"ROBOTS! - INTRODUCTION TO ENGINEERING AND COMPUTER SCIENCE"

Matthew Young Lionel Love, Jeffrey R. Anderson, Steven F. Barrett
Electrical and Computer Engineering
University of Wyoming

Abstract

For close to three decades, Wyoming rising sophomore students have had the opportunity to have their first taste of college life while attending the University of Wyoming Summer High School Institute (HSI). For over ten years we have team taught two courses to expose students to computer hardware and software programming techniques. We were challenged to combine the course into a single offering. Our goal was to develop an intensive, hands-on, motivational experience where each student would build, program, and develop the interface between the programming board and the robot hardware. We hoped that along the way the students would learn about different engineering fields, computer science, and also the basics of computer programming and interfacing. The course concluded with a robot competition. Students competed to see which robot could go through an unknown maze without bumping into maze walls in the shortest time. The course objectives included: 1) Take the mystery out of engineering and computing, 2) Show that engineering and computer science is fun and exciting, 3) Demonstrate that engineering is for both women and men, 4) Emphasize hands-on, learning by doing exercises, and 5) Inform students of the excellent educational opportunities at the University of Wyoming. For the course a low cost Dagu Magician robot kit was used. Each student built their own robot. The robot was equipped with infrared sensors and an Arduino UNO R3 processor board. The Arduino processor was specifically designed for use by students without a technical background. They are easy to use and program. Students completed a pre- and post-course survey to determine their level of understanding in a number of course concepts. A substantial increase in student understanding was noted.

Overview

For many decades the University of Wyoming has offered the Summer High School Institute for rising sophomore students. Typically 100 students are competitively selected from throughout the state to attend the three week residential program. Students are required to complete an arts or sciences and a math or engineering course during the program. Classes meet for 27 contact hours (twelve sessions of 2.25 hours each) over the three week course [1].

For over ten years the authors have participated in the HSI Program. Initially two separate 27 contact hour courses concentrating on computer programing using Lego robots and computer hardware were offered. Typical enrollment for each course was 12 students. Students tended to lose interest in the courses by the third week. Also, students were not exposed to both software and hardware concepts. As a remedy we shortened both courses to 13.5 contact hours and allowed students to complete both shortened courses over the three week period.

Due to budget challenges the HSI Program was recently scaled back. We were challenged to develop a single 27 contact hour course for 12 students. We took this as a challenge to significantly update the course with a low-cost robot platform controlled by an Arduino processor. Our goal was to develop a low cost, take home kit to spark interest in computer science, computer engineering, and electrical engineering. In this paper we provide background information on kit contents including the Arduino processor and programming environment and the Dagu Magician robot platform. In the Methods section we discuss curriculum developed for the course. We then provide and discuss results based on student surveys and conclude with recommendations for improvement.

Background

To develop a low cost, motivational robot platform for the HSI course we chose the Dagu Magician Robot equipped with an Arduino UNO R3 controlled by an algorithm written using an Arduino Sketch. The design was modified from one originally appearing in "Arduino Microcontroller: Processing for Everyone! [2]." In this section we describe the robot design, interface circuits required, the Arduino UNO R3, and the control algorithm.

The overall goal was to develop an autonomous, low cost robot that could navigate through a maze as quickly as possible without bumping into any maze walls. This was based on the maze competition originally introduced by Trinity University some time ago. Background on related efforts is provided in a number of references and will not be repeated here. The interested reader is referred to [3-12]. What is novel about the present work is the concentration on a low cost, take home design, and the age of the participants.

Dagu Magician Robot

The Dagu Magician Robot is a low cost (approximately US$20), two acrylic plastic platform robot equipped with two motorized wheels and a drag ball that provides tripod stability. The Dagu kit contains hardware to stack the platforms and mount the motorized wheels. The top platform was equipped with a locally manufactured aluminum bracket to mount three Sharp infrared proximity sensors (Sharp GP2Y0A21YK). The top platform was also equipped with the Arduino UNO R3 processor using a standoff kit (Jameco #106551) and also a 3.3" x 2.1" solderless breadboard (Jameco #20601). The breadboard provides a prototype area for the sensors and motors interface electronics. Provided in Figure 1 is the assembled robot.

