PROJECT BASED ROBOTICS AT QUEENSBOROUGH COMMUNITY COLLEGE

Danny Mangra, John Buoncora, and Marvin Gayle
Department of Electrical and Computer Engineering Technology
Queensborough Community College
The City University of New York

Abstract

This paper will examine various project based strategies of instruction, which are applied to enhance the students' understanding of concepts in robotics. The topics are dispersed throughout the Electrical and Computer Engineering Technology [ECET] curriculum because there is no course dedicated solely to robotics. A project based approach is used to introduce robotics as opposed to a purely objective driven approach. The projects are typically assigned to students in teams of two, and each team is assigned a common task. The idea is to assign projects that can be accomplished within the allotted class time. The projects also challenge the students' analytical ability and allow the students to work collaboratively towards a meaningful solution.

The three areas that this paper will concentrate on highlight the different aspects of our approach to robotics. The software class focuses on programming fundamentals and the project lab class focuses on hardware concepts. The Robotics Club builds on the fundamentals of the aforementioned classes and integrates the electronic/electrical, mechanical, and programming concepts involved in the analysis, design, and construction of robotic systems. The topics include sensors, actuators, motors, high power driver circuits and microcontrollers. The sensors, which are discussed, include infrared object detectors, light sensors, sound sensors, ultrasonic range finders, position sensors (potentiometers) and speed sensors (analog tachometers/generators). The electrical equivalent circuit of a motor armature is covered along with the relationships between the back electromotive force and motor speed, and the proportionality between motor torque and current. Motor drivers such as H-Bridges are considered along with speed control techniques including pulse width modulation. Microcontrollers, which are programmed in languages including BASIC and C, serve as the robot brain/controllers.

Introduction

"The collaborative nature of the investigation enhances all of these valuable experiences ... as well as promotes a greater appreciation for social responsibility [1]."

The Electrical and Computer Engineering Technology program at Queensborough Community College offers its' students an Associate in Applied Sciences Degree [AAS]. The students either transfer to a four year college or enter the work force at the completion of the program. As such, the program has to be flexible enough to include a variety of topics to cover both contingencies. Typical Robotics programs are offered in a Mechanical Engineering program; however, this paper will focus on the electrical and software projects and activities in robotics offered through our Electrical and Computer Engineering Technology program at Queensborough Community College [QCC].

ECET students take courses in mathematics, physics, electric circuits, electronic devices, computer circuits, programming, and general education areas. The students learn college level algebra, trigonometry, geometry, and calculus in their math courses. Algebra and trigonometry are applied to physical problems in the electrical circuits, electronic devices, physics, and programming courses. The students also use complex numbers to solve problems in AC circuit analysis. Although the students take a course in calculus, which includes derivatives and integrals, they are not typically asked to apply calculus in their electronics and physics courses, which are mostly based on algebra, trigonometry, and geometry. However, calculus is used in the derivations of certain equations in electronics courses, such as the equation for the small signal dynamic resistance of a diode.

In the technology program, robotics concepts are introduced at appropriate times throughout our curriculum. These topics are well received and have stimulated significant interest among our students. This appeal helps to support our thriving robotics club, which also provides hands on experience for our students. This paper will examine three project based strategies of instruction, which are applied to enhance the students' understanding of concepts in robotics.

Programming Projects

The programming class uses the Parallax Boe-Bot Robot as a platform to introduce concepts in robotic control from a software point of view. The introductory Programming class uses Microsoft Visual Basic to introduce programming concepts in Visual Basic. The course covers programming concepts and topics. Once the students have mastered these topics, they begin to use the Parallax Boe-Bot Robot Kit and some knowledge of Visual Basic concepts to solve a challenge in a robotic application.

