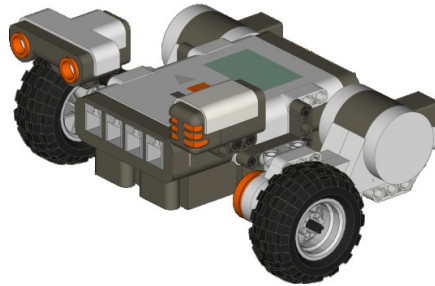


# Robotics in Education eJournal

Volume 3 – July 2010

Compiled by Damien Kee – Domabotics



Robotics in education is fast becoming a popular way to engage students in the fundamental STEM concepts (Science, Technology, Engineering and Math). This eJournal brings together articles from teachers from all over the world who are using Robotics in different and exciting ways. Please join us on the *Robotic in Education* mailing list, and let us know how you have been using robotics in your classroom.

## Contents:

- **Digital Design with LEGO Robotics** - Wayne Burnett (Singapore)
- **Storytelling and scenario building as an enforcement in LEGO introductory activities** - Roberto Catanuto (Italy)
- **Robotics in the Greenhouse** - Eduardo Ventura M. (Dominican Republic)
- **Effective education with limited programming knowledge** - Damien Kee (Australia)
- **NXT vs. RCX vs. Pico Playing to their strengths and being a wise steward of resources** - Laura Jones (USA)
- **Teaching the Path of Regular Polygons – An Approach to Introductory Programming** - Craig Shelden (USA)

---

Damien Kee – *Robotics in Education* mailing list coordinator and moderator.



Dr Damien Kee holds a PhD in Robotics and Bachelor of Electrical Engineering from the University of Queensland. Damien has been heavily involved with the RoboCup Junior competition since 2001, currently serving as Chairman of RoboCup Junior Australia and Technical Chair : RoboCup Junior International – Rescue League

He has been running robotics workshops in Queensland, Australia and Internationally for students and teachers since 2002 and has worked with over 1000 teachers and countless more students.

# Digital Design with Lego Robotics

## Wayne Burnett (Singapore)

With all the demands on school time, the class called “technology” often gets the short end of the stick. Who can really deny that language arts and math deserve priority? But, when you want to get the most of the robotics unit and the expensive Lego kits, time is critical. Students need time to design, build, programme, test, evaluate and re-design, re-build and re-programme.

This was the challenge I faced as I started teaching robotics at the middle school level. Prior to this, robotics in my schools were offered as after school activities. There was less time pressure to achieve a set of benchmarks and assessment progress. But, once in middle school and with not a lot of lessons per week, I struggled with the question of how to give students more access to the kits. Of course, sending home a Mindstorms kit was not an option. True, they could do some of the written work at home, documenting their research or design plans. I have had students take digital photographs of their robots, sometimes to document the step-by-step creation process, other times to show design options they were considering. However, photographs cannot give the 3D perspective that fully reflects what students have created.

Finally, on a robotics discussion groups, a member mentioned MLCad, a computer assisted design (CAD) programme designed for Lego. In conducting research, I also came across LeoCAD, Bricksmith and Lego Digital Designer (and there are others). What follows are some comments on my experience using them, primarily MLCad.

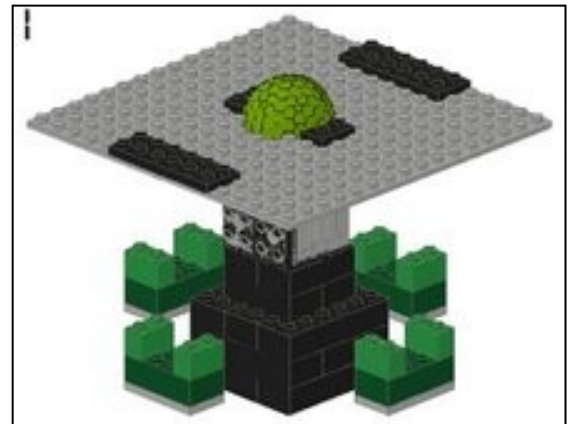
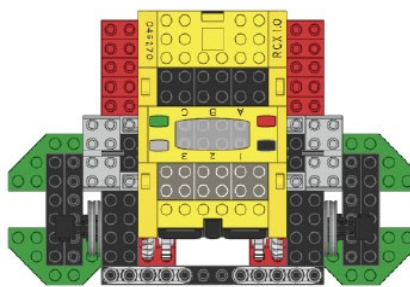
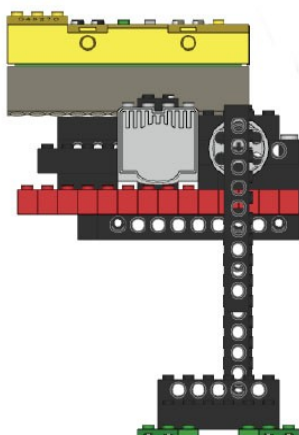


Figure 1: MLCad image of an amusement park ride design option created by a grade 8 student



Figure 2: Photo of the amusement park ride as built

*These three images explore a different form of locomotion using Bricksmith.*



## Using the software in educational robotics

I use the software in two ways and I am still evaluating what works best. One year, I introduced the software before the hands-on robotics kits but found students had difficulty relating the digital Lego pieces to the real ones, particularly in terms of relative size. The following year, I worked with the Mindstorms kits first and then introduced the software. This probably works better, but still raises a question for me: Should students be designing first, building second? Is there a **design→build→programme** process in educational robotics, or is it more a **build, re-build and programme** strategy? Does it matter?

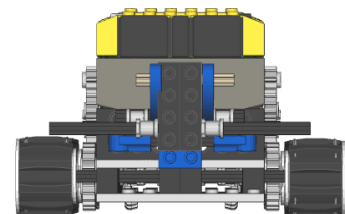
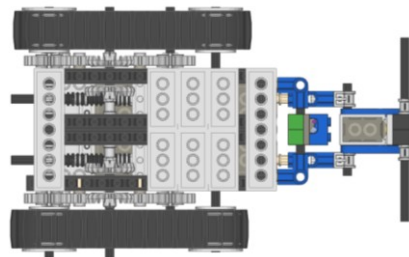
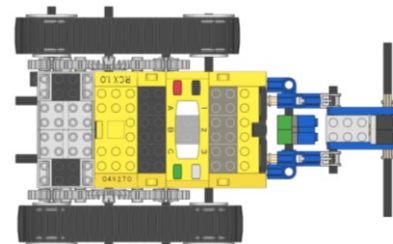
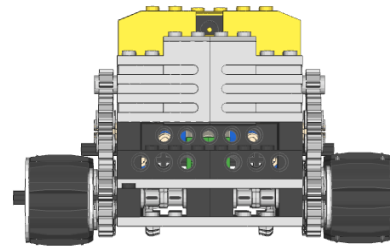
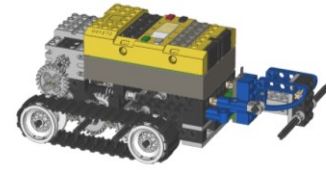
Even if not used to design robots, I have found the software particularly useful in documenting what they have done. It is easier to take digital pictures of a robot than to create (or re-create) it using CAD software but photos do not provide all the information that you want. Not only does the software give you a 3D view (you can get different perspectives – top, bottom, left, right, front and back), a related viewer (LDView) is available that allows you to rotate the creation in various directions on a computer monitor. In addition, MLCad, at least, generates a parts list and building instructions. All this helps if students are required to submit documentation of what was considered, which option was selected and designed and the pieces necessary to construct it.

Moreover, I see this as an opportunity to introduce some CAD into the middle school. We are planning to enhance our design technology programme and this might include other examples of computer assisted design and manufacture (CAD/CAM). These robotics software options are a good way to introduce students to this area.