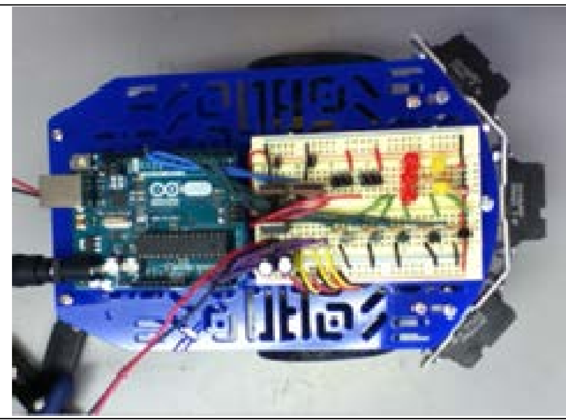

Figure 1. Assembled DAGU Magician Robot.

Sensors and Motor Interface Hardware

The Arduino UNO R3 is a 5 VDC platform with limited current source and sink capability. Interface circuits were developed to interface the IR sensors and the robot motors to the Arduino processor. The interface circuit is provided in Figure 2. The IR sensors left, middle, and right are directly connected to the Arduino analog-to- digital converter channels A0, A1, and A2 [2].

The Dagu robot is equipped with 6.0 VDC motors with a no load current of 190 mA and a stall current of 1 A [13]. To interface the motor control signals from the Arduino processor an interface circuit was required. The interface for each motor consists of an NPN Darlington pair transistor (TIP120) and also four 1N4001 rectifying diodes. Three of the diodes are used to reduce the voltage from the 9 VDC outboard power supply down to the motor's operating voltage. Another 1N4001 is used for reverse voltage protection [2].

The robot was also equipped with five LEDs to indicate when a wall had been sensed by the individual IR sensor and also to provide turn signals for the robot. A 2N2222 small signal