Using the Parallax Boe-Bot Robot Kit and hardware platform has many advantages. It is small enough and the cost is relatively low, so that each student can get one if necessary. Once the students have some background in programming with Visual Basic, the Boe-Bot Robot is fairly easy to program using the PBASIC programming language which is very similar to Visual Basic. The hardware used is the Parallax Board of Education, which houses a BASIC Stamp microcontroller. The processor used is relatively low speed, the specification are shown below: [2]

- **Processor Speed: 4 MHz**
- **Program Execution Speed: ~2,000 instructions/sec.**
- **RAM Size: 16 Bytes (2 I/O, 14 Variable)**
- **EEPROM (Program) Size: 256 Bytes ~80 instructions**
- **Number of I/O Pins: 8**
- **Current Draw @ 5 VDC: 1mA Run, 25 µA Sleep**
- **Source/Sink Current per I/O: 20 mA / 25 mA**
- **Source/Sink Current per unit: 40 mA/ 50 mA**

The hardware package also includes the motor, and sensors that can detect contact, infrared and visible light. The interface connection to the computers is via the serial port even though USB ports are available on newer models. The platform can also support basic wireless communication between two points with additional hardware. From a mechanical perspective the motor and gear interface is simple enough so that the projects can be tackled without the need for an in-depth understanding of gear mechanics.

One of the objectives of the class is to introduce basic robotics software control concepts and this is done by assigning the students a project to be implemented using the Parallax Boe-Bot Robot. Students learn how the servo motors are controlled using the Basic Stamp microcontroller. The discussion then transitions to microcontroller programming, and how the microcontroller handles inputs and outputs. The students are then assigned one of two projects.

The students are expected to understand the programming concepts that they were exposed to earlier. Since PBASIC is similar to Visual Basic, a very brief discussion of syntax, loops, counters and other nuances of PBASIC is presented and an introductory assignment is then given, which focuses on the microcontroller programming aspects. To illustrate concepts in robotics control and object detection we assign students two projects to solve.

Project one requires that the students control the direction of the robots movement by controlling the pulse and duration of the directionally controlled servo motors. There is one motor on each side of the robot. The packaged documentation from Parallax has included some sample code that will turn the Boe-Bot. The student will have access to this and they have a chance to input and investigate what each line of code does. This is by no means a herculean task since the package has an appropriate level of comments that will assist in their investigation. The challenge comes from them both taking the code and using it as an example to accomplish a similar task of turning the robot 90, 180, 270 or 360 degrees. The sample code included in the Parallax package is shown below

```
' Robotics! v1.5,            Program Listing 2.3:[3]
' {$Stamp bs2}                          ' Stamp Directive.

'--------|          Declarations              |-----------
pulse_count   VAR   Word  'A word size loop counter.

'--------|          Initialization            |----------
OUTPUT 2                                ' Set P2 to output.
FREQOUT 2, 2000, 3000                   '3 kHz signal for 2 s.
LOW 12                                  'P12 to output-low.
LOW 13                                  'P13 to output-low.

'---------|         Main Routine              |------------
main:                                   ' Main routine

Left_Turn:                              'Left turn routine.
FOR pulse_count = 1 TO 35               ' 35 left rotate pulse.
PULSOUT 12, 500                       ' 1.0 ms pulse to R servo.
PULSOUT 13, 500                       ' 1.0 ms pulse to L servo.
PAUSE 20                                ' Pause for 20 ms.
NEXT
PAUSE 500                               ' Pause for 0.5 s.

Right_Turn:                             'Right turn routine.
FOR pulse_count = 1 TO 35               ' 35 left rotate pulse.
PULSOUT 12, 1000                      ' 2.0 ms pulse to R servo.
PULSOUT 13, 1000                      ' 2.0 ms pulse to L servo.
PAUSE 20                                ' Pause for 20 ms.
NEXT
PAUSE 500                               ' Pause for 0.5 ms.
STOP                                    ' Stop until reset.
```

Project two requires that the students use the contact sensor on the Parallax Boe-Bot to “detect” an object. The robot has the ability to detect its' proximity to an approaching object and can use that information to avoid contact using the infrared sensors, as a result this project could easily be modified to include IR detection. For our project however we use the contact sensor to detect an object by bumping into it. Contact with an object is used to open or close a switch, which is used as an indicator telling the robot on which side the contact occurs on. The Boe-Bot then uses the knowledge of whether the obstacle is on the left or the right side of the robot and generates an appropriate interrupt. The Boe-Bot has to use that contact and pivot or turn to avoid the contact. In the end, the two programming projects are combined so that the contact initiates the turning of the Boe-Bot and the two projects working together is what avoids the full contact with the obstacle.

