/5	6	1.33	273	0.77	158	0.70	144
7/4	7	1.55	317	0.80	164	0.73	150
7/3	8	1.68	345	0.81	165	0.70	143
7/2	9	196	401	0.83	169	0.75	153
7/1	10	1.99	407	0.88	180	0.76	156
7/01	11	2.01	412	0.90	184	0.85	175
7/02	12	2.01	412	0.87	179	0.77	157
7/03	13	1.99	408	0.95	195	0.90	185
7/04	14	2.00	409	1.10	226	1.08	222
7/05	15	2.05	420	1.27	260	1.29	264
7/06	16	2.05	420	1.51	310	1.54	316
7/07	17	2.02	414	1.62	332	1.69	346

Table 3: Sensor data for Green Colour.

COLOUR CODE	BRIGHTNESS LEVEL	RED		GREEN		BLUE	
		VOLT	DATA	VOLT	DATA	VOLT	DATA
31/10	1	0.50	102	0.81	165	0.74	151
31/9	2	0.47	96	0.86	176	0.75	153
31/8	3	0.49	101	0.95	194	0.77	158
31/7	4	0.49	100	1.01	206	0.76	156
31/6	5	0.47	96	1.12	229	0.82	167
31/5	6	0.49	101	1.20	246	0.83	170
31/4	7	0.50	102	1.26	258	0.85	175
31/3	8	0.51	104	1.36	278	0.93	190
31/2	9	0.53	109	1.48	304	0.96	196
31/1	10	0.59	120	1.54	316	1.01	207
31/01	11	0.54	110	1.53	314	0.99	203
31/02	12	0.56	115	1.53	313	1.01	207
31/03	13	0.61	125	1.59	325	1.09	223
31/04	14	0.80	164	1.69	347	1.26	258
31/05	15	0.99	203	1.78	364	1.40	286
31/06	16	1.29	265	1.87	382	1.58	324
31/07	17	1.44	295	1.88	386	1.64	336

Table 4: Sensor Data for Blue Colour.

COLOUR CODE	BRIGHTNESS LEVEL	RED		GREEN		BLUE	
		VOLT	DATA	VOLT	DATA	VOLT	DATA
19/10	1	0.48	98	0.73	149	0.78	159
19/9	2	0.46	94	0.72	148	0.83	170
19/8	3	0.45	92	0.76	155	0.89	183
19/7	4	0.46	95	0.73	149	0.92	189
19/6	5	0.45	92	0.74	152	0.93	191
19/5	6	0.45	93	0.73	150	0.96	197
19/4	7	0.46	94	0.74	151	0.97	199
19/3	8	0.46	94	0.80	163	1.06	218
19/2	9	0.45	93	0.81	165	1.12	230
19/1	10	0.48	99	0.91	187	1.27	260
19/01	11	0.43	89	0.85	174	1.18	242
19/02	12	0.46	94	0.89	182	1.24	253
19/03	13	0.51	105	1.05	215	1.44	294
19/04	14	0.60	122	1.17	239	1.47	302
19/05	15	0.76	155	1.34	274	1.61	329
19/06	16	1.02	208	1.50	308	1.71	351
19/07	17	1.31	269	1.66	339	1.81	371

Data from experiments done on the colour sensor confirms that the sensor works and can be use to differentiate the three different primary colours. Based on these preliminary results, a fuzzy logic system was designed and will be discussed in the next section.

III. FUZZY LOGIC COLOUR SENSING

The MATLAB Simulink is used for rapid fuzzy logic inference design and system testing. This tool is a quick view process to the students to understand the concept of fuzzy logic design. The sensor digital value output will be fed by the microcontroller to MATLAB software via serial communication as shown in Fig. 10. The acquired digital data (Red, Green and Blue value) will be sequencely captured by MATLAB and used as Fuzzy sets input.

Fig. 10. Colour Sensor-MATLAB setup.

A. Fuzzy Sets

Fuzzy control system design is based on empirical methods. The membership function is a graphical representation of the magnitude of participation of each inputs. It associates a weighting with each of the inputs that are processed, define functional overlap between inputs, and ultimately determines an output result [9]. Three fuzzy sets for input and one fuzzy set for output using triangular membership function are created as shown in Fig. 11.