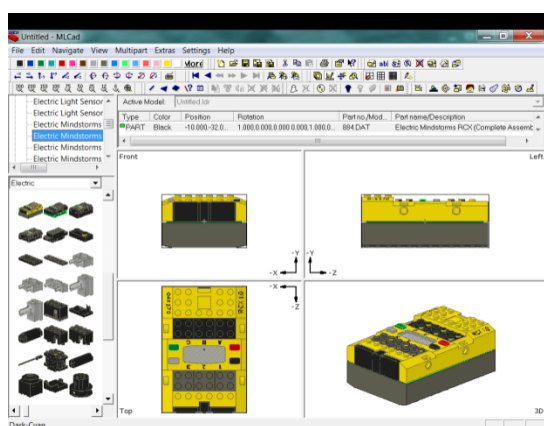
### Software Options

All of the options I have looked at (MLCad, LeoCAD, Bricksmith and LDD) are free software packages that work not just for Lego robotics pieces but for other Lego pieces as well. Bricksmith is specific for Macs (not an option at my school, though some students used it at home) while LDD supports Mindstorms NXT only. At the time, I was limited to the yellow RCX bricks, so LDD was not an option (but will be as of this August!). This left me with MLCad and LeoCAD. Both of these (along with Bricksmith) are based on older software called LDraw. LDraw is where the digital specifications for each piece are developed and maintained and MLCad and LeoCAD are ways to use the LDraw Library using a graphics user interface (GUI). In other words, MLCad and LeoCAD let you use a mouse.

*Figures 3, 4, 5 & 6: Series of images depicting a potential design similar to Wall-E made in MLCad designed by a student*



I tried both and there is a danger that the first one I tried became the benchmark of comparison. MLCad is sophisticated and probably more difficult to use than LeoCAD but still it seemed to be a better option for my students. It seems to have more power and flexibility.



would be easier.

Figure 8: Screenshot of MLCad

All the software has to be installed and can be done without having to register or sign up (no personal information). I have found MLCad can be a bit more difficult to install but essentially all students were able to do this at home in the end. Volunteers design the pieces and some pieces are not yet officially approved but can still be downloaded. The NXT pieces are available in MLCad and should be for LeoCAD as well.

## Simulated Programming

In addition to using CAD software for design and building documentation, I used a subscription web site to introduce programming. Robolabonline is a fairly inexpensive service that allows students to programme in the four Pilot levels. Unfortunately, it supports neither the Inventor level nor NXT. As Lego is phasing out support for Robolab, Robolabonline might also be phased out. However, I have an email from Lego suggesting that it will not be “taken down” soon.

Robolabonline has several built in challenges with visual examples of what the car should do. Then, students create their own programme and watch to see if they were successful in meeting the challenge. I found that my students found this to be a fun activity, certainly useable at the elementary school level. It was particularly helpful in my elementary school after school activities where I could get my students to focus a bit more on programming (they much preferred to build). My middle school students went through it quickly and it was a pity that the Inventor level was not available. It is unclear whether Lego will implement or support the development of an NXT version.



Figure 7: Screenshot of Robolabonline showing the See Example view. Tabs at the top allow students to select the Design programme view and the View Results view.

## Going Forward

I am interested in how using the software improves learning. Do students make better robots when they have to place digital piece after digital piece together, thus developing a better idea of what works, what fits and what options exist? Are CAD skills in the context of robotics transferrable to other CAD contexts? Once students have had experience with a kit and related software, is it possible (or advisable) to have students design first and build second? They might be directed to generate three designs and evaluate which one seems to be the best before they build it. At least for large projects like planes, this probably better reflects what happens in the real world.

As I transition to NXT, I will be interested in any different experience students have as we move to LDD. The other packages can also use NXT pieces and students will always have their choice at home, but I will teach just one, LDD, at school.

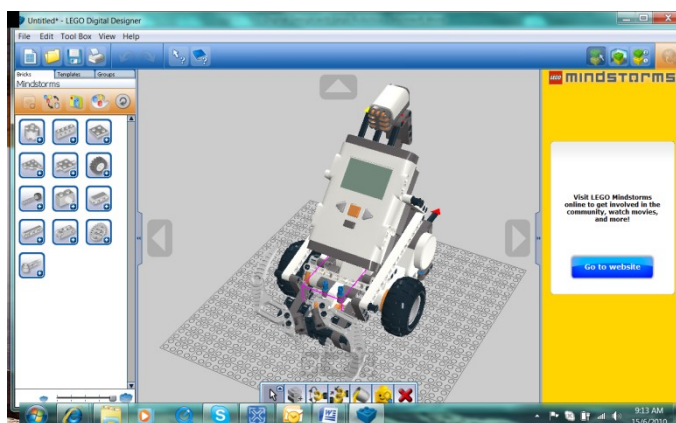


Figure 9: Screenshot of Lego Digital Designer with a pre-loaded NXT model

## Software Download Sites

MLCad: <http://www.lm-software.com/mlcad/>

LeoCAD: <http://www.leocad.org/>

Bricksmith: <http://bricksmith.sourceforge.net/>

Lego Digital Designer: <http://ldd.lego.com/>

LDraw (including information on parts and related software): <http://www.ldraw.org/>

Robolabonline: <http://www.robolabonline.com/>

# **Storytelling and scenario building as an enforcement in**

## **LEGO introductory activities**

Roberto Catanuto, Ph. D. (Italy)

Robotics Projects Coordinator

Primary and Secondary Schools

### **1. Introduction**

After a long run, primary schools were introduced to robotics in Catania (Italy) school district from February, 2010. They were already highly interested in this educational field, also thanks to the annual robotics competition and exhibition Minirobot ([www.minirobotics.org](http://www.minirobotics.org)), a joint effort largely supported by DIEES Engineering Department of the local university.

This paper addresses two very useful engagement strategies for kids, aged 7-10: storytelling and scenario building for their robots. The following remarks regard two different primary schools.

### **2. Planning the robotics activity.**

A first problem arose when the author had to plan the robotics course for the two primary schools: neither the students nor the teachers have ever had previous experience with robotics – flavoured activities. On one hand (i.e. the students) this was not a big issue, since they were highly attracted by the opportunity, hence they were eager of putting their hands on the Lego sets. On the other hand (i.e. the teachers) this was a big issue, since they felt not comfortable at collaborating on a project whose core topic was almost totally obscure for them. One of them even rejected any invitation to learn something regarding robotics and was going to set herself apart. Hence the author and the school staff had to face clearly this problem, well before the course started.

The simple proposed solution was to integrate robotics related activities (like building models and programming them) to other activities more traditional in those school settings: painting, building scenarios with rough paper, and especially storytelling.

The rest of this paper addresses and shows the outcomes from the two schools.

### **3. First primary school: a little group**

The first school to collaborate with primary kids was already a partner of the 2009 project, and they proposed to enlarge the audience of students, inviting also younger kids. The group formed had 7 kids, but they were really too many for one kit only. That's why we decided to split them into two smaller groups: the first one decided to work with Lego WeDo and the second one preferred the traditional NXT set.

The former was invited to choose one of the model proposed by Lego instructional materials and the kids decided to start working on the “hungry alligator”. They easily built it and went on to program its movements. This part of the work was divided into two: simple programming with no sensors and programming a more intelligent behavior, using the motion sensor.

## Storytelling:

One of the female children of the group decided to write a nice story centered on the presence of the hungry alligator and of another NXT robot which was added later to satisfy and enrich the narration, a big grab.