The Hardware Project Laboratory

The approach to Robotics in the project laboratory class at QCC focuses on digital and analog electronic circuit theory, and some mechanical concepts. The learning objective is for students to understand electronics as related to applications in robotics. Students also get practical experience in troubleshooting electronic circuits and motor controllers, using effective instrumentation and measurement techniques. The prerequisite for this course is the study of microprocessor systems, which include memory, memory interfacing and I/O systems. The students also need to know the basic theory and operation of electronic devices including semiconductor switching diodes, zener diodes, and transistors. The students use the schematic diagram to troubleshoot the digital and analog electronic systems of the robot in order to solve any functional problems.

The unique feature of the robot used in the project lab class is that it consists of discrete subsystems, which include power supplies, clock oscillators, memory, digital logic circuits, transistor drivers, and motors. The students build and test each of these subsystems in a sequential manner, and then integrate them into a functional robot.

The following discussion will focus on some of the specific subsystems used in the Graymark 603A Digital Programmable Robot project kit [4]. The design incorporates two DC voltage sources. The 9V source powers the digital circuitry and two 1.5 V batteries in series provide power for the DC motors The use of a separate power supply prevents electrical noise and transients from interfering with the operation of the digital circuits. The voltage regulator contains a zener diode and series pass transistor, which converts the 9V input to a 5V output that powers the robot's digital systems. The voltage regulator circuit is shown in Figure 1. The zener diode provides a constant 5.6 V at the base of the transistor. Students calculate the output voltage of the regulator circuit by taking the zener diode and transistor base to emitter voltage into account. The students also analyze the relationships between the load, BJT, and zener diode currents.

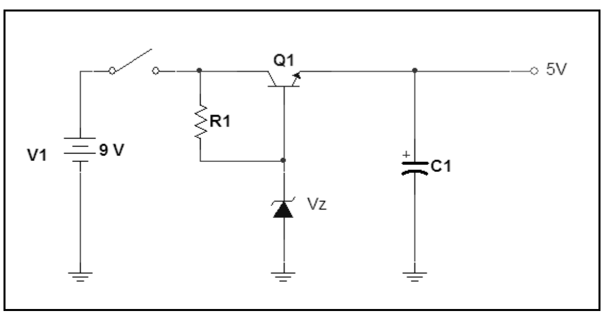

Figure 1. Voltage Regulator Circuit.

The students analyze the operation of the OR Gate of Figure 2, which is based on discrete components. The transistor Q1 behaves like a switch that is "on" when 5 volts is applied to either or both of the inputs. The LED will illuminate if there is an over flow condition into memory or if the program instructs it to do so.

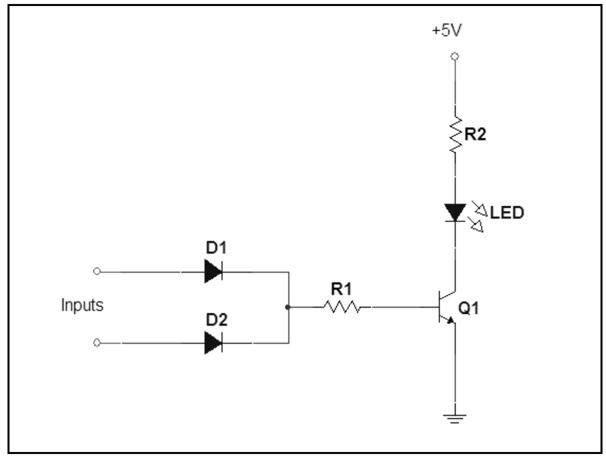

Figure 2. Two Input OR Gate.

