

# **FLL**

# **Coaches' Primer**

**"Everything"**  
**You Need to Know ...**

Version: 3.0  
Date: July, 2005  
Written by: Skye Sweeney [Skye@fl-freak.com](mailto:Skye@fl-freak.com)

Table of contents

1	Preface .....	4
2	Introduction .....	4
3	What Is FLL .....	4
4	Getting started .....	6
5	Child Safety .....	7
6	Money.....	8
7	What to Expect.....	9
7.1	Standard Meeting.....	9
7.2	Competition Day.....	10
8	Forming a team.....	10
9	Schedule .....	11
10	Preparation.....	12
10.1	Install Software.....	12
10.2	Purchase Supplies .....	12
10.3	Coach’s Binder .....	13
10.4	Team’s Binder .....	13
10.5	Competition Table .....	13
10.6	Subject Matter Experts .....	13
10.7	Mentors.....	13
11	The First Few Meetings.....	14
11.1	Gracious Professionalism .....	14
11.2	Team rules .....	15
11.3	KISS .....	15
11.4	Photo/videography.....	15
11.5	Clean Up.....	16
12	Coaching.....	16
13	Teaching Phase.....	17
13.1	Teaching Hard Skills .....	17
13.1.1	RCX.....	17
13.1.2	Programming .....	18
13.1.3	Locomotion.....	19
13.1.4	Odometry.....	21
13.1.5	Line Following .....	23
13.1.6	Landmark Navigation .....	25
13.1.7	Manipulators.....	26
13.1.8	The LEGO Modulus .....	27
13.1.9	Gearing .....	27
13.1.10	Ganging of Sensors.....	28
13.1.11	Mini Challenges.....	28
13.2	Teaching Soft Skills.....	28
13.2.1	Brainstorming .....	28
13.2.2	Analyzing the Challenge.....	28
14	Solution Phase .....	29
14.1	Sub Teams .....	29
14.2	Dress rehearsal.....	30
15	The Research Assignment .....	30
16	Competition Day .....	31
16.1	Travel.....	31
16.2	Setup.....	31
16.3	Opening Ceremonies/Coaches’ Meeting .....	32
16.4	Rounds.....	32
16.5	Food.....	33
16.6	Research Assignment .....	33
16.7	Software Interview .....	34

16.8	Tech Interview .....	34
16.9	Floating Judges .....	35
16.10	Spirit .....	35
16.11	Closing/Awards Ceremonies .....	35
17	Technical Information .....	36
17.1	Mindstorms Software .....	36
17.2	Batteries .....	37
17.3	Competition Mats .....	38
17.4	Competition Table .....	38
17.5	Programming garage .....	39
17.6	RIS “My Block” issues .....	39
17.7	RoboLab Tricks .....	40
17.8	Lighting .....	43
17.9	The Rotation Sensor .....	44
17.10	A Primer on Mindstorms .....	45
17.10.1	Variations on the Mindstorms Kit .....	45
17.10.2	Working with Mindstorms .....	46
17.10.3	Storage of Mindstorms Kits .....	46
17.10.4	Multiple Mindstorms Kits .....	47
18	Books .....	47
19	Publicity .....	48
20	Resources .....	49
20.1	FIRST .....	49
20.2	PTO/PTA .....	49
20.3	Documents .....	49
20.4	Web Sites .....	49

# 1 Preface (to the 3<sup>rd</sup> edition)

If this primer reminds you of the official FLL Coaches' Handbook (black and orange book) it should! Early in 2005 FIRST asked if they could use this document as a resource for their new handbook. I happily agreed. In my mind this is what FLL is all about; people helping each other out to make the best possible product for the students.

To prevent confusion with the FIRST document, I renamed my document from the Coaches' Handbook to the Coaches' Primer. I will continue to maintain this document as it contains more information than the FIRST document.

## 2 Introduction

*"The trouble with doing something right the first time is that nobody appreciates how difficult it was." –Walt West*

This primer was written in the hopes that nobody will appreciate you!