The story is as follows<sup>1</sup>:

### The Hungry alligator and the superGrab

*An extraterrestrial ship ran into troubles while travelling through the space. It carried a little alien, called Sevi. He was trying desperately to keep himself on a good track, wandering and wandering, but finally he reached the Earth and landed in Egypt, in a very small town near Nile river.*

*He got out from the ship and soon asked for help. But all the people eventually meeting him were really scared and ran away, shouting. Hence, he decided to call telepathically his alien mates for help. They were guided by Sevi's mother, and soon moved to search him in the entire universe.*

*Unfortunately, they did not land just near Sevi, so they had to look for him in the surroundings.*

*In the meantime, an alligator was resting near the Nile river and he felt a strange and new presence over there.*

*Also, Sevi was getting nearer and nearer, still not aware of the alligator, hidden amongst Nile's papyrus. All of a sudden, the alligator came out and tried to catch Sevi. But then the aliens arrived and succeeded to call for the attention of the alligator, who hogged some of the aliens all the same !*

*The inhabitants of the small village were very scared by the scene, thinking about the possibility of ending up like the aliens and so they called some scientists to help them with the superGrab !*

*The supergrab was the robot built by the scientists to protect all the people in the world. The robot started a very fierce battle with the alligator, that was killed after a long war.*

*That's how the inhabitants of the village and the survivors of the Navi people were saved by the supergrab. The Navi finally moved back to their Esperia planet, and Sevi lived happy again with his friends.*

*Thank you for your attention !*

## Scenario building:

The construction of the robot and its mate had to be completed and enriched with the scenario surrounding their interaction. Hence, the kids decided to create a nice colored world map on a large paper sheet, where all the five continents were depicted (Figure 1). Moreover, they built also the aliens and their extraterrestrial ships. The superGrab moved over the sea and, using a color sensor, it eventually recognized the red color of Africa. After that, it tried to catch the alligator using a simple arm built on top of itself.

The importance of this second activity is prominent, since it helps the students to create a meaningful and relevant goal for building the robot. The alligator and its mate are not isolated constructions, but only to learn something more or less abstract and more or less engaging. They are strictly linked with a surrounding entire world, created from scratch by kids, where they can mirror themselves and their very imaginative thoughts.

This empowers their motivation for learning to build and program better and smarter robots, whose goal is clear for the students.

---

<sup>1</sup>The story is translated from Italian to English.



#### 4. Second primary school: a larger group

The second school provided a larger group of 80 children, divided in smaller teams. Each team was given a Lego Wedo set. It was the first time for this school to be engaged in robotics activities, hence there was a greater expectation.

At the beginning, the students were asked to imagine freely the robot they would have liked to build, without taking into account the actual limitations of the kit. They dreamed of a whole set of fantastic robots, mainly inspired by what they were studying at school: the prehistoric era and their animals. Hence they planned to build something like ancient dinosaurs with fantastic abilities. After two runs of construction – programming – test of the model(s) built, students were asked to decide which one should have been their definite robot to be presented at the final exhibition at school or at the university.

The robot chosen were:

- the roaring lion
- the airplane rescue
- the goal keeper
- the goal kicker
- the sail boat storm
- the dancing birds
- the drumming monkey
- the hungry alligator

#### Storytelling:

After that, the teachers and the author asked the students to create a story where at least a couple of robots could interact with each other. So the groups were paired and the storytelling started. The alligator was paired with the “sail boat storm”, the goal kicker was paired with the goal keeper, the drumming monkey with the roaring lion and a second alligator with the airplane rescue.

Here, we report only one of the stories created by the children.



## Pelliper, the fish-eating Pelican<sup>2</sup>

Pelliper was a pelican  
it had a wing and not a hand  
and it ate only fishes,  
It warned its children "Please, don't get  
out from our home!  
It was really harsh  
[...]  
But all of a sudden its children  
were attacked by rabbits  
and Pelliper didn't see them anymore  
[...]  
It wasn't severe anymore ...  
Can't you trust me? It's totally true !

Pelliper now ate carrots  
It didn't have any wheels  
but flew and flew and flew  
It landed in its cave.  
And dreamed about its children  
eaten by the rabbit.  
And it cried a lot while flying  
cause it wanted its children back  
[...]  
When it thought about its children  
neither it could eat nor drink  
[...]

### Scenario building:

A lot of stuff was built by this school to support the storytelling and their robots. Here we report pictures showing the final products. Of course, all the observations about scenario building made for the previous school can be repeated here.

The first picture shows the monkey and the lion as well, living and acting inside the forest.

The second picture shows the rescue plane and another plane in the background.

The third picture shows the big bird, the sailing boat and a volcano in the background.



<sup>2</sup>The original Italian version of the poem is written using correct rhymes!



### Final remarks

All the primary students involved in this project were asked to undergo the following working cycle:

- choose the robot
- build/program the robot
- create the story for the robot(s)
- create the surrounding scenario
- explain to your schoolmates why your group chose that model and how you underwent all the working cycle

This strategy has been proven very useful since it helped the children to make a real sense out of the first two steps of this process, which otherwise could have ended to be less clear without a focused goal in mind. Moreover, the children found a very rewarding experience to tell the other schoolmates what they have done and especially to show their work to parents and other children from other schools.

Finally, they also presented their creations to the annual competition/exhibition Minirobot 2010 where more than two hundred students participated.

### Acknowledgments

The author would like to thank principals, teachers, parents and administrative staffs of the schools involved, who supported the activity with great efforts and patience<sup>3</sup>:

- “S. Domenico Savio” Elementary and Middle School, S. Gregorio (CT) – Italy
- “Diaz – Manzoni” Elementary and Middle School, Catania – Italy

The project of the second school was funded by EU under the P.O.N. general effort.

---

<sup>3</sup>Permissions for using pictures are granted from schools staff or directly to teachers involved in projects reported here.

# Robotics in the Greenhouse

Eduardo Ventura M. (Dominican Republic)

<http://aularobotica.blogspot.com>

## Introduction

The phenomenon of globalization, characterized by increased international competition resulting from the vision of the world as one big market, brings the need for new technological alternatives for boosting competitiveness in food production.

In a greenhouse production system increases considerably when controlling efficiently the weather variables that directly affect the vegetative cycle of plants.

## Hypothesis

Why use robots to improve agricultural efficiency in greenhouses?



## Objective

This project aims to implement tools and robotic technologies in greenhouses to control the climatic variables that optimize efficiency and increase productivity.

## The greenhouse effect

The greenhouse effect is a natural phenomenon in which a portion of solar energy emitted by the earth is absorbed and retained as heat in the lower atmosphere. Existing gases in the atmosphere, primarily water vapor, causing the greenhouse effect. Other gases such as carbon dioxide, methane, nitrogen oxides, ozone and hydrocarbons, also play a role in the greenhouse effect. "

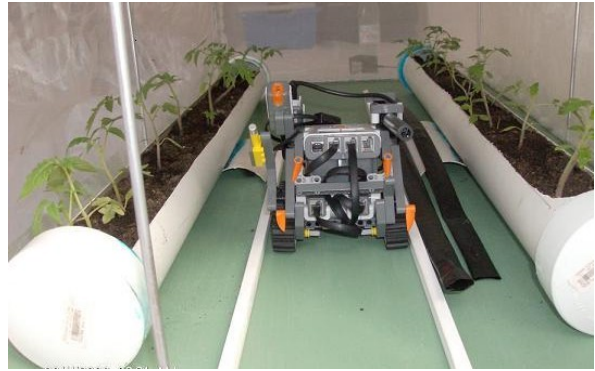
## What is a greenhouse?

It is a closed structure covered with transparent materials, within which it is possible to obtain an artificial microclimate that achieves high productivity, low cost, in less time, without environmental damage, protected from rain, hail, insects or excesses of wind that could harm a crop and thereby grow plants all year round in ideal conditions.

- Sunlight penetrates the walls of the greenhouse warming inside
- The coating is a transparent glass or plastic material.
- These materials scatter light so as to prevent the formation of shadow inside
- The plants, soil or inside the substrates are heated by solar radiation effect

## The advantages of growing in greenhouses are:

- Earliness in fruits and crops.
- Improving the quality and performance.
- Production continued throughout the year.
- Water saving and fertilizer.
- Improving the control of insects and diseases.
- To get more production cycles.
- Separate the production of the climate.
- Allows the implementation of food solutions in agriculture.



## The most important climatic variables to control in a greenhouse are:

- Temperature
- Relative humidity
- Radiation (light level)
- Control of CO<sub>2</sub>

## The robotics in greenhouses

In greenhouses automation has made good progress so that it is now possible that a robot can perform many repetitive tasks with great precision. Its use in the case of agriculture, specifically in greenhouses, this high productive potential.

A robot can be used to measure and monitor climate variables in order to provide optimum plant development.