The Robot is clocked by the astable multivibrator circuit of Figure 3, which consists of three inverters of the CMOS 4069 IC. The astable multivibrator circuit produces a periodic square wave with an output frequency, which is controlled by potentiometer R_3. The time constant is calculated by using:

$$\tau = R_T C_1$$

Where:

τ = Time Constant

$R_T = R_1 + R_2 + R_3$.

The frequency is inversely proportional to the time constant. A second oscillator circuit generates a 2.5 kHz audio tone, which drives the robot's speaker, under the control of the program. The students measure the frequency using the oscilloscope, and they compare the measured and calculated frequency values.

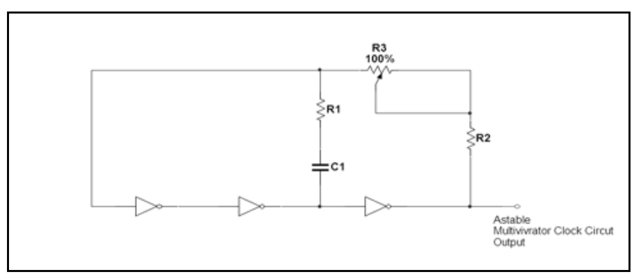

Figure 3. The Astable Multivibrator Clock Circuit.

The robot contains one motor on each side, which are driven by BJTs. Figure 4 shows the driver circuit for one of the motors. A high is applied to resistor R1 when the motor run command is executed in programming mode, which drives transistor Q1 into saturation. Therefore, the transistor behaves approximately as a short circuit from collector to emitter, which results in the application of 3 V to the motor.

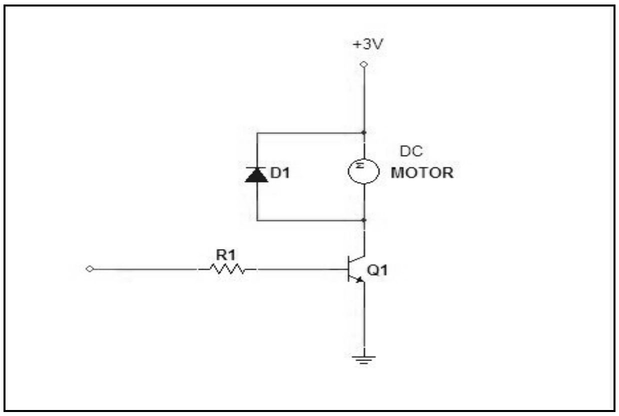

Figure 4. Motor Driver Circuit.

The students build the robot by soldering components to a printed circuit board, which is mounted to a mechanical chassis assembly that consists of motors and gears. The motors and gears used in the mechanical assembly of the robot provide the steering function. The students learn that gears can be used to proportionally reduce rotational speed, while increasing torque in accordance with the conservation of energy principle.

To program this robot, instructions are fed through a keypad that connects directly to the memory load sequence circuit, which increments each command into memory. There are a maximum of 256 instructions that can be programmed into the 256 nibble memory. There are five command types that can be programmed into the robot. These commands are Forward, Right and Left, LED, and Speaker. The instructions for the robot are programmed into memory in desired order such that the robot can navigate through its' environment. The students control the speed by adjusting the oscillator rate. The robot exits the programmable mode and enters the execute mode upon removal of the keypad. The instructions are read from memory in the order in which they were programmed and are repeatedly executed in a continuous loop. The students interrogate memory locations, to identify the contents, which determine the appropriate behavior. The commands are stored in the memory circuit. Data fetch from the input lines are stored using binary numbers assigned by the address counter. The memory IC can be selected by Chip Select $\overline{CS}$ being set Low. Once the OD (Output Disable) pin is low, data will appear at the output data line.

This method is fully utilized in a robot that performs the same task repeatedly. Students gain a great deal of experience in analog, digital and mechanical theory by building this robot. They have the opportunity to observe how all of these systems work individually and coincide as a whole.