I became a FIRST LEGO League (FLL) coach in 2001 when BAE Systems, the company for which I work, started to sponsor local teams. I was already familiar with the LEGO Mindstorms<sup>1,2</sup> kit and my son was interested, so I signed up to coach. It was a great experience not just for the team members but for me as well. FLL matched up closely with my personal interests, so I invested large amounts of time with the team. By the end of that first year, my rookie team had a first place win at a local tournament and a second place win at a state competition. At the end of the second year, we had added four more trophies including a Director's Award at a state competition.

I have tried to capture everything I learned that first year in this primer. Understand that what is presented here are my personal experiences and opinions. They are not necessarily those of BAE Systems or FIRST. They are not necessarily those of other coaches. Hopefully the information, ideas and suggestions will be useful.

Some of this information was distilled from email conversations I had with other local coaches during that first year. I wish to thank these people for all their help. Some of this information came from the numerous web pages I have read. When remembered, I have given attribution. When not, I give my apologies.

Every coach's style is different. You should consider this primer to be an idea book, not an instruction manual. I am very much a non-conformist and would rebel at being told exactly how to do something. Do what makes sense to you as long as it conforms to the FLL rules and to matters of safety.

If you have any information to share or a differing opinion, please send it my way.

[Skye@fll-freak.com](mailto:Skye@fll-freak.com)

## 3 What Is FLL

FIRST LEGO League is a program to promote technology and science at the grade school level. In an eight week period, teams of students, aged 9 to 14, create two major products. The first is an

---

<sup>1</sup> Mindstorms is a trademark of the LEGO Corporation.

<sup>2</sup> Editor's note: The trade mark Mindstorms is indeed plural. It will be used this way even when a singular kit is implied.

autonomous robot to perform various tasks. The second is a presentation based on a research project. At the end of the eight weeks, teams meet for local/regional, and state competitions. At competition, a team's products are judged, and awards are presented in numerous categories.

FLL gives a team member a glimpse of what life is like in a technology career. A team member gets to see a product develop from a specification to a design approach, to prototyping, to development, to integration, to test, and finally to product. Each stage provides its own unique problems and challenges to the team.

Notable among these problems is that problem solving takes time and that not all solutions work. FLL team members soon realize that much of what they do does not meet expectations, but that persevering will eventually give them a working solution. Working as a team will accelerate this process only if they are willing to listen to each other's ideas. Adults could benefit from FLL!

*"Failure is an event, never a person."-- William D. Brown*

Defining a problem, brainstorming, evaluation, conflict resolution, negotiations, and time management are other skills team members will have to deal with. FLL is truly a microcosm of adult life. Learning these skills now will give a team member a huge advantage in their life.

FLL teams use the LEGO Mindstorms kit as the basis for their robot. This kit is a wonderful collection of LEGO parts. In center stage is a large LEGO brick. This brick is in fact a sophisticated microcontroller. In essence, it is a small computer all to itself. The microcontroller is known as either the "Brick" or the "RCX". Also in the kit are several hundred other parts including traditional LEGO building bricks, gears, motors, cams, sensors, axles, beams, and plates.

The FLL program can be an in-school or after-school activity. Most teams around the country are after-school based. Teams are selected from and represent a given school. These can be public, private, parochial, or home school cooperatives. These teams meet outside of the school day. They can meet right after school, in the evenings, weekends, or any combination. They tend to meet at the convenience of the coaches and mentors.

Teams can also be created from other pools of students. A group of friends, a Scout pack or troop, or a church group are all possible. A company could also create a team drawing from employee children.

FLL teams meet on a regular basis for as little as 45 minutes a week to 10 or more hours a week. The time commitment is up to the coach and the team as a whole. To be competitive, a rookie team will often need to meet more often than a veteran team. There is a large amount of learning that must happen before you can give serious thought to a solution to the year's challenge.

Your team can meet wherever is appropriate. For a school based program, the school itself is ideal. They will often have the needed computers and the space to set up the playing field. For other types of teams, a meeting hall, a private home, or a company conference room are all possible. Storage for the playing field (competition table) should be an important consideration.

Students on a team must be ages 9 to 14. In general, this corresponds to grades 4 to 8. The age cutoff is January 1 of the year of competition. This means that some 15 year olds can be on a team.