## System description

In the automation of our study the greenhouse there are three important elements: the robotic system consists of a mobile robot and a fixed station, sensors and actuators.

**The mobile robot** moves inside the greenhouse, unattended with a controller and its sensor, taking measure of climatic variables and sending it wirelessly (bluetooth) to the fixed station for comparison and decision making.

**The mobile robot** has various sensors to measure the microclimate in the greenhouse, temperature sensor, relative humidity, radiation, inspection chambers or robot arms for data collection, which send data wirelessly (bluetooth) to the fixed station.

**The fixed station** is another robot but steady, receives data from the mobile robot and discusses the changes in various parameters with respect to securities initially set according to crop type and stage of plant growth, is responsible for powering actuators.

The third part consists of sensors and actuators; in our study model the mobile robot is equipped with three sensors, temperature, lighting and humidity.



The actuators are part of the fixed station used to trigger the mechanisms that open the windows, activate the irrigation system and exhaust fans. The mobile robot and the fixed station is constructed with an educational robot kit "LEGO MINDSTORMS NXT"

## Operation Greenhouse

The aim with this model of gases is to automate the control of micro-climatic variables in order to improve efficiency and achieve greater performance.

### Temperature

It is the most important variable to control in a greenhouse environment, because it is of vital importance for growth and development of crops.

In our case study, tomato, it is important to know the needs and constraints of temperature at different stages of growth of tomato plant. In our country the optimal temperature range for growing tomatoes is 22 to 28 ° C.



### In all cultivation is vital to know:

The minimum lethal temperature, which is below that which the plant suffers irreparable damage to the plant. The biological maximum and minimum temperature, which indicates the optimal temperature range for the proper development of the plant in a given vegetative phase, as photosynthesis, flowering, fruiting, etc.

### Temperature control by the robot.

Our robot is equipped with a temperature sensor, as the robot can move through the greenhouse, taking moves data of temperature in different areas previously identified, these data are sent to the fixed station wirelessly (bluetooth) is compared to the maximum and minimum expected, if a high temperature alert, higher than expected peak, the temperature is reduced in two ways in our model of study:



The advantage of using a robot to measure this variable is that you can perform with greater precision, at any time and any place inside the greenhouse, taking immediate actions automatically, you can also create a log of every time and place that has been measured for analysis and decision making later.

### Relative humidity (RH)

This is another climatic variables to consider in a greenhouse. The relative humidity is the amount of water in the air, in relation to the maximum you would be able to contain the same temperature "(2) A hygrometer is an instrument used to measure the moisture content of air, soil, plants or a particular gas by means of sensors that receive and indicate its variation. "

### In our model

The robot is equipped with a hygrometer, which measures the RH of air in any area inside the greenhouse, sent to the station sets the value of HR measured, if the value is outside the optimum range (60-70%), the fixed station triggers the corresponding actuator to correct the value of HR and take the appropriate value which does not harm the plant.

In case of over-HR is corrected by activating the irrigation water, spraying water in the environment, increasing ventilation, creating shadows. In our model of study if the RH is very high fixed station automatically activate a water pump to start watering the plants.

## **Importance of Lighting.**

The amount of daylight received at the conservatory is of vital importance to the process of photosynthesis in plants. In our country, receives a considerable amount of daylight, could arise if it is necessary to create shade to control excessive light in the greenhouse.

In our study model, the mobile robot has a light sensor that measures the intensity of light inside the greenhouse. If the intensity is so high that it causes an increase in temperature to a maximum value greater than expected, sending a warning signal to the fixed station in the fixed station compares the values and drives the actuator that moves the mesh to spare on the roof of the greenhouse, thereby reducing the amount of light entering the greenhouse and therefore temperature.

## **Results**

The results of this experience, "Greenhouse Automation" using robots in the control of micro climatic variables has allowed us to observe the following advantages and disadvantages:

### **Advantages**

- Greater precision in the data collected and therefore more responsive.
- High efficiency is obtained in the management of the greenhouse.
- Increased production and improved quality of crop products.
- Compared with open-air crops significantly reduces the environmental impact.
- It frees humans from tasks that might be dangerous when handling chemicals.
- Especially improves considerably the quality of life for producers to have more income.

### **Disadvantages**

- High cost of robotic equipment.
- Lack of qualified personnel in the area of robotics.
- Design of greenhouses with special routes for the movement of robots

## **Conclusion**

The use of robotic technology in the automation of greenhouses in order to increase efficiency and productivity therefore seems a dream for many. The robot is a reality, one should study the feasibility of its implementation at the moment when talking about the need to produce more food to address food crisis approaching across the globe. The conditions to start with this technology are given and the technology exists, just need to unite wills between the public and private universities mainly in the areas of training of qualified personnel in robotics.

# Effective education with limited programming knowledge

Damien Kee, PhD (Australia)

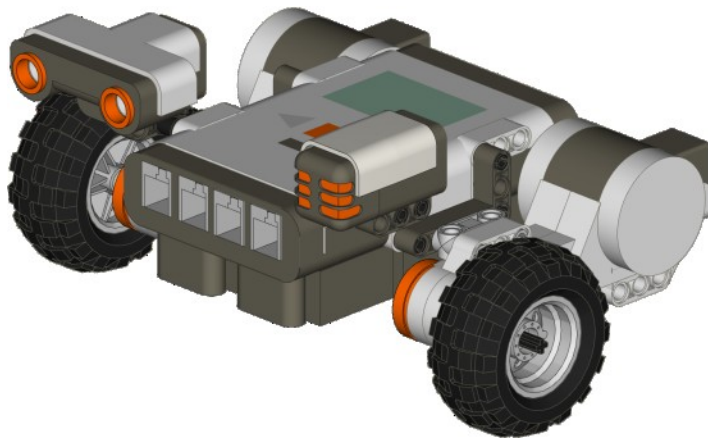
The NXT Classroom – [www.theNXTclassroom.com](http://www.theNXTclassroom.com)

## Introduction

When introducing teachers to robotics, often the first immediate reaction is "That's too complex, I don't know how to program". This attitude, whilst understandable, is often incorrect, especially in light of the teaching tools currently available. This paper outlines a number of educational activities that are possible with only minimal programming knowledge. The activities presented are suited to middle years, but could easily be adapted to accommodate other age groups.

## LEGO Mindstorms NXT

The LEGO MINDSTORMS NXT kit is an excellent way to introduce student to robotics (fig 1). It is robust, easy to build with and many students have had prior experience with modular building systems. The software used is a graphical based software, allowing students to drag and drop various commands, linking them together on screen to create easily customizable programs. The software, whilst simple in look, can be harnessed to perform quite complex tasks such as mathematical operations, variable handling and datalogging. Custom functions can be created, information can be easily passed between blocks and even between separate robots. It is often this high level of software capability that is used to promote the system.



*Fig 1. LEGO Mindstorms NXT Robot*

In many cases however, this serves to discourage teachers, who often incorrectly believe that they will have to program at this high level in order to get any educational value out of the product (fig 2). A high level of complexity provides more sources of possible errors, which can be difficult to diagnose without previous programming experience. This discourages the teacher which in turn discourages the students resulting in a less than optimal educational outcome.

One of the teaching strategies we employ with new and nervous teachers is to show them the vast range of educational activities that can be achieved with only minimal programming. We stress not the teaching of robotics, but the use of robotics to teach fundament maths and science concepts.

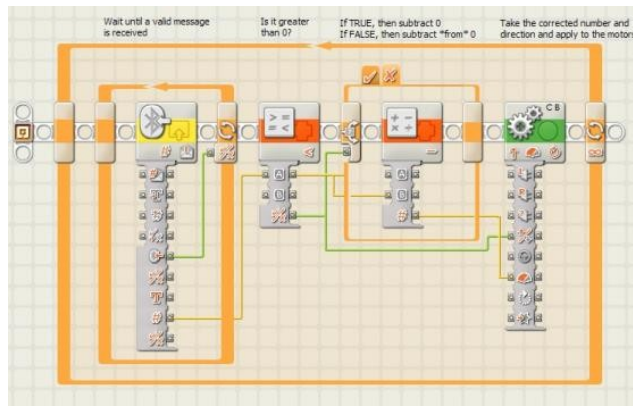


Fig 2. Complex NXT-G program

### Programs with one block type

One way in which this is done, is the use of only a single block for all the initial exercises. The block in question is the *move* block, as shown in fig 2.