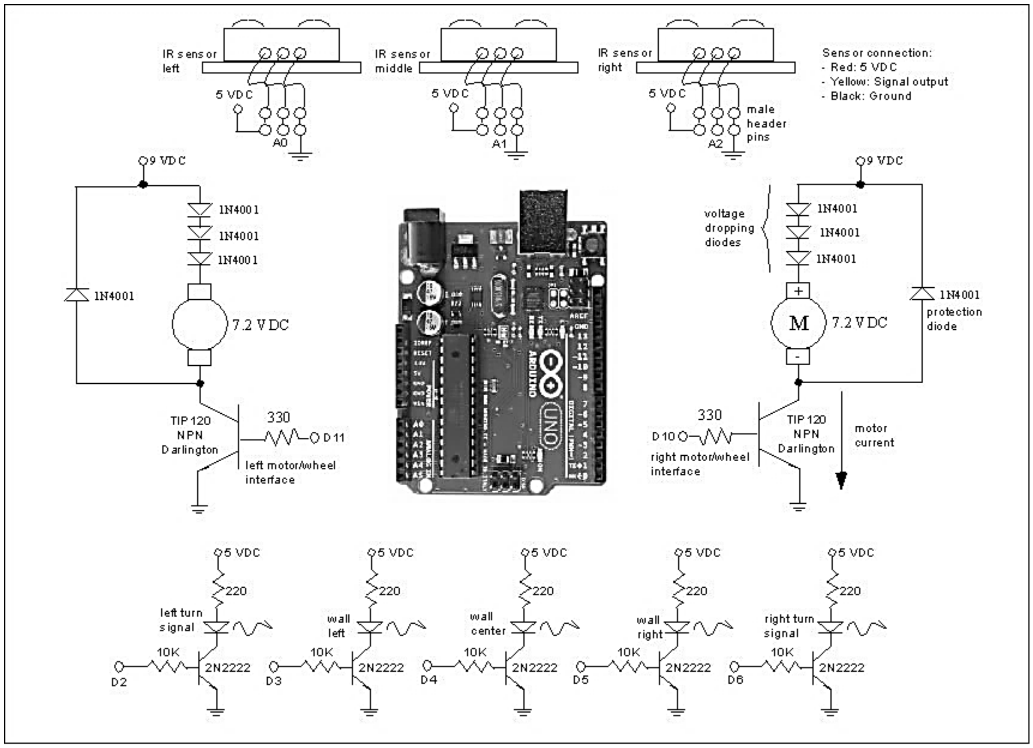

Figure 2. Sensors and Motor Interface Hardware [adapted from 2].

transistor with biasing resistors was used to interface each LED to the Arduino processing board [2]. The Arduino processing board hosts the Atmel ATmega328. Atmel documentation indicates each processor pin can sink/source 20 mA. However, pin characteristics indicate when sourcing 12-15 mA current as required by an LED, the pin voltage is pulled up or down as determined by the resistive interface circuit chosen. A transistor based interface provides a conservative interface circuit while exposing students to the concepts of interfacing hardware and transistor operation.

Power Supply. Rather than using onboard batteries, the robot was equipped with a 9 VDC, 2A outboard power supply (Jameco #1952847). A flexible umbilical cable is used to connect the power supply to the robot as it navigates about the maze. An onboard 5 VDC regulator (LM7805) is used to step down the 9 VDC motor supply to 5 VDC for the Arduino processor board and the IR sensors.

Arduino UNO R3. The Arduino line of processors was developed to allow those without a background in microcontrollers to easily incorporate processing power into projects. The Arduino, open source concept has become a worldwide phenomenon. For this project we chose the Arduino UNO R3 processing board [14]. The R3 hosts the Atmega328 processor. This is a 28-pin processor with a full complement of common microcontroller subsystems [15]. The R3 contains an onboard voltage regulator, timing source, and USB support for a host computer. The R3 provides female connectors to easily interface the processor to peripheral devices.

Arduino Sketch. The Arduino Sketch environment provides a free, open source method of programming the Arduino line of processors. The environment includes a wide variety of software routines to easily interface the R3 to a host of peripheral devices and onboard systems. Due to the open source nature of the software, a number of new applications are constantly being added to the Arduino library.

Control Algorithm. As a starting point, students were provided a basic robot control algorithm from "Arduino Microcontroller: Processing for Everyone! [2]." A portion of the control algorithm is provided in Appendix 2 with the permission of Morgan and Claypool Publishers. The algorithm provides the basic maze navigation features; however, it requires modification to allow for successful navigation.

Methods

Two electrical and computer engineering faculty members and a senior undergraduate mechanical engineering student developed the curriculum for the course. Our goal was to provide enough theory to allow the students to progress rapidly in the laboratory exercises. The curriculum was divided into ten 2.25 contact hour sessions. We also planned a related session on supercomputers and modeling. The session on supercomputers was taught by a Department of Mathematics faculty member. Also, a field trip was taken to the National Center for Atmospheric Research (NCAR) - Wyoming Supercomputing Center (NWSC) and a nearby high technology Walmart regional distribution center. The developed curriculum overview is provided in Table 1.

Table 1. Curriculum Schedule.

Week 1		
Lesson	**Topics**	**Lab Activities**
Tuesday 1:45-4:00 PM	- Course overview - Robots - Maze competition - Engineering - Computer Science - Gears - Motors - Motor interfacing - End of day feedback	Assemble robot kit
Wednesday 1:45-4:00 PM	- Transducers - IR sensor theory - End of day feedback	Assemble robot kit ADC: characterize IR sensor, print to screen
Thursday 1:45-4:00 PM	- High Performance Computing discussion and activities	
Friday 11:30-5:00 PM	- Computer related tours o NCAR Wyoming Supercomputing Center (NWSC) o Walmart Distribution Center	
Week 2		
Lesson, date	**Topic**	**Lab Activities**
Monday 1:45-4:00 PM	- Arduino programming - Programming environment - LEDs under Arduino control - Counting program - Counting program with LEDs - End of day feedback (5 minutes)	Finish robot Blink an LED under Arduino Control
Tuesday	- LEDs	Illuminating LEDs