Teams are properly limited to 10 students. Consider that most teams only have one Mindstorms kit. Ten kids trying to get their hands on one kit is asking for trouble. Personally, I would limit the team based on the number of Mindstorms kits at your disposal. For one kit a team of 4 to 6 would be about right. More kits allow you to add team members or to do more with the team you have.

## 4 Getting started

Before you start a team, you need to figure out who will sponsor and pay for the team (see section 6). Most teams are sponsored and based as an after school activity. This section assumes that this is the case. If not, the same points apply, but just the cast of characters will be different.

Arrange a meeting with the school principal. Explain the concept behind FLL and the benefits of having a team at the school. If done properly, an FLL team can benefit more than just the team members. If you involve the teachers early, they can bring some of the experience into the classroom. Team members and coaches are recruited by teachers to talk to classes about all aspects of the challenge. A school wide assembly can demonstrate the robot in action and generate excitement.

The school should provide you with a place to meet. Preferably this will be a room that has a few computers. A school library is often ideal. It has open floor space, computers, and research books. Meeting at your home is probably not a good idea. First it is unlikely you will have the space to host 10 kids, a competition table, the computers, and all the LEGO. Second, you run the risk of liability (see section 5 on Child Safety).

If you elect to build a competition table, you will need a place to store it. Without the optional legs and lights, the table is simply a sheet of plywood with a wood lip around the edge. Storage for other supplies is at first appealing, but I found it to be un-needed and un-wanted. The Mindstorms kits are valuable and can disappear too easily. Besides, you will need to return the kits home to replace batteries. Having the kids unload and load the equipment from your car is not a hardship and teaches them responsibility.

Ask the principal to try to find a teacher to act as liaison. The liaison does not need to help or even attend any meetings. Their involvement is simply to be able to give the other teachers an update on the team's progress. Send your liaison a progress report by email one or twice a week and cc the principal. The liaison in turn informs the staff at their next staff meeting. This gets the whole school to understand what you are doing.

Your principal must also explain to you the school districts rules on adult involvement and child safety. In my district, adults that meet with students after hours in a non-supervised environment must have a criminal background check and be fingerprinted. This is done at the school's expense. All my volunteers and I submitted to this simple procedure. It's a small price to pay for our kids' safety.

You must also ask permission to use the school's computers. These may be NT or XP computers, and you will need to be given accounts and have their computer service personnel install your software for you. Even if you can install the software yourself, involve or at least inform the school. Computers that have access to the Internet would be ideal. A notebook would also be nice as a desktop computer gets a little awkward dragging to competitions.

The principal must also grant your meeting time. Evening use of the building may need special authorization.

Your school district may also have rules on photographing students. Make sure each team member's parent has signed the school's video/photography release form. You will need this permission to take photos during meetings and other events. Also be aware that most FLL tournaments will have a similar form for the parents to sign as they are often video taped.

## 5 Child Safety

The students' mental and physical safety must be foremost in your mind. The place to start is a chat with your school principal. Make sure you understand and follow all of the rules and regulations. These days this is likely to include a police background check and fingerprinting for you and other key volunteers.

Always make sure you have at least one other certified adult present when working with the team. What would you do in the event of an injury or other situation? The two-deep leadership also protects you from accusations of ill behavior. In the same vein, do not allow other adults to separate students from the group without them being two deep as well.

Make sure you have with you each student's emergency phone number. Learn to use the telephone system where you meet. Some phones require special keys or codes to access an outside line. Learn your local emergency numbers. Not all communities have 911. Also know the address of your location.

Make sure the parents inform you of any allergies or medical issues you should be aware of. Food allergies can be very severe. The safe (and clean!) choice is not to provide snacks. If a snack is appropriate, have the students bring them for themselves.

When taking a field trip or going to a competition, make sure to fill out the school's paper work. If not working through a school, make sure all drivers are licensed, registered, and insured.

Do not allow parents to simply drop off their children. They need to wait with the team member until the two-deep leadership is in place. You do not want to be in a position to have to call all the parents if a certified volunteer cannot make a meeting.

Do not allow the students to leave without informing you. Nothing is as frightening as having a child disappear on you. Keep the team together as long as possible. We all walk to the curb together with our equipment, perform our team spirit 'thing'. At this time, students can let me know they are leaving with their parent.