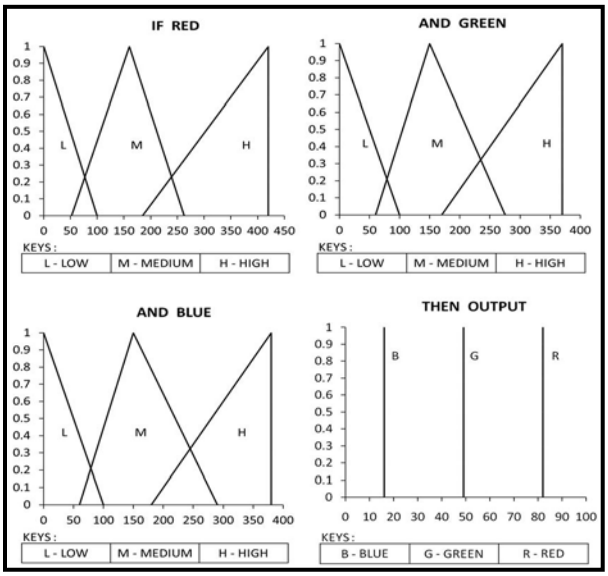


Fig. 11. RGB membership functions.

In this case, the Sugeno-type Fuzzy Interference was chosen for the process of formulating the mapping from the given inputs to the output. For defuzzification process, all consequent membership functions are represented by singleton spikes. The weighted average (WA) of these singletons is used to get the crisp output. The equation is given below:

$$WA = \frac{[\Sigma\,\mu(\mathrm{BLUE})]\mathrm{BLUE} + [\Sigma\,\mu(\mathrm{GREEN})]\mathrm{GREEN} + [\Sigma\,\mu(\mathrm{RED})]\mathrm{RED}}{[\Sigma\,\mu(\mathrm{BLUE})] + [\Sigma\,\mu(\mathrm{GREEN})] + [\Sigma\,\mu(\mathrm{RED})]} \tag{3}$$

B. Rules

Linguistic rules describing the control system consist of two parts; an antecedent block (between the IF and THEN) and a consequent block (following THEN) [9]. In this system, it may not be necessary to evaluate every possible input combination since some may rarely or never occur. Table 5 shows the number of rules and they are determined based on the logical thinking of the designer. The rules use the input membership values as weighting factors to determine their influence on the fuzzy output sets of the final output conclusion. It accommodates three input variables and expresses their logical product (AND) as one output variable.

Table 5: Fuzzy Set Rules.

RULE	IF RED IS	AND GREEN IS	AND BLUE IS	THEN OUTPUT IS
1	Low	Low	Medium	Blue
2	Low	Low	High	Blue
3	Low	Medium	High	Blue
4	Medium	Low	High	Blue
5	Low	Medium	Low	Green
6	Low	High	Low	Green
7	Low	High	Medium	Green
8	Medium	High	Low	Green
9	Medium	Low	Low	Red
10	High	Low	Low	Red
11	High	Medium	Low	Red
12	High	Low	Medium	Red
13	Medium	Medium	High	Blue
14	Medium	High	Medium	Green
15	High	Medium	Medium	Red

Tuning the system can be done by changing the rule antecedents, changing the centers of the input and/or output membership functions, or adding additional degrees to the input and/or changing the colour level output functions. The final RGB Fuzzy system interface using MATLAB Simulink is shown in Fig. 12.

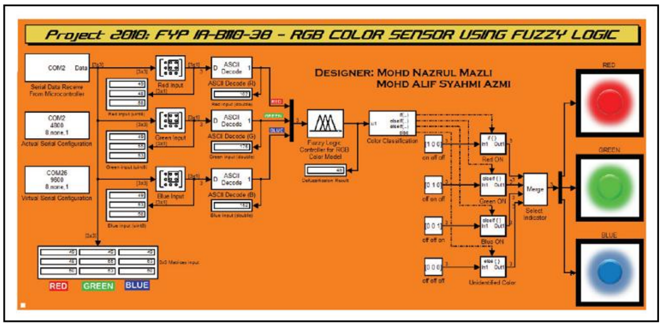
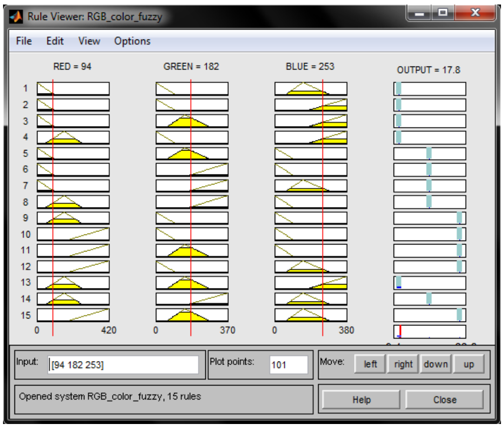


Fig. 12. Fuzzy Logic using MATLAB Simulink interfaced with colour sensor.