1:45-4:00 PM	- LED interface circuits
	- Transistors
	- Processors
	- Programming basics
	
	- Arduino programming
	- Analog-to-digital conversion
	  o Input from IR sensors
	- Pulse width modulation (PWM) theory
	  o PWM motor speed control
	
	- End of day feedback (5 minutes)
		ADC: characterize IR sensor, print to screen
		PWM: control motor speed with PWM

Thursday 1:45-4:00 PM	- Arduino programming – introduction to the maze
	- Top down design, bottom up implementation
	  o Block diagram (structure chart)
	  o Flow chart
	- End of day feedback (5 minutes)
	-
		Assemble sensor interface

Friday 1:45-4:00 PM	- Sensors: giving your robot "eyes," sensor interface - End of day feedback (5 minutes)	Assemble sensor interface

Week 3		
Lesson, date	**Topic**	**Lab Activities**
Monday 1:45-4:00 PM	- Motors: giving your robot "legs," motor interface - End of day feedback (5 minutes)	Assemble robot interface
Tuesday 1:45-4:00 PM	- Linking "eyes" to "legs," control algorithm development - End of day feedback (5 minutes)	Basic control algorithm
Thursday 1:45-4:00 PM	- Full up control algorithm - Develop maze maps - End of day feedback (5 minutes)	Maze dry runs
Friday 1:45-4:00 PM	Robo Wars – timed maze competition	

Results

On the last day of the course, each student (n=13) was given a short survey asking them to rate six different aspects of the course (0: none ---5: some---10: significant) . On the last two questions the ranking metric was (0: strongly disagree ---5: Neutral---10: strongly agree). The survey results are provided in Table 2.

Table 2. End of course survey results.

	Ave	SD
How much did you know about **robots** before taking this course?	2.00	2.91
After completing this course, how much do you feel you know about **robots**?	6.20	1.99
How much did you know about **software programming** before taking this course?	1.10	2.60

After completing this course, how much do you feel you know about **software programming**?	5.30	2.26
How much did you know about **digital electronics** before taking this course?	1.60	1.84
After completing this course, how much do you feel you know about **digital electronics**?	5.90	2.08
I am glad I took this course "Robots! - Introduction to Engineering and Computer Science" at the High School Institute.	8.00	2.31
I would recommend taking this course to future High School Institute attendees.	7.90	2.38

In addition, students were provided open-ended questions. The four questions are provided below with selected, representative answers.

What did you like best about the course?

- It was just something new to learn. I had zero experience coming into the class. I feel that I have learned quite a bit.
- I liked the professors. I felt they were always engaging. They were always there to assist me.
- I like how now I know that the math I'm learning in high school-not here- has a use and isn't useless.

What did you find most challenging?

- Trying to keep up with all we were learning.
- The most challenging thing was learning and understanding coding. I still am not 100% capable of writing code on my own but have learned a lot.
- Circuitry and electronics are hard and seem to run on some sort of black magic. Great job explaining though.
- Problem solving when my robot didn't work
- I found wiring the robot most challenging.

What can we do to improve the course?

- Do more hands-on teaching instead of the lecture at the beginning of class.
- Help students understand programming. Hit programming harder!
- I felt a bit rushed near the end to complete my robot. A lot of class time was lectures. It's not that lectures are bad. It's just that we would be working and we'd be stopped to listen to the professors.
- If you could do a simple lecture when we first start where you hold up the piece of the machine/programming that you're talking about and then draw the picture (schematic).

Anything else you would like us to know concerning the course?

- I want to get into robotics now!
- I had a lot of fun learning this new topic and I am excited to show my family.
- Explain the process of putting the robot together better. The "just go and do it" didn't help. You taught us some circuits while other stuff made no sense to me.
- I love this course, I hate math but mostly because I don't understand it and I find no use for it but after this course I now have a use for math; so thank you so much.