Keep your ears, eyes, and mind open at all times.

## 6 Money

Forming an FLL team is not a cheap endeavor. The first year cost is likely to be around \$900.00. Veteran teams will have an easier time at about \$425.00. The major costs involved are in purchasing Mindstorms kits, Challenge Kits, FLL registration, batteries, competition table, tackle boxes, and tee shirts and/or other team spirit items. The following is a sample list.

<b>Item</b>	<b>Rookie</b>	<b>Veteran</b>
<b>FLL Registration</b>	\$150.00	\$150.00
<b>FLL Robot Set<sup>3</sup></b>	\$260.00	\$0.00
<b>FLL Challenge Set</b>	\$50.00	\$50.00
<b>Tournament registration</b>	\$35.00	\$35.00
<b>Tee-shirts<sup>4</sup> for 12 people</b>	\$120.00	\$120.00
<b>Materials for simple competition table</b>	\$65.00	\$0.00
<b>Folders, binders, paper, ...</b>	\$40.00	\$10.00
<b>Tackle box</b>	\$17.00	\$0.00
<b>NiMH rechargeable batteries and charger<sup>5</sup></b>	\$90.00	\$0.00
<b>Alkaline batteries</b>	\$60.00	\$60.00
<b>Total</b>	\$887.00	\$425.00

You have many possibilities in funding your team. The first would be to simply write a check from your personal funds. Doing so is easy, but certainly unrealistic. You can also simply split the bill between the members of the team. Somewhat more realistic but still a large investment.

The second is to find a sponsor. Look to your community for a technology company. They have a vested interest in creating a talented work force. Teaching students that technology is a good fun career will pay back as these kids graduate from college. In my community, the largest employer funds 20 or more FLL teams, underwrites the state competition and funds two FIRST teams. This is a significant investment. Improved work force and community relations make their financial commitment a win-win situation.

A small technology company will not be able to match the above, but they should be able to fund one team. Let them know the benefits of sponsoring you.

Don't just ask them for money! Ask them to provide mentors to come to your meetings occasionally. Ask them to allow your team to dry run their robot and Research Assignment presentation at a brown bag lunch. Ask for their logo to place on your competition poster. Ask them for the keys to the vault!

When going to talk to their representative, try to bring a working Mindstorms robot. If you do not have your own kit, borrow one. Make a sample robot from the instructions or a book and bring it.

<sup>3</sup> An FLL Robot Set is a modified Mindstorms kit. It has an extra motor, light sensor, rotation sensor, and some miscellaneous plastic parts. If you already have a Mindstorms kit, it would be cheaper to purchase the extra parts separately.

<sup>4</sup> This is assuming a simple one color silk screen on a Hanes tee-shirt at \$10.00 a shirt.

<sup>5</sup> Charger and batteries for two Mindstorms kits. See section 17.2



Nothing speaks louder than a robot doing its thing. Most adults in a technology company would have given their eye teeth for a program like this when they were kids.

Fund raising is also possible. However, I would think really hard before pursuing this avenue. The time commitment of just running a pre-paid team is substantial. Trying to fundraise at the same time might be too much. Of course, with the proper lead time, a team can fundraise in advance of the competition season. Fundraising also tends to bind a team together even tighter. They have all shared in the pain of begging for money.

Even if you have found a sponsor to fund your team, you should still require a payment from the team members. It makes them appreciate the experience more, and will give you a kitty for costs your sponsor does not pickup.

## **7 What to Expect**

FLL can be a fantastic experience for the coaches, mentors, students, and volunteers. The energy and enthusiasm is simply incredible. As a coach, you will have the greatest impact on the success of a team. The fate of your team will not come down to your technical skills. You will find that the kids can figure stuff out by themselves with terrifying speed. What will set the tone for the team is your desire to be on the team. Like animals sensing fear, your team will sense if your heart is in it.

FLL is a major commitment. For three to four months, your life is likely to revolve around your team. To start, it will take a large block of time. It can range from as little as 4 hours a week to well over 15 hours a week. It will depend on how often your team meets. A rookie team of younger students has a large learning curve to be competitive. Three two hour meetings a week would not be amiss. A veteran team of older students with resources at home can get by with as little as meeting one hour a week.