IV. EXPERIMENTAL RESULT

Seventeen colour tones for each Red, Green and Blue colours were tested on this fuzzy system. The following Fig. 13 show the sample of experimental result for colour code 19/2 captured from MATLAB Simulink and Table 6-8 resume the overall experimental result.



Input: [94 182 253]
Plot points: 101
Move:

Opened system RGB_color_fuzzy, 15 rules

Fig. 13. Experimental result for RGB colour classification using Fuzzy Logic in MATLAB

Table 6: Experimental Result for Red Colours

COLOUR CODE	RED	GREEN	BLUE	FUZZY RESULT	
	DATA	DATA	DATA	DATA	COLOUR
7/10	110	142	136	X	X
7/9	116	146	126	X	X
7/8	142	148	142	X	X
7/7	192	153	144	82	Red
7/6	241	154	143	82	Red
7/5	273	158	144	82	Red
7/4	317	164	150	82	Red
7/3	345	165	143	82	Red
7/2	401	169	153	82	Red
7/1	407	180	156	82	Red
7/01	412	184	175	82	Red
7/02	412	179	157	82	Red
7/03	408	195	185	82	Red
7/04	409	226	222	82	Red
7/05	420	260	264	82	Red
7/06	420	310	316	X	X
7/07	414	332	346	X	X

Table 7: Experimental Result for Green Colours

COLOUR CODE	RED	GREEN	BLUE	FUZZY RESULT	
	DATA	DATA	DATA	DATA	COLOUR
31/10	102	165	151	X	X
31/9	96	176	153	49	Green
31/8	101	194	158	49	Green
31/7	100	206	156	49	Green
31/6	96	229	167	49	Green
31/5	101	246	170	49	Green
31/4	102	258	175	49	Green
31/3	104	278	190	49	Green
31/2	109	304	196	49	Green
31/1	120	316	207	49	Green
31/01	110	314	203	49	Green
31/02	115	313	207	49	Green
31/03	125	325	223	49	Green
31/04	164	347	258	49	Green
31/05	203	364	286	49	Green
31/06	265	382	324	X	X
31/07	295	386	336	X	X

Table 8: Experimental Result for Blue Colours

COLOUR CODE	RED	GREEN	BLUE	FUZZY RESULT	
	DATA	DATA	DATA	DATA	COLOUR
19/10	98	149	159	X	X
19/9	94	148	170	X	X
19/8	92	155	183	16	Blue
19/7	95	149	189	16	Blue
19/6	92	152	191	16	Blue
19/5	93	150	197	16	Blue
19/4	94	151	199	16	Blue
19/3	94	163	218	16	Blue
19/2	93	165	230	16	Blue
19/1	99	187	260	18	Blue
19/01	89	174	242	16.9	Blue
19/02	94	182	253	17.8	Blue
19/03	105	215	294	16	Blue
19/04	122	239	302	16	Blue
19/05	155	274	329	16	Blue
19/06	208	308	351	X	X
19/07	269	339	371	X	X

By referring to Table 6-8, the result shows colour testing on various RGB colour tone levels from dark to light colours. The colour sensor system succeeded to classify 70.5% for red, 82.3% for green and 76.5% for blue. From the observation, the system is not recognized most of the brightest and darkest colour level for each colours. Based on this result, the system is able to classify the primary colour and the fuzzy logic design will be used in writing the software for microcontroller that will be discussed in the next section.

V. MICROCONTROLLER BASED COLOUR SENSING SYSTEM

The C program is written by using the AVR Studio, AVR-GCC 4.1 compiler and the code is embedded into ATMEGA 16 microcontroller. The successful Fuzzy Logic algorithm simulated in MATLAB is translated inform of C language. The algorithm of the fuzzy control system is shown in Fig. 14 as a Program Description Language (PDL).

```
BEGIN
  DO FOREVER
      Trigger On Red LED
      Read analogue LDR sensor and convert to digital
      Save Red digital value into register
      Transmit Red digital value to PC
      Trigger Off Red LED
      Trigger On Green LED
      Read analogue LDR sensor and convert to digital
      Save Green digital value into register
      Transmit Green digital value to PC
      Trigger Off Green LED
      Trigger On Blue LED
      Read analogue LDR sensor and convert to digital
      Save Blue digital value into register
      Transmit Blue digital value to PC
      Trigger Off Blue LED
      Calculate membership function
      Apply Fuzzy Logic rules
      Calculate the crisp output
      Trigger Red or Green or Blue indicator
   ENDDO
END
```

Fig. 14. RGB colour sensing with Fuzzy control flow.