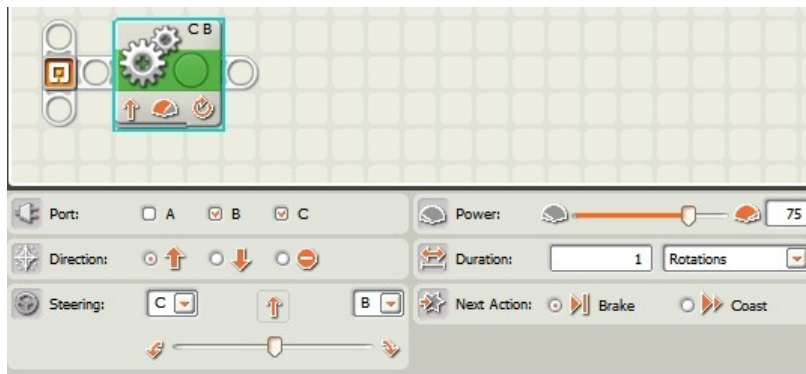


Fig 3. The move block and its associated configuration panel

The *move* block has a configuration panel, which can be adjusted to vary the direction, steering angle, power level and duration of movement (fig 3). Multiple *move* blocks can be strung together to begin to create more complex programs.

### Experiment 1- How Far?

The following questions are posed to teachers/students;

- How far does the robot travel with 1 rotation of both wheels?
- How far for 2 rotations?
- Is there a relationship between how far the robot travels and how much the wheels turn?

This in turn introduces the idea of the circumference of the circle and how it relates to the diameter of the wheel.

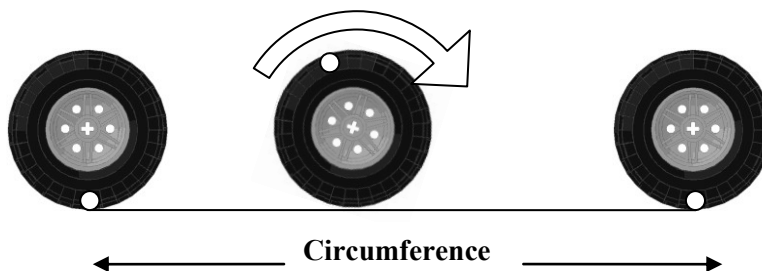


Fig 4. The circumference of a wheel can be defined by the distance it travels in 1 complete rotation.



## Experiment 2 – How Fast?

The following questions are posed to teachers/students

- How far does the robot travel in 2 second? (at 50% power)
- 4 seconds? (50% power)
- 6.3 seconds? (50% power)
- What is the velocity of the robot at this speed? (m/s)
- How many seconds are required to travel 72cm?

For this activity, teachers/students are required to construct a graph of the distance travelled vs the time taken (fig 5). This in turn gives us a roughly linear relationship between distance and time. By reading off the graph at 72cm, it is possible to make an estimate of how long the robot needs to travel the required distance.

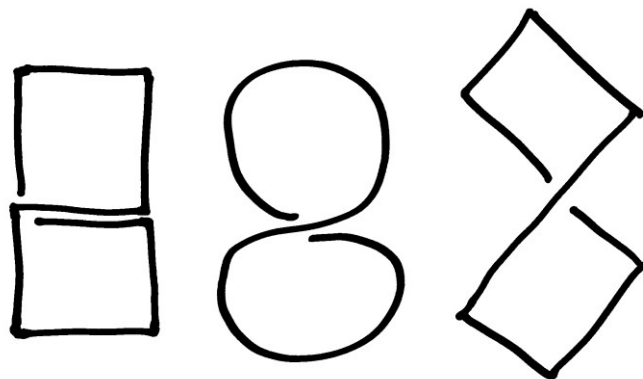


*Fig 5. Distance travelled by the robot, as a function of the time travelled*

This experiment can be repeated for different power levels.

## Experiment 3 – Figure of 8

Make the robot drive in a figure of 8 (fig 6).



*Fig 6. Various styles of a 'figure of eight'*

This experiment now requires the robot to perform multiple movements. Students have to consider where they start, what orientation they begin with and what order of movements is required to complete the task.

Initially, most students attempt to compose the entire program in one go, with a vague hope that it will work as they imagine. In reality, the movements are not as expected resulting in an often comical rendering of the '8'. At this point we ask them to compose just the first two blocks, and to work on those until they are perfect. Once that has been achieved, they are then permitted to add more and more blocks until the task has been completed.

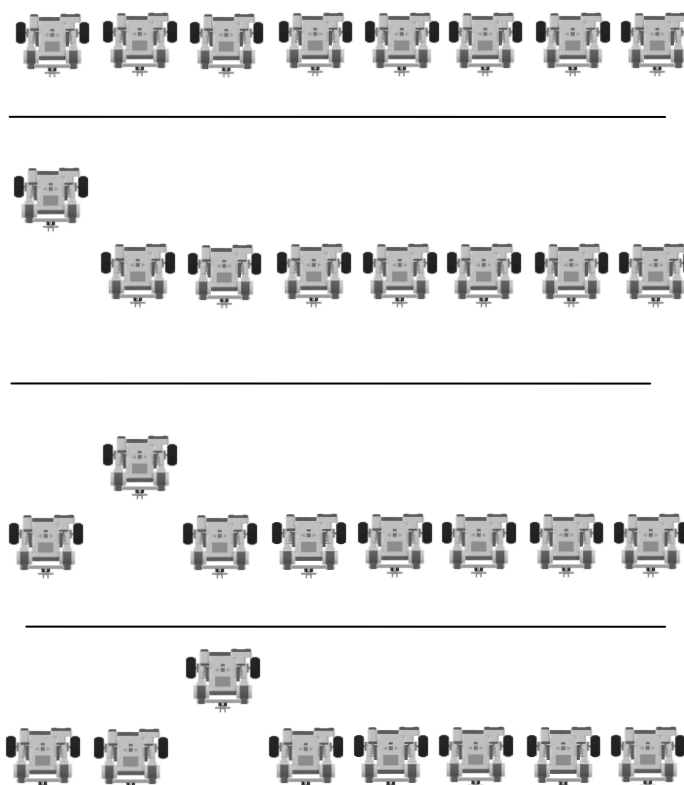
This method encourages students to plan ahead, with a task that requires them to think of several moves at a time. Often sketching out their path on paper, with appropriate notes about 'how far to travel' and 'how much to turn' prove to be extremely helpful.

Once the figure of 8 is mastered, more complex shapes and paths can be proposed.

#### Experiment 4 - Mexican wave

Up until this point, activities have been concentrating primarily on individual robots and their movements. Now, we look towards some sort of co-operation between robots. The method successfully used in class is the challenge of replicating a Mexican Wave, often seen at sporting grounds around the world. In a human Mexican Wave, each individual person in the group only puts their hands up and their hands down. Were they to do that by themselves, it would not be a Mexican Wave. The wave is generated by many people all performing this simple action, but with a co-ordination of the timing when each starts.

A robot Mexican wave involves each robot driving forward for 1 second and driving backward for one second, analogous to the 'arms up, arms down' movement of a human Mexican Wave. Fig 7 outlines how this would look with a line of 8 robots.



*Fig 7. Robot Mexican wave. The first robot on the left moves forward and then backwards. Following this, the 2nd robot performs the same action and so on.*

The start timing can be manually derived, with each robot handler, starting their program a pre-determined time after the preceding robot. Starting each robot manually requires good concentration and an accurate internal sense of timing for each robot handler, something that young students are perhaps not so good at. There is a better, more accurate way to achieve this.