You will spend at least 2 hours each week in prep work. This will include reading the newsgroup discussions on the FLL web site, coordinating help, maintaining equipment, interfacing with your sponsoring organization, purchasing supplies, registering for competition, and countless lunchtime talks with frazzled friends.

Besides the time, you must have the right attitude. First, you must enjoy working with kids. Some meetings will run like clockwork, others will be disasters. You must accept them both, learn from your and their mistakes and continue with a smile. Second, FLL is a competition for students not adults. It is their ideas and their robot that must be judged, not your ideas hinted, cajoled, or forced on them. You must be willing to allow the team to make its own decisions and its own robot.

Think carefully before you volunteer or allow yourself to be drafted as a coach. It might be one of the most rewarding experiences of your life. It could also be a disaster. If you don't volunteer, the kids will suffer for lack of a team. If you do volunteer, they can suffer just as much if you can't be there for them. It's not an easy decision.

*"You have a lifetime to work, but children are only young once." -- Polish Proverb*

### **7.1 Standard Meeting**

Here is a typical FLL meeting day. First I have to get to work a little early so that I can get home earlier. After work, I rush home and eat dinner right away. As my family is cleaning up, I load the car with all the team's equipment. My son and I then drive over to the school at 6:15 and unload the

car onto the sidewalk. We now wait for the team to start showing up. The first team members that arrive carry all the equipment into the school and into the library where we meet. When the equipment is inside, I go in and help set up the competition table. Parents must stay until that evening's volunteer or my co-coach arrives.

At 6:30 we normally have most if not all of the team present. We sit at a table and go over any announcements I may have. Then we go around the table for any information they need to share. We then look at the schedule to see how far behind (we are never ahead!) we are. It is then off to work. The team is normally broken down into sub teams, each working on some aspect of the challenge. My co-coach and I walk the room, and answer the thousands of questions they have. At 8:15, we give them a five minute warning. At 8:20 I call for clean up. Each team member has their own assigned job. When it's done, they must go back to the table to sit and wait. When we are all ready, we go over what we accomplished and update the schedule.

It is now 8:30 and the team leaves as a group to the curb outside carrying all the equipment. We perform our team spirit stunt. They are then allowed to leave with a parent after they tell me. I wait till the last member is picked up before getting the car and loading it up. My son and I get home around 8:50. He goes off to get ready for bed and I unload the car. I now go over the equipment charging batteries and such. I am normally done by 9:30.

## **7.2 Competition Day**

Read section 16 for a full description of what the competition day is like.

# **8 Forming a team**

Early in the school year you sent a flyer home about the LEGO team. On the night of the informational meeting you drive over to the school. "Strange, the parking lot is full...must be a basketball game in the gym" you think. Surprise! All these cars brought kids who want to join your team. A feeling of dread takes hold. How are you going to select just 10 kids from the sea of smiling faces?

I have had this happen to me. I have seen it happen to other coaches. It could easily happen to you. With a little planning you can get through this!

To start with, when you send home the flyer, try to downplay the LEGO aspect. Many of the kids thought that this would be 'playing' with LEGO. Like making a castle and storming it with LEGO people in an epic battle to see who will rule the universe. Try to promote the robotics and technology aspect, but not so much as to scare everyone away! It's a fine balance.

Next, realize that you cannot have an informational meeting and select the team the same day. The first meeting should be at the meeting location at the time you will have your meeting. The first meeting should explain to the parents and students what they are getting themselves into. Tell them the meeting times. Tell them about their involvement as helpers. Tell them as much as you can. Answer their questions. Hand out the contract that the parents and team members must sign (see Appendix C). Tell them they must bring it back signed to the next general meeting. And tell them they must do some activity before the next meeting. I asked them to make a name tag related to the current year's challenge.

I also hand out an open ended question about why they want to be on the team, or what they could offer. The students must fill this out before the next meeting.