Projects in the Robotics Club

The learning objectives are to integrate the use of mathematics, physics, electronics, and computer programming in a robotic design project, which involves the robotics club students. The concepts covered in the Robotics Club build on the topics discussed in the software and project lab classes. The goal is to apply these disciplines to real world problems. The principle of a closed loop control system with a PID (Proportional plus Integral plus Derivative) controller is presented to the students. The PID control equation would be presented using a formal application of calculus in the engineering curriculum of a four year college. However, the PID control equation is presented at Queensborough Community College in an approximate form using algebra:

$$\mathbf{PID_{OUTPUT} = K_P\, E + K_I\, \Sigma\, (E\, \Delta t) + K_D\, (\Delta E / \Delta t)}$$

Where:
E represents the error signal (difference between setpoint and process variable):

$\mathbf{PID_{OUTPUT}}$ represents the output of the PID controller, and $\mathbf{K_P}$, $\mathbf{K_I}$, and $\mathbf{K_D}$ represent the Proportional, Integral, and Derivative Gains respectively.

The fact that the PID control equation is presented without the use of formal calculus is not an issue because our intention is to implement PID control (approximately) in the software used in a line following robot. Therefore, the PID control equation provided above is implemented in a C language program that is running in the robot's microcontroller. The Gain parameters are determined experimentally by using a trial and error approach, instead of an analytical approach as used in the engineering curriculum of a four year college. The C program, which illustrates the use of PID control, is provided in the Pololu 3pi Robot User's Guide by Pololu Corporation [5].

The students learn to program the Parallax Scribbler Robot using the Scribbler PBASIC Programming Guide by Parallax [6]. The students then build robots, which are based on differentially steered drive systems. The analysis of the differentially steered drive system requires an application of algebra, geometry, and physics. The students derive the equation relating the radius of the turn, which the robot is negotiating, to the rotational (angular) speeds of the (two) wheels and the wheelbase width of the robot (width between the two wheels). The equation is then used to calculate the radius of the turn based on the numerical values of the rotational speeds of the wheels and the wheelbase width of the robot. The microcontroller outputs a rectangular waveform to the driver circuits for each motor, and the duty cycle of each of the waveforms is set by the parameters used in the microcontroller's program. The duty cycle has different values for each of the motors when the robot is turning. The students learn that the microcontroller outputs must not be connected directly to the motors because the current rating of the microcontroller output pins is **in**sufficient to drive the motors and the operating voltage of the motors is different from the microcontroller output voltage levels (in general). An H-Bridge is used to drive the motor, which allows the motor to run in either the forward or reverse direction, or come to a stop by coasting or braking. An integrated circuit H-Bridge is typically used in the robot to conserve space, as opposed to using an H-Bridge based on discrete transistors. However, the internal transistor circuit is analyzed theoretically and simulated using circuit simulation software. The robotics club students are using the disciplines of science, technology, engineering, and mathematics to analyze and implement the mechanics, electronics and software for a differentially steered robotic drive system. All technology students belong to the STEM (Science, Technology, Engineering, and Mathematics) Academy at Queensborough Community College, and the study of robotics involves all of the STEM disciplines.

One example of an application of physics applied to robotics is the equation, which states Torque = Force × radius (assuming a 90 degree angle between the force and radius vectors). Students solve this equation for Force, in order to obtain the propulsion Force caused by a robot's motor/gear head assembly at the wheel to ground contact point, as a function of the Torque at the output of the motor/gear head assembly and the radius of the drive wheel. The students learn that most of the motors designed for applications such as robotics run at a "high" rotational speed and produce "low" toque at the motor shaft, which is coupled to a gear head, that produces a lower rotational output speed at a higher output torque. Students learn the principle of conservation of energy and the equation for power in terms of torque and angular velocity (rotational "speed") in their physics courses. Robotic drive system analysis provides the opportunity to apply these concepts to practical real world applications, where the students can "see" the conservation of energy principle in action.