Discussion

Overall we were quite pleased with the results of the inaugural offering of this course. Students came to the course with a wide background and interest in topic areas related to the course. As a group, the students indicated an increased knowledge of robotics, programming, and digital electronics as a result of the course. We also found the daily feedback session instrumental in planning our instructional plan for the following day. We were encouraged by the positive feedback received and will modify the course to address some of the concerns mentioned.

Several students indicated discomfort with the compressed schedule. We will work to improve the content, timing, and delivery of the mini-lectures presented throughout the course.

Conclusions

We are very excited about the opportunity to teach this course again in June 2016. Our future plans are to modify the curriculum to address some of the suggestions provided in the end-of-course student survey. We are also investigating a lower cost, easy-to-use microcontroller platform and software suite.

We are encouraged by the results of this inaugural offering of the course. We believe our objectives of 1) Taking the mystery out of engineering and computing, 2) Showing that engineering and computer science is fun and exciting, 3) Demonstrating that engineering is for both women and men, 4) Emphasizing hands-on, learn by doing exercises, and 5) Informing students of the excellent educational opportunities at the University of Wyoming have been achieved.

Acknowledgements

This educational project was funded by the John P. Ellbogen Foundation via the College of Engineering and Applied Science's Engineering's Next Generation Program. We especially thank Mary Ellbogen Garland, President and Chair of the Foundation.

References

1. Summer High School Institute, www.uwyo.edu/hsi/

2. S.F. Barrett, "Arduino Microcontroller – Processing for Everyone" Morgan-Claypool Publishers, 515 pages, third edition, 2013, ISBN: 978162705253.

3. I. Verner, S. Waks, and E. Kolberg, "Upgrading Technology Towards the Status of a High School Matriculation Subject: A Case Study," *Journal of Technology Education*, Volume 9, Number 1, Fall 1997.

4. E. Mar, "Mobile Autonomous Robot", MSME Thesis, The Cooper Union for the Advancement of Science and Art, 1998.

5. "Trinity College Fire Fighting Home Robot Contest," Trinity College, Hartford, CT,http://www.trincoll.edu/events/robot, 2001.

6. J. Mendelsohn, "Come On Baby, Unlight My Fire," *IEEE Intelligent Systems Magazine*, pp. 5-6, 2001.

7. MIT Artificial Intelligence Laboratory, www.ai.mit.edu, 2001.

8. D. Pack, G. York, P. Neal, and S. Stefanov, "Constructing a Wall-Follower Robot for a Senior Design Project", Proceedings of the 1996 American Society for Engineering Education Annual Symposium, Washington D.C., June 1996.

9. S.F. Barrett, D. Pack, C.H.G. Wright, S. Stefanov, P. Neal, A. Klayton, "Innovative, Student-Centered Educat-ional Tools for the Computer Engineering Curriculum", *American Society for Engineering Educators 1998 Annual Conference*, Seattle, WA, June 1998.

10. S.F. Barrett, D.J. Pack, G.W.P. York, P.J. Neal, R.D. Fogg, E. Doskocz, S.A. Stefanov, P.C. Neal, C.H.G. Wright,

A.R. Klayton, “Student-centered Educational Tools for the Digital Systems Curriculum,” *Computers in Educational Journal*, American Society for Engineering Educators, Vol. IX, No. 1, Jan-Mar 1999.

11. S. Barrett, J. Anderson, C. Dolan , T.V. Edgar, P. Crips, “Robots in Education: Middle School to Grad School and Beyond!” Computers in Education Journal, Volume 2, Number 3, July-September 2011, pp. 57-69.

12. A. Griffith, S. F. Barrett, D. Pack, “Verilog HDL Controlled Robot For Teaching Complex Systems Design,” Computers in Education Journal, Vol XVIII, No. 1, Jan – Mar 2008, 63-72.