At the second meeting, you should have many fewer people. The 'playing' kids will not be there. The kids who can't make your meetings will not be there. The parents or students who can't sign the contract will not be there. The kids who were scared off by the essay question will not be present. If you are lucky, the number of people will not be more than 10.

If you are unlucky, you now must select the team. I bring an icebreaker activity with me (Appendix B) that I use to watch the interaction of the students. I also bring another activity (Appendix A) that they can work on, turn in, and I can grade. After this meeting, the principal and I go over the essays, the activity, my notes about interactions, and the list of parents that will volunteer. Between all this information we select the team.

Of course the other solution is to start a second team! You have the parents together. Give them the speech! Just make sure you tell them the whole story!

You can call me "reverse biased", but if a girl wants to join a team, she should be given priority. Women have too few opportunities to experience technology not to give them a little extra help.

*"Educate a man and you educate an individual.  
Educate a woman and you educate a family." --A.Cripps*

Be aware that if you need to downsize the team after the second meeting, you are likely to get flack. Try not to let it bother you. If it's so important for them to have their child on the team, they could step forward to be a coach. You can only ever satisfy yourself.

*"I don't know the key to success, but the key to failure is trying to please everybody." -- Bill Cosby*

## 9 Schedule

The FLL challenge is issued in mid September and competitions occur in mid December. FIRST assumes that you will only truly get the months of October and November to work the challenge. This results in their 8 week period. But if you can start right out of the gate, you can get about three months to work on the challenge.

You will need to visit the FLL web site to get all the proper dates. The web page will certainly have the date they intend to issue the challenge. Dates for competitions in your area may be sometime in coming. Local and regional events are often put together at the last minute. State events are more stable, but final scheduling is often up in the air. You may also find that there is no competition in your immediate area and your only choice is a road trip or to host your own competition. If you want to run a competition, contact FIRST. They have resources to help you.

Do not count on the Thanksgiving vacation as a time to meet. Many team members will be at grandma's in Timbuktu eating roasted yak. It can be a time for some team members to catch up on some solo projects.

As a coach, get yourself a calendar of this time period and add all your commitments to it. Block off the meetings, and free up your weekends. Weekends are great times to meet with a subset of the team to work out problems. We used this time to finalize the art work for the tee-shirt, and to master complex computer skills needed for their PowerPoint presentation. Towards the end of the season, my co-coach hosted Saturday programming marathons at his home.

Get to the dentist and the doctor before or after the FLL season. And under no circumstances are you allowed to get ill during this time.

Tell your boss what you are up to. Hopefully they are understanding and can give you some flexibility. Also it is not a good time to have them send you to Katmandu for six months.

Next get yourself a large sheet of paper and make a calendar for the team. Add the meetings and important dates. Set major goals in pen and minor goals in pencil or on PostIt notes. Bring this calendar to each meeting and update it as part of your business meeting. It's good for the team to visually see how they are doing.

*“There cannot be a crisis next week. My schedule is already full.”*  
– Henry Kissenger

## **10 Preparation**

There is quite a bit you can do in advance of your first meeting. Learning as much as you can about FLL would be at the top of your list.

### **10.1 Install Software**

Work with your sponsor to install the software of your choice (see section 17.1) on the computers. Make sure you can write a simple program and download it to the robot. Your first meeting is not the time to learn you have compatibility issues, or your programming tower does not work.

### **10.2 Purchase Supplies**

Besides the materials supplied by FLL you will need some other items. These items will change from challenge to challenge as well as from team to team. I would suggest purchasing a plastic bin to simplify storage.

- A pencil box with pencils, pens, scissors, glue sticks, colored markers, ...
- Roll of masking tape.
- Roll of black electrical tape.
- Roll of paper or large sheets. Perhaps you can use the art room supplies?
- Ruler.
- Batteries. See section 17.2
- Coach's binder. (2 or more inches). See section 10.3
- Team binder. (1 ½ or more inches). See section 10.4
- Index cards.
- A tackle box for each Mindstorms kit.
- Materials for the challenge table. See section 10.5.
- A box of floppy disks.
- Good heavy duty exterior grade extension cord and power strip.