Below is part of program for RGB colour sensing using Fuzzy Logic.

```
if (red_input <= 100)//LOW
    red[1] = calculateM(red_input, 100,0);
if ((red_input >= 52) && (red_input <= 160)//MED
    red[2] = calculateM(red_input, 52,160);
if ((red_input >= 160) && (red_input <= 263//MED
    red[2] = calculateM(red_input, 263,160);
if ((red_input >= 185) && (red_input<=420))//HIGH
    red[3] = calculateM(red_input, 185,420);
```

Fig. 15. Fuzzy membership for red input.

```
float tempC = 0, tempD = 0, output;
int i;

for (i=1; i<16; i++)
            tempC = tempC + tempB[i];
for (i=1; i<16; i++)
            tempD = tempD + tempA[i];
output = tempC/tempD;
```

Fig. 16. Weighted Average (WA) defuzzification method.

A discrete input, six discrete outputs, a serial port and an analogue input are the specification needed to build the control circuit of the system. The block diagram of system hardware architecture is illustrated in Fig. 17 and the actual system hardware is shown in Fig. 18.

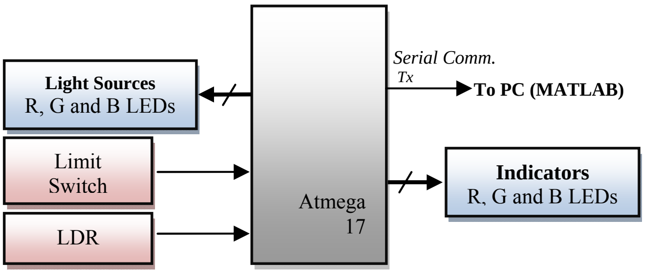


Fig. 17. System hardware architecture

Fig. 18. The RGB colour classification using Fuzzy Logic kit.

From this standalone microcontroller based fuzzy system, the output result will be compared and verified with the MATLAB simulation. The user can directly monitor the result from the hardware indicators (LEDs) and observe the fuzzy system process through MATLAB simulation simultaneously.

VI. CONCLUSION AND FUTURE WORKS

Fuzzy logic controllers have become popular in recent decades with successful implementation in many diverse fields and consequently many technical colleges and universities are now offering fuzzy logic courses. Some limitation on certain software and teaching-aid may limit the student understanding on the course. This project presents the development of low-cost educational toolkit for Fuzzy logic laboratory. In this research project, the development of the colour sensing using Fuzzy Logic system may contribute on the learning process for undergraduate engineering students. With this teaching aid, the student can easily understand Fuzzy Logic algorithm and enhance their hand-on skill in solving engineering problems.

From this project, the Fuzzy Logic system is able to classify the primary RGB colour but with some limitation on the sensor, the system is unable to recognize the brightest and darkest colour level for each colour. Thus, the enhancement on the sensing element will be focused in the future for better wide range spectrum colour detection.

REFERENCES

[1] Timothy J. Ross, *Fuzzy Logic with Engineering Applications*, McGraw Hill, 2001.

[2] Kazuo Tanaka, *An Introduction to Fuzzy Logic for Practical Applications*, Springer, 1996.

[3] Naotoshi Sugano, “Fuzzy Set Theoretical Approach to the RGB Triangular System,” *Journal of Japan Society for Fuzzy Theory and Intelligent Informatics*, vol.19, pp. 31-40, 2007.

[4] Michigan State University: Department of Chemistry (2011, February). Visible and Ultraviolet Spectroscopy [Online]. Available:http://www2.chemistry.msu.edu/faculty/reusch/VirtTxtJml/Spectrpy/UV-Vis/ spectrum.html

[5] Philippe E. Hurbain and Michael Gasperi, *Extreme NXT: Extending the LEGO Mindstorms NXT to the Next Level*, Apress, 2007.

[6] Society of Robots. (2011, February), Colour Sensors Tutorial [Online]. Available:http://www.societyofrobots.com/sensors_colour.shtml

[7] Don Jusko. (2011, February), Original Real Colour Wheel Pen Tablet Palettes in RGB [Online]. Available:http://www.realcolourwheel.com/rcwpentablet.html

[8] Colour Matters. (2011, March), How the Eye Sees Colour [Online]. Available:http://www.colourmatters.com/seecolour.html

[9] Encoder. (2010, December), Fuzzy Logic Tutorial [Online] Available:http://www.seattlerobotics.org/encoder/mar98/fuz/flindex.html