13. Dagu Magician Acrylic Chassis Kit, www.jameco.com

14. Atmel ATmega48A/PA/88A/PA/168A/PA/328/P 8-bit Microcontroller with 4/8/16/32Kbytes In-System Programmable Flash Datasheet, Atmel-8271J-AVR- ATmega-Datasheet 11/2015.

15. Arduino, www.arduino.cc

Biographical Information

Matthew Young Lionel Love is a Senior Undergraduate Student at the University of Wyoming, who graduated in May 2016 with a B.S. in Mechanical Engineering and a B.A. in Spanish. He accepted a position with the Naval Air Warfare Center, China Lake, CA.

Jeffrey R. Anderson received his PhD in electrical and computer engineering at the University of Wyoming with a research emphasis in image processing in 2004. He received his BS and ME from the University of Utah in 1989 and 1992, respectively. He has worked on a closed loop controller for mechanical ventilation of patients with adult respiratory distress syndrome. Additionally, he has worked on a servo lung simulator with related control method that was awarded a U.S. patent. He is currently an associate academic professional at the University of Wyoming in the Electrical and Computer Engineering department.

Dr. Steven F. Barrett, P.E., received the B.S. in Electronic Engineering Technology from the University of Nebraska at Omaha in 1979, the M.E.E.E. from the University of Idaho at Moscow in 1986, and the Ph.D. from the University of Texas at Austin in 1993. He was formally an active duty faculty member and professor at the United States Air Force Academy, Colorado and is now professor of Electrical and Computer Engineering and associate dean for Academic Programs, College of Engineering and Applied Science, University of Wyoming. He is a senior member of IEEE and chief faculty advisor of Tau Beta Pi. His research interests include digital and analog image processing, computer-assisted laser surgery, and embedded control systems. He is a registered professional engineer in Wyoming and Colorado. He authored/co-authored several textbooks on microcontrollers and embedded systems. His book, ”A Little Book on Teaching,” was published by Morgan and Claypool Publishers in 2012. In 2004, Barrett was named ”Wyoming Professor of the Year” by the Carnegie Foundation for Advancement of Teaching and in 2008 was the recipient of the National Society of Professional Engineers (NSPE) Professional Engineers in Higher Education, Engineering Education Excellence Award.

Appendix 1. Parts list

Source	part number	description	qty per robot	cost	extended cost
local manufacture		sensor mounting bracket	1		$ -
www.sparkfun.com	SEN-00242	Infrared proximity sensor - Sharp GP2Y0A21YK	3	$ 13.95	$ 41.85
www.sparkfun.com	SEN-08733	Infrared Sensor Jumper Wire - 3-Pin JST	3	$ 1.50	$ 4.50
www.jameco.com	20601	3.3 x 2.1 solderless breadboard	1	$ 5.95	$ 5.95
www.jameco.com	2210001	Dagu Magician robot (DG007)	1	$ 17.95	$ 17.95
www.mouser. Com	782-A000066	Arduino UNO R3 REV 3	1	$ 24.47	$ 24.47
www.sparkfun.com	DEV-11021				$ -
www.jameco.com	35975	1N4001	10	$ 0.05	$ 0.50
www.jameco.com	32993	TIP120	4	$ 0.39	$ 1.56
www.jameco.com	690742	330 ohm	2	$ 0.09	$ 0.18
www.jameco.com	1952847	9 VDC, 2.0A power supply	1	$ 11.95	$ 11.95
www.jameco.com	51262	5 VDC regulator (7805)	1	$ 0.29	$ 0.29
www.jameco.com	93761	10 uF, 25 VDC capacitor	2	$ 0.12	$ 0.24
www.jameco.com	106551	standoff kit for Arduino	1	$ 2.95	$ 2.95
www.jameco.com	40970	screw	9	$ 0.08	$ 0.72
www.jameco.com	40943	hex nut	9	$ 0.06	$ 0.53
www.jameco.com	106850	washer	9	$ 0.06	$ 0.53
www.jameco.com	28628	transistor, PN2222	5	$ 0.07	$ 0.35
www.jameco.com	690700	220 ohm resistor, 1/4 W	5	$ 0.04	$ 0.20
www.jameco.com	691104	10K ohm resistor, 1/4 W	5	$ 0.04	$ 0.20
www.jameco.com	34825, 333973, 34761	LED (red, yellow, green)	5	$ 0.12	$ 0.60
www.jameco.com	226094	screwdriver set	5	$ 3.95	$ 19.75
www.usplastics.co	57449	small plastic box (6 quart)	1	$ 2.03	$ 2.03
				Total	$ 137.30