### **10.3 Coach's Binder**

Create a large (2 or more inch) three ring binder. All critical information you need should be in this one place. This should include:

- A roster of the kids and their emergency contact numbers.
- Attendance sheets.
- A copy of the challenge
- The instructions for the challenge kit
- The coaching information that came from FLL.
- The contracts and other selection material.
- Copies of any school forms
- A copy of every paper you hand out.
- Blank paper for notes.

### **10.4 Team's Binder**

Create a medium three ring binder the team will use to collect information. Material from the Research Assignment, art work ideas for their team logo, design ideas for the robot, and anything else of importance should live here.

### **10.5 Competition Table**

Instructions for making a competition table are included in the manual that FIRST supplies with your challenge material. Decide how you are going to handle the competition table and mat. You have several choices.

- Simply place the competition mat on the floor. You will not have the black wood border to rely on.
- Place the mat on the floor and surround it with wood pieces to make the border. The fact that they are not connected to each other minimizes the storage requirements.
- Build a limited table with the sheet of plywood, the bottom stiffeners, and the wood borders. Place the table on saw horses, on an existing table, or simply on the floor.
- Build a true table by adding legs to the limited table.
- Add the optional florescent lighting system to approach the competition tables. See section 17.8

### **10.6 Subject Matter Experts**

Use this time to try to find people that would be willing to come in and talk to your team. These people should have some kind of expert knowledge in the theme of the challenge. For the Arctic Impact theme, I found someone who had been to both the North and the South poles to come talk about what it was like. It is great to have someone with first hand knowledge to share with the team.

### **10.7 Mentors**

You can also use this time to try to find mentors. Find some people that might have valuable skills that can meet with your team at times. Mentors can help you teach software, mechanism, or even

team dynamics. FIRST team members can make good mentors. Ask your sponsor to look for you as well.

## **11 The First Few Meetings**

After you have selected your team you are ready to start. Be ready for the first meetings. There is a lot to do!

Hand out a folder to each team member. Have them decorate the folder and add their names. Each team member should bring their folder to meetings. It will be used to hold important papers.

Learn each others names. This should be done at the start of each meeting until they all know each other. I was the last to learn all the names!

Hand out the Mindstorms kit and tackle box. Let them sort the kit as they see fit. Now is the time to go over your expectations as to how they will keep the kit in good order!

Hand out the challenge kit and its instructions. Let them choose components in teams and start to build the parts. When done, have other team members look them over to make sure they are built properly! This is a good time to talk about quality control!

Introduce the challenge. Hand each team member a paper copy of the challenge, a list of the awards they could win, and any other information from the FLL web page you think they should have. Roll out the challenge mat and place the challenge pieces into position. Talk about the different tasks. They will be eager to come up with a solution. Let them, but explain that they will need to do some learning before they can implement it!

### **11.1 Gracious Professionalism**

Introduce the key FLL phrase “Gracious professionalism”. It means that team members treat each other with respect. It also means treating an opposing team member with the same respect. It is the only acceptable behavior for each and every team member, coach, mentor, parent, and volunteer.

Gracious Professionalism includes but is not limited to:

- Respect for the feelings, opinions and lifestyles of others team members
- No physical pushing, shoving, or other physical displays.
- Respect for equipment
- Best behavior at all times
- Good sportsmanship
- A positive attitude
- A clean vocabulary
- Being friendly and polite at all times to all persons

If you don't think what you are about to do is in the best interest of the team, then don't do it!

## 11.2 Team rules

Introduce your rules early, keep them simple, and follow them to the letter. Have the team determine what the punishments should be for the more minor infractions. Here are my top rules:

- Gracious Professionalism.
- Respect the LEGO and work on the floor with it.
- Do not mix parts from different kits.
- Everyone helps to set up and clean up.
- You do not leave the building for any reason.
- You do not leave for home until you tell me.

## 11.3 KISS

Introduce KISS-ing to your team. The acronym not the action! KISS stands for “Keep It Simple Stupid”, or perhaps “Keep It Simple Silly”. In the engineering world, simple solutions are much more desirable than complex ones. The complex solution has many more places to fail, is more difficult to repair, it costs more, and its operation is less intuitive.