### One extra block - *Wait for Time*

The '*Wait for Time*' block can be found in the common palette, and enables the program to 'wait' at a certain point in the software until a specified amount of time has elapsed, before continuing on to the next instruction.

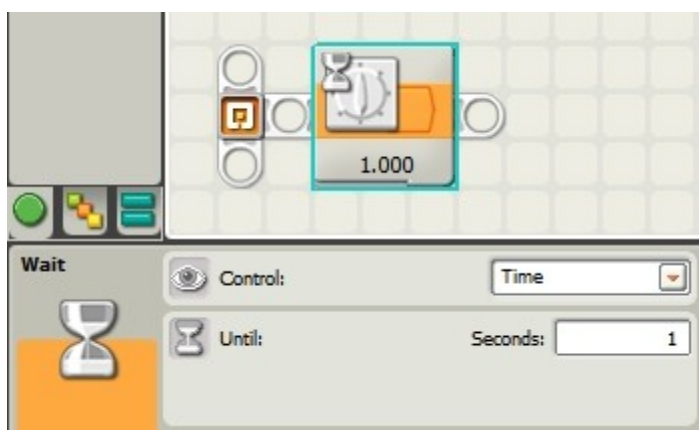


Fig 8. '*Wait for Time*' block

If we assign each robot a number, and allocate a 'wait' time for each, then providing the amount of time between each robot is constant, if all robots start their program at the same time, a Mexican Wave will result.

Action	Robot 1	Robot 2	Robot 3	...	Robot $n$
Wait	0.5 sec	1.0 sec	1.5 sec	...	$n/2$ sec
Forward	1 second	1 second	1 second	...	1 second
Backward	1 second	1 second	1 second	...	1 second



Fig 9. Complete Mexican Wave program. Each robot waits a pre-determined time, and then travels forward 1 second and backward 1 second.

### Conclusion

The above experiments are a good example of how simple programs can be used effectively in an classroom setting to achieve significant educational outcomes.

Yes you can create very difficult programs with NXT-G.... But you don't have to!

# NXT vs. RCX vs. Pico

## Playing to their strengths and being a wise steward of resources

Laura Jones (USA)

I run a lunch-time and before and after –school program (for no salary and no resources) to develop engineering mindsets in elementary school students in a suburb of Washington DC. On average, 250 kids pass through this program every school year learning, among other things, software such as Lego Digital Designer, Scratch and Alice, building roller coasters with K'nex sets and Taurus Toys, and robotics with Pico Cricket, RCX sets and NXT sets. We have spun off into the competitive world of Robotics by recruiting parents to run JFLL and FLL teams.

I get no funding from the school or from the parents of the participants—instead I beg or write small grants. Over the years I have accumulated 5 Pico sets, 7 old RCX sets (new when purchased) and 6 NXT sets. We use these over and over, with my purchasing lost pieces from eBay every summer.

Because of the lack of funding, I need to take good care of my kits and use them to their full potential, which leads me to alternate which version I use at different times during the year.

The kids and I have talked about the differences among the three robotics programs and they have come up with some good insights that I think can be useful to others considering purchases of these kits.

### PicoCricket:

<http://www.picocricket.com/>

From MIT, a programming language and kit of parts that is designed to develop and encourage creativity. The programming language is easy for novices and not at all intimidating. Students get to use both the electronic parts such as the light, sound box, and other sensors and the art supplies to create wacky machines and contraptions that entertain and amaze them.



### RCX/Robolab:

<http://www.lego.com/eng/education/mindstorms/home.asp?pagename=robolab>

You all remember this—the big yellow brick. We have the kits with the Robolab 2.9 license and the USB towers.



### NXT/Mindstorms:

[http://www.lego.com/education/school/default.asp?locale=2057&pagename=nxt\\_concept&l2id=3\\_2&l3id=3\\_2\\_3](http://www.lego.com/education/school/default.asp?locale=2057&pagename=nxt_concept&l2id=3_2&l3id=3_2_3)

The newer version with the grey “brain” and sensors.



I use Pico Cricket with the third and fourth graders (8 and 9 year olds) as an introduction to the wonders of robotics and programming. Kids work in groups of 2 or 3. I find that with a little training in Lego building and teaching what programming is all about, they all can produce something cool by the end of the 8-week session. I also find that the creative emphasis of the product “levels the playing field” for children who do not have much building/computer experience. The goal is to create something and make something happen, so kids who have rarely used Legos or computers can easily figure out what to do. Also, the sensors lend themselves to easy use and creation, as they “make sense” to the kids. I have begun to introduce Scratch programming to younger and younger students, so I anticipate in future years, Pico will be even easier to pick up as the programming interfaces are so similar.

RCX (old Mindstorms)—I use this with 5<sup>th</sup> and 6<sup>th</sup> graders (10-12 years old) and debate every year whether or not they are worth the hassle, but in the end I always find that they are. The hassle involved is that the bricks tend to “lose” the firmware even though it has been downloaded many times, and it takes time and causes frustration when we have to reload every week. The other problem, which I think we finally solved this year, was that the computers wanted us to reinstall the towers from the CD every single time they were plugged in, making more delays. And children today who are on the cusp of adolescence and used to instant gratification do not take these delays well.

NXT –I also use this with 5<sup>th</sup> and 6<sup>th</sup> graders and insist that they work in groups of two, as collaboration is what engineering is all about.

Below I list some of our conclusions

NXT	RCX
Very well suited to “battle” –see comments* below	Open-ended in construction –more flexible and less “war-like”
Seem to be “masculine”—definitely more preferred by the boys	Gender-neutral—while the boys tend to try to make their robots fight, both genders create very funny and creative things
Options for building and support are more readily available	Have to really search for support and instructions, although they are out there
Programming is easier, with a whole set of built-in tutorials and visual instructions, if only they would use them!	Programming is less intuitive. Kids need a better understanding of math and physics to do it well.  Steeper learning curve
Kits make compact robots, but they all tend to look similar	A wide range of finished products become available when the kids get comfortable and start moving beyond the basics
Kids need to understand the principles of motors and gears quickly to get things working	Motor connections make more sense in the beginning
These sets are more for the child who is technology oriented—the look, building and programming seems to be more tempting to the child who is comfortable with machinery already	The open-endedness of the plain brick seems to be more friendly to the child who can envision possibilities  Better for novices

I spend a great deal of time at the end of the sessions convincing the boys that it is not in anyone's best interest to destroy their robots by battling each other. What is it about kids, and boys in particular that they want to destroy?

I hope that these insights are helpful, and I highly recommend that anyone with both types of equipment holds onto them and uses both in the classroom and extra-curricular programs. Each style has its positives and negatives, and they appeal to different kinds of children. But if you really need to get rid of the old yellow brick sets, send them my way! We will treasure them!

# Teaching the Path of Regular Polygons – An Approach to Introductory Programming

Craig Shelden (USA)

**Abstract:** This paper describes one approach to manifest the abstract ideas of geometric shapes in the motion of a Lego NXT robot. Although the explicit goal was preparation for a Lego Robotics competition, the event also exercised each student's problem solving skills and the relationship between the robot and its circular components.

**Introduction:** In the fall of 2009, the author volunteered to support a Southern Maryland, United States elementary school's part-time STEM robotics program. This program began in October 2009 and culminated in an early May 2010 competition for our county. It emulated many elements of FIRST Lego League competitions, including a research project, a presentation, and of course, robotics table challenges.<sup>i</sup>

About twenty fourth graders participated in weekly sessions, conducting the research project through the fall. Due to the large number of participants, two teams were fielded, but most training sessions worked with both teams simultaneously. In early February, after sufficient training robots<sup>ii</sup> became available, robot design and programming began in earnest. One unique aspect to this competition was a requirement to use the Robolab programming environment.

The coaching teacher and volunteer mentors shared the perspective that most of this competition required the robot to drive to a location on the competition table, perform a task, and then return to the home base. They agreed the first learning goal for the students would be the accurate navigation of the robot. However, a weekly meeting schedule, limited competition table access, and twenty children, required an approach that did not rely on the competition table – and the training venue shifted to the classroom and its linoleum floor.