Appendix 2. Code listing.

```
//**************************************************************************
//robot_control
//Used with permission from:
//“Arduino Microcontroller: Processing for Everyone!”
// Morgan and Claypool Publishers, 2013
//**************************************************************************

//analog input pins
#define left_IR_sensor 0                //analog pin - left IR sensor
#define center_IR_sensor 1              //analog pin - center IR sensor
#define right_IR_sensor 2               //analog pin - right IR sensor

//digital output pins
//LED indicators - wall detectors
#define wall_left 3                     //digital pin - wall_left
#define wall_center 4                   //digital pin - wall_center
#define wall_right 5                    //digital pin - wall_right

//LED indicators - turn signals
#define left_turn_signal 2              //digital pin - left_turn_signal
#define right_turn_signal 6             //digital pin - right_turn_signal
```

```
//motor outputs
#define left_motor 11                    //digital pin - left_motor
#define right_motor 10                   //digital pin - right_motor

int left_IR_sensor_value;                //declare variable for left IR sensor int
center_IR_sensor_value;                  //declare variable for center IR sensor int
right_IR_sensor_value;                   //declare variable for right IR sensor

void setup()
{
//LED indicators - wall detectors
pinMode(wall_left, OUTPUT);              //configure   pin   1   for   digital   output
pinMode(wall_center, OUTPUT);            //configure   pin   2   for   digital   output
pinMode(wall_right, OUTPUT);             //configure pin 3 for digital output

//LED indicators - turn signals
pinMode(left_turn_signal,OUTPUT);        //configure pin 0 for digital output
pinMode(right_turn_signal,OUTPUT);       //configure pin 4 for digital output

//motor outputs - PWM
pinMode(left_motor, OUTPUT);             //configure pin 11 for digital output
pinMode(right_motor, OUTPUT);            //configure pin 10 for digital output
}

void loop()
{
//read analog output from IR sensors left_IR_sensor_value =
analogRead(left_IR_sensor); center_IR_sensor_value =
analogRead(center_IR_sensor); right_IR_sensor_value =
analogRead(right_IR_sensor);

//robot action table row 0
if((left_IR_sensor_value < 512)&&(center_IR_sensor_value < 512)&&
   (right_IR_sensor_value < 512))
  {
  //wall detection LEDs
  digitalWrite(wall_left, LOW);   //turn LED off
  digitalWrite(wall_center, LOW); //turn LED off
  digitalWrite(wall_right, LOW); //turn LED off

  //motor control
  analogWrite(left_motor, 128);

  //0 (off) to 255 (full speed)
  analogWrite(right_motor, 128);

  //0 (off) to 255 (full speed)
  //turn signals
  digitalWrite(left_turn_signal, LOW); //turn LED off
  digitalWrite(right_turn_signal, LOW); //turn LED off
  delay(500); //delay 500 ms digitalWrite(left_turn_signal,
  LOW); //turn LED off digitalWrite(right_turn_signal, LOW);
  //turn LED off delay(500); //delay 500 ms
  digitalWrite(left_turn_signal, LOW); //turn LED off
  digitalWrite(right_turn_signal, LOW); //turn LED off
  delay(500); //delay 500 ms digitalWrite(left_turn_signal,
  LOW); //turn LED off digitalWrite(right_turn_signal, LOW);
  //turn LED off analogWrite(left_motor, 0); //turn motor off
  analogWrite(right_motor,0); //turn motor off
  }
  :
  :
  //Other maze wall configuration options
 }
//*************************************************************************
```