Technology students study the characteristics of permanent magnet motors. Analyzing a permanent magnet motor requires an application of Kirchhoff's Laws, Ohm's Law, Faraday's Law, and the concept of the magnetic force exerted on a current carrying coil of wire. Each of these concepts is studied in electrical circuit courses, electronics courses, and physics courses; however, the analysis of a motor used in a robot allows all of the concepts to be applied simultaneously to a problem. Once again, the equations in technology programs are presented using algebra, as opposed to the use of calculus in a four year college program. For example, the (average) angular speed is expressed as the change in angle divided by the change in time, instead of discussing the instantaneous angular speed, which would be expressed as the derivative of the angle with respect to time. The students are provided with the opportunity to use unit analysis and perform unit conversions, such as converting the units of angular speed between radians/second and revolutions per minute. Of course, the inductance of the armature of a motor is discussed; however, the presentation uses algebra, as opposed to the use of calculus. Furthermore, the armature circuit is considered under steady state DC conditions where the voltage term across the equivalent armature inductance becomes zero because the current is constant. Four year college students actually write the differential equation for the motor armature circuit and solve the equations to obtain the complete response, which applies in the transient and steady-state time intervals.

The programming of a robot in any language presents unique challenges because the microcontroller program is frequently monitoring inputs and updating outputs. Consider the microcontroller program for a robot, which is executing random wandering with obstruction avoidance behavior. The microcontroller/program must monitor the sensors, process information by executing computation and decision making statements, and then update outputs. Monitoring the sensors and updating the outputs requires the consideration of real time constraints, which are unique to embedded systems programming and robotics. The related educational objectives include the consideration of tradeoffs between hardware and software implementations. For example, a simple infrared LED emitter and series resistor, along with an infrared detector can be used for object/obstruction detection. In this case, the microcontroller must output a signal at the appropriate frequency to the infrared LED and then monitor the detector. However, another solution is to use a packaged digital distance sensor, which contains the LED, detector, and support circuitry. The advantages of using the packaged digital distance sensor are that it is no longer necessary to use a microcontroller output pin to pulse the infrared LED and the lines of code (and processor time) required to generate the signal are eliminated. However, the advantage of using the software approach with simpler discrete components is a savings in hardware complexity and cost. The students learn the advantages and disadvantages of both hardware and software implementations in a practical real world robotics design project. Furthermore, the students are engaged and they can appreciate the tradeoffs involved in taking different approaches to implement a system. Students also learn that the time used by the microcontroller is a valuable resource and the results can be disastrous if the time is not allocated appropriately to critical task such as monitoring sensors. For example, the robot will crash into an object if the microcontroller is busy performing a task for too long of a period of time, which excludes monitoring the object detection sensors. Robotics applications provide the opportunity to observe such problems first hand. This is in contrast to pure processing program applications, which do not involve computer control applications, where an inefficient program may still be successful at solving a problem, but will merely require an extended period of time.

Pulse width modulation is used to control the speed of permanent magnet DC motors or pulse code modulation is used to control the speed of servo motors. The pulse width modulated output is typically implemented by a microcontroller program which contains a loop with a command that generates a pulse for a certain time duration followed by a command that produces a certain time delay. The programs used in robotics applications must typically perform other tasks within the loop. However, if these tasks require too much program run time, the duty cycle of the output will decrease and the motor will typically run too slow as compared to the desired running speed. In fact, the motor may actually stop running. For example, the students and the instructor were troubleshooting a program by adding a command which sends data back to a computer, when it is connected to the microcontroller. This troubleshooting method worked fine in the static case where the robot is immobile and objects can be moved in front of the sensors to test the microcontroller/program response. However, after disconnecting the cable and allowing the robot to operate freely in a random wandering with obstruction detection mode, the students and the instructor observed that the robot was moving extremely slowly for no apparent reason. The experimenters then realized that the robot was trying to send data back to the computer, which was disconnected. This caused a long time delay before a data communication timeout, which significantly decreased the duty cycle of the microcontroller output pulse to the motor driver circuit, resulting in very low speed operation. The software implementation of pulse width or pulse code modulation has serious drawbacks. These drawbacks suggest the possible use of external motor controller circuitry or the use of microcontroller counters or timers with interrupts, which have been mentioned, but not yet implemented in the robots, which were designed and built by the robotics club participants. The robotics club students constructed a two wheel drive autonomous robot, which uses a differentially steered drive system to turn, and infrared sensors to detect obstructions. The microcontroller program for a robot that is executing random wandering with obstruction avoidance behavior is shown below.