An additional goal the supporting adults held was that each student should program, every session. Others may approach a competitive event with some team division of labor, however this group strove to ensure every attending student programmed during each meeting. This was not always possible, but a consistent effort made its impact felt later.

## The Basics: Straight Ahead

To support mastery of basic straight-line movement, the author prepared an example Robolab program for the students to copy. Although originally intended to be projected, the locations of the classroom computers and the fixed projector screen made handouts more viable. Even with a copy of the program in hand, developing students' Robolab tool and palette familiarity required significant one-on-one attention. Students with slight Robolab experience were soon assisting others, cooperation that emerged more fully as the teams developed.

Two lengths of black electrical tape were placed on the classroom floor an arbitrary but parallel distance apart. Using the program they had just copied, students were then asked to calculate the number of degrees the wheels would need to turn in order to move the robot from one line to another. This request led to mixed compliance.

Some students started by measuring the circumference of the wheels and working through the arithmetic necessary to predict how far the robot would move. Others chose to guess a number of degrees which resulted in some robots moving only slightly, and others careening into the opposite wall. Some programming errors resulted in robots only moving one wheel and pivoting in place; others did not move. Every student that experienced a program failure persevered and eventually made a program that worked, often deviating from the handout.

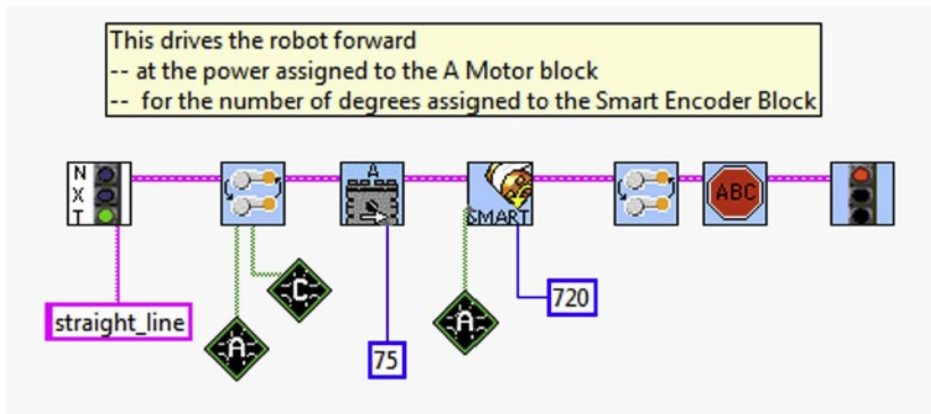


Figure 1. A screenshot of the Robolab Straight Ahead Program handout

### The Basics: Making a Turn

Once the students mastered the straight-line program, the training shifted to making the robot turn in place. The author chose a turning method that kept the robot in place while changing only its heading by driving one wheel forward and one wheel backward.

Again, the students were provided a sample Robolab program to copy and test. The adults discussed the relationship between the robot's wheelbase, wheels, and the desired change in heading with the students. After working through the concepts, a spreadsheet detailing approximate relationships was provided to the students.

Student execution of the sample turning program followed a path similar to the straight programming session – with some students working to copy the program directly, and others striking off in new directions. Regardless of which programming approach they chose, each student benefited from overcoming the errors they encountered along the way.

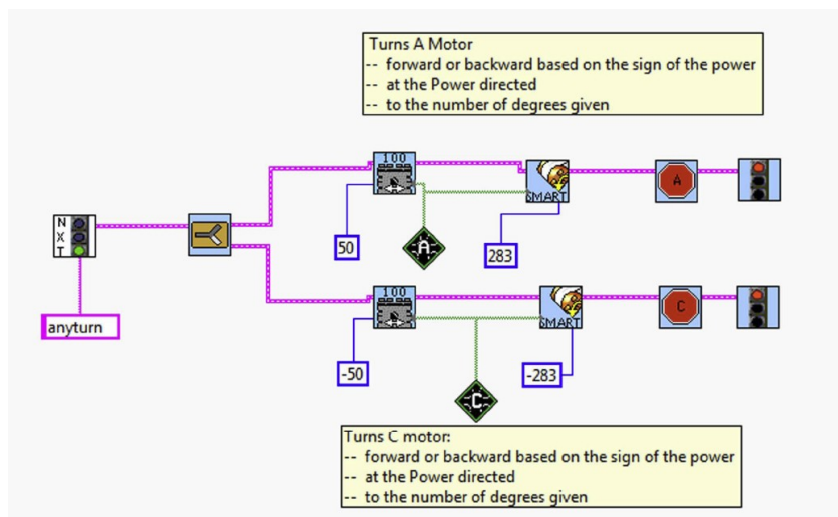


Figure 2. A screenshot of the Robolab Turn Program handout

### Combining Two Simple Programs: Introducing subroutines and Loops

Once the students demonstrated the ability to move the robot forward in a specific direction and to achieve a required heading change; the instructors then provided a sample program to include new control structures and to develop more complex operations from these two simple programs. For this tutorial,



the two basic programs described above were incorporated into the new polygon program as subroutines and then called from within a loop.

When the sample polygon program was first presented to the students, many responded with complaints about how complicated it looked. But after they were shown where the two programs they had just worked through were included in it, they started to really study the program to see what it did. From this perspective, the cumulative development of programming comprehension and skills continued, as each student developed their own insight into the program.

After each student wrote their version of the polygon program and tested it on a square, they were asked to modify it to either change the direction (clockwise or counterclockwise), to change the polygon driven, or to change the length of each side. This required each student to understand which program elements needed changes to accomplish the goal and further cemented the relationships between the components of regular polygons.

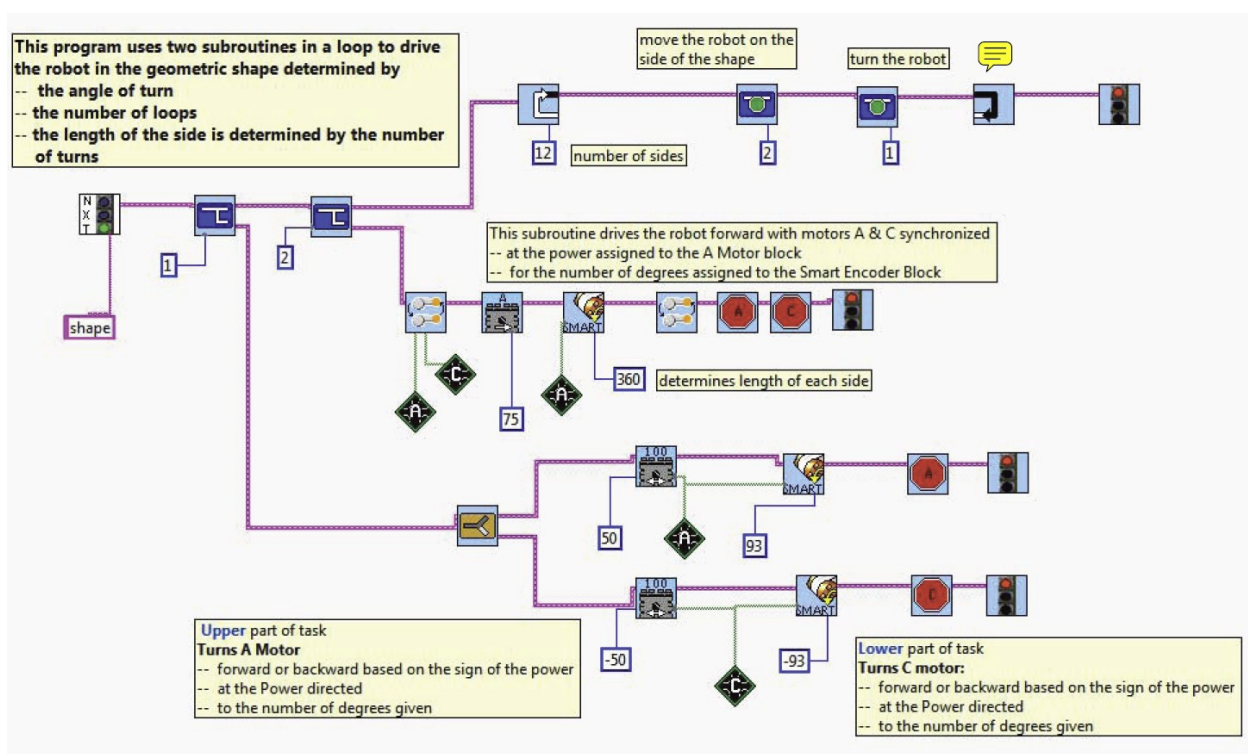


Figure 3. A screenshot of the Robolab Regular Polygon Program handout

### Competition Preparation.

Once the students satisfactorily completed the regular polygon tutorial, they were turned loose to work on the various missions for the competition. Some of the tools provided included:

- Tape measure to determine distance and to calculate the number of degrees required to navigate their robots to the desired positions.
- Paper protractors to allow them to determine the required turn angles.<sup>iii</sup>
- Small printed copies of the competition field with an overlaid scale allowed the students to plan their missions when away from school.<sup>iv</sup>

## Results.

The students were almost all able to navigate their robot to any competition table location using odometry.

Teamwork developed as some students developed programming expertise. After professional requirements forced the author to miss some later competition preparations, he noted the distinct lack of calls for his programming advice as students helped each other. Although this absence was beyond the author's control, the removal of the adult crutch proved very useful for generating teamwork and independent programming skills. A more deliberate approach to removing adult support would probably yield similar benefits.

## Follow Up.

- **Repetition First – Then Loops?** As means to lock in the concept of loops, an alternative could have used repeated program steps – even to the point of making the programming somewhat repetitious. Then by sharing the loop control structure to those who display boredom, the lesson may have been more deeply appreciated, and other students could be more likely to follow the example. This may be an approach for the next team preparation.
- **Programming Language Variations.**
- Some of the students had experience with the NXT-G programming environment, although most had no programming experience. Two of the adult support team had some NXT-G experience, and no Robolab skills. The competition's requirement to use Robolab therefore led to some quick adult learning. The event organizers supported this with several training sessions. The tutorials provided the students also served to familiarize the adult mentors with Robolab.
- One of Robolab's strengths this effort revealed was the self-documenting nature of its screenshots. Screenshots from NXT-G need to include the palette details from each block – which leads to more complex images.
- Another revealed Robolab strength was the ability to display the subroutines as part of the program. This allowed the students to connect the parts of the program very intuitively, without the learning overhead of developing NXT-G MyBlocks, or LabView SubVIs.
- Because Robolab requires motor turns to be directed in degrees, each student had to work a bit more arithmetic than if they had only needed to program in units of rotations as NXT-G allows.
- **Mimicry or Learning?** Perhaps the most contentious element of this approach is the use of sample programs and handouts. Some may argue that the students only mimicked the programs from the handouts, and did not develop a detailed understanding of the material. Although this may be a valid criticism of the approach taken, the instructors needed to start somewhere, and sharing sample programs that worked seemed appropriate to set an initial baseline of programming knowledge. Further, once the mimicry portion of the tutorial was completed, the students then had to use that knowledge *in different ways* to meet the competition's mission requirements.
- **Leaving the Path.** Students who chose to deviate from the program handouts in their own programs often ran into trouble making their robot perform as desired. Some chose to review the handouts and revise their programs to more closely resemble the handouts. Others chose to plow ahead and often developed very different programs that accomplished the same tasks. Depending on the individual's style, either approach seemed to meet the learning objectives.

## Resources.

- Robolab's online help provided most of the support needed, once the right programming block was selected.
- Eric Wang's podcasts posted at Tufts University Center for Engineering Education and Outreach provided outstanding introductions to Robolab: <http://legoengineering.com/podcasts-subminuteachingresources-114.html>
- Eric Wang's book, Engineering with Lego Bricks and ROBOLAB, Third Edition <http://www.legoeducation.us/store/detail.aspx?KeyWords=wang&by=20&ID=1436&c=0&t=0&l=0> provided outstanding examples of Robolab approaches for the adult coach and mentors.
- Robolab Reference Manual [http://www.legoengineering.com/index.php?option=com\\_docman&task=cat\\_view&gid=46&Itemid=78](http://www.legoengineering.com/index.php?option=com_docman&task=cat_view&gid=46&Itemid=78) This comprehensive reference guide quickly resolved any remaining questions.
- A more detailed version of the tutorial in both Robolab and NXT-G as well as presentations and spreadsheets are available at <http://www.sheldenrobotics.com>.

### About the Author:

Craig Shelden is a retired naval officer. He lives in Southern Maryland and occasionally mentors Lego Robotics teams. He recently founded Shelden Robotics in an effort to lower the cost of entry for STEM robotics programs.

### Contributors:

- Mrs. Wendy Bowen
- Mrs. Kelli Short
- Mr. Ajai Viswam

---

i See College of Southern Maryland, Robotics Competition, Junior Division – Save the Bay at <http://www.csmd.edu/roboticschallenge/ms/2010/>

ii Domabots

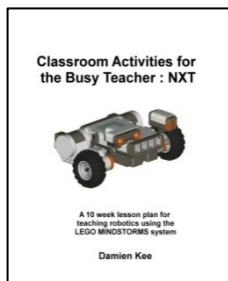
iii Small plastic protractors seemed inadequate for most of the robot navigation tasks, so the author purchased a couple pads of Defense Mapping Agency Maneuvering Boards and had the students cut out the circular element to use for their heading change estimation tools. Since each sheet is larger than a Domabot, the challenges associated with small protractors was minimized. See Landfall Navigation item # 5090 at <http://www.landfallnavigation.com/dmacharts1.html>

iv For an example, see the Field Strategy Worksheet at [http://www.telepathicturtles.org/csm\\_2010.html](http://www.telepathicturtles.org/csm_2010.html)

# Teacher Resource Books

## Classroom Activities for the Busy Teacher: NXT

This book outlines a 10 week set of lesson plans for teacher wishing to implement robotics in their classroom. A set of robotics challenges are presented, centered around the LEGO NXT MINDSTORMS system. The workbook includes 10 robotic based challenges as well as 3 additional modules with assessment activities covering Robots in Society, Flowcharting and Multimedia Presentations.

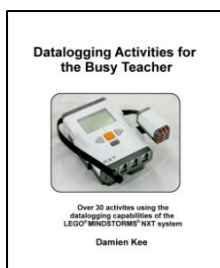


Each module includes:

- A real world scenario
- Theory of the concepts presented
- Teachers notes outlining common issues and how to solve them.
- Example Programs in the NXT-G development environment
- Extension activities
- Student worksheets

## Datalogging Activities for the Busy Teacher: NXT

This book provides over 25 different datalogging activities that can be easily implemented in class. It utilises the new NXT-G 2.0 software to quickly and easily configure experiments, and display the results. Each experiment comes with teacher notes, sample graphs and student worksheets.



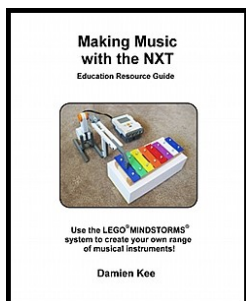
Experiments are provided for the following sensors:

- Touch Sensor
- Sound Sensor
- Light Sensor
- Distance Sensor
- Rotation Sensor
- Temperature Sensor

## Making Music with the NXT

Looking for new and exciting activities to extend your LEGO MINDSTORMSNXT system?

This book will take you, step by step, through the construction and programming of a variety of fun and engaging musical instruments. Each chapter addresses a different way to make music, and provides suggestions for further projects.



Chapters include:

- Onboard speaker
- Rotation Sensor
- Ultrasonic Sensor
- Percussion Instruments
- Drums
- Trumpet
- Complete building instructions for the Trumpet and Xylophone player

# www.theNXTclassroom.com