This program was written by the robotics club students in PBASIC for this robot, which uses a BASIC Stamp Microcontroller as the brain, using commands contained in the BASIC Stamp Manual [7].

Random Wandering with Obstruction Avoidance Program

```
' {$STAMP BS2}
' {$PBASIC 2.5}

'--------|Variable and Pin Declarations     |-----------

counter VAR Byte
light VAR Word
Level PIN 3
INPUT 1
INPUT 2
INPUT 3
INPUT 5

'--------|            Endless Loop               |-----------
DO
light = 0

' Wait for Light pulse before wandering
DO WHILE ((light < 22) AND (IN5 = 1))
  COUNT Level, 1, light
LOOP
HIGH 4
PAUSE 1500
light = 0

'--------|Conditional Loop for wandering |-----------

DO WHILE ((light < 22) AND (IN5 = 1))
' Check Sensors
  IF IN1 = 0 THEN GOSUB Straight
  IF IN1 = 1 THEN PAUSE 10
  IF IN1 = 1 THEN
    GOSUB Halt
    GOSUB Back
    GOSUB Halt
    GOSUB Spin
    GOSUB Halt
  ENDIF
  COUNT Level, 1, light
  IF ((light >= 22) OR (IN5 = 0)) THEN LOW 4
LOOP
PAUSE 2000
LOOP
END
```

```
'--------|           Directional Subroutine    |-----------
Straight:
  PULSOUT 6, 790
  PULSOUT 7, 700
  PAUSE 20
  RETURN
Halt:
  FOR counter = 1 TO 25
    PULSOUT 6, 750
    PULSOUT 7, 750
    PAUSE 20
    NEXT
  RETURN
Back:
  FOR counter = 1 TO 50
    PULSOUT 6, 710
    PULSOUT 7, 800
    PAUSE 20
    NEXT
  RETURN
Spin:
   FOR counter = 1 TO 68
    PULSOUT 6, 710
    PULSOUT 7, 710
    PAUSE 20
    NEXT
  RETURN
```

Conclusion

In conclusion, the ECET program uses robotic projects of various types to engage the students in the STEM disciplines. The educational value is to enhance the students' knowledge and skill sets in all the areas of robotics. These skill sets include the application of mathematics, physics, electronics, and programming to project based robotics challenges.

References

1. Carolyn A Scott. "Project-based science: Reflections of a middle school teacher. Elementary School Journal", Volume 95, Number 1, September 1994.

2. Parallax homepage General Product Information Page http://www.parallax.com/ProductInfo/tabid/110/Default.aspx_Accessed [March 2010].

3. Parallax Sample Code package Program Listing 2.3

4. The Graymark 603A Digital Programmable Robot http://www.graymarkint.com/newweb/newweb/Robot%20Kits%20and%20Books_page20.htm Accessed [March 2010].

5. Pololu 3pi Robot User's Guide by Pololu Corporation http://www.pololu.com/docs/pdf/0J21/3pi.pdf Accessed [March 2010].

6. Scribbler PBASIC Programming Guide http://www.parallax.com/Portals/0/Downloads/docs/prod/robo/scribbler/SPPG-Writing Programs-v1.2.pdf Accessed [March 2010].

7. BASIC Stamp Syntax and Reference Manual 2.2 http://www.parallax.com/dl/docs/prod/stamps/web-BSM-v2.2.pdf Accessed [March 2010].

Biographical Information

Danny Mangra is an Assistant Professor at Queensborough Community College. He received the B. E. (E.E.) degree in Electrical Engineering and the M.S.E.E. from Polytechnic University of New York

John Buoncora received the B. S. degree in Electrical Engineering and the M. S. degree in Electrical Engineering from Polytechnic University of New York. He worked in industry as a test equipment engineer and is now an Instructor at Queensborough Community College.

Marvin Gayle received the B.E. [E.E.]and M.E. [E.E.] degree from The Grove School of Engineering at The City College of New York [C.U.NY] and is a Licensed Professional Engineer [P.E.] registered in New York State.