Consider the fate of the high tech electric potato peeler that was all the rage one Christmas. How many are still in use? Why did they fade away even if they were faster than a normal peeler? Was it perhaps the bothersome cord, the difficulty in cleaning, the big fat handle, or perhaps the motors that keep burning out?

Does this mean all high tech devices are not KISS? Of course not. Consider microwave popcorn. It's certainly much higher tech than kernels and oil in a pan, but it's much simpler.

Students are often drawn to the complex “Rube Goldberg” (see <http://www.rube-goldberg.com>) type solutions. Keep reinforcing the KISS principle. Keep asking them if their solution is as simple as possible. Have them distill their ideas down. Keep your team KISSING!

*“Simplicity is the ultimate sophistication.” -- Leonardo da Vinci*

## 11.4 Photo/videography

Before taking any photos or videos, check with your sponsor. They may have a form that the parents need to sign to authorize photography. If not, ask the parents yourself. There are some very good legitimate reasons for a parent to not want a child's picture to make the front page of a newspaper or even your scrapbook. You must honor that decision.

Pictures are great to have. Not just to look at when you are old and gray, but to use during the competition itself. Pictures can be used in a scrapbook, on posters, or part of a button. The ideal solution is a digital camera. Take as many pictures as you want and only print those you/they like. If you are comfortable with your team, assign one member to be the official photographer. Let them document the progress with pictures and maintain the team's scrapbook. If this is not appropriate, charge one of your adult helpers.

Videos are great as well. They are less easy to use on a poster, but they let the kids see themselves as a judge would. Videotape their Research presentation and show it to them. They can then see the problem areas and correct them.

At the end of the year consider making a copy for each team member of the videos and pictures. If you have a CDR, consider placing all the pictures and other electronic information on a CD for each team member.

## 11.5 Clean Up

Clean up, of course, is the least favorite part of a meeting. Who wants to clean up? Who wants to stop working on the robot? Who wants to go home and take a bath or go to bed? My first few attempts at clean up were quite humorous. It reminded me of young kids playing soccer (every player from both teams going after the one ball). I realized quickly that the problem was mine to solve. I simply assigned each student a clean up job. When that team member's job was done, they had to return to the table and take a seat. That allowed me to see the progress, I could send out reinforcements, and they would not distract each other!

The job list was:

Name	Jobs
Bashful	Restore chairs and tables, general cleanup, turn off lights.
Sneezy	Pack Green bucket. Check contents list.
Sleepy	Pack Blue bucket. Check contents list.
Happy Grumpy	Pack Mindstorms kits. Robots in bottom of kit or in blue or green buckets.
Doc	Restore table and pack white container. Tell adult helpers when you are done!
Dopey	Return towers to hardware team members,
Ms. White	Save programs,
Charming	Turn off computers, recover printouts, ...
Adult	When Doc is done with table, move it to the back hall.
Coach	Pack his briefcase. Scream. Run around in circles. Misbehave.

The first ones to get finished were placed on FOD sweep detail. In the military, FOD stands for Foreign Object Damage. A FOD sweep is conducted each morning on an aircraft carrier flight deck looking for trash (foreign objects) that can damage aircraft. My FOD sweepers would get down on their hands and knees and sweep the floor with their hands looking for missing LEGO pieces. We found lots of parts this way!

A good idea is also to keep something that the team likes to do in reserve for after cleanup. My team loved its cheer and physical stunt. If there was time after cleanup, I had them run from place to place in the school to perform the cheer and stunt. The parents probably hated me winding up their kids at night, but I got the nasty job done.

Whatever you do, do not clean up after them. It's their mess. This is a life skill that must be learned. It also impresses the judges to see the kids cleanup after themselves.

## 12 Coaching

There is no one way to coach a team. All teams will have their own unique style. This is a good thing. Too many organizations take conformity to the extreme. FLL is an attempt to think outside the box. Do what makes sense for you and your team. With that said, certain guidelines must be considered.



As much as you might like to, the robot is to be designed by the team members, not by the coach and/or mentors! If you find yourself pushing a solution you are doing your team a disservice. Not only are they not thinking for themselves, but also you may be suppressing the truly revolutionary "out of the box" idea that is the hallmark of FLL.

Does this mean you should stand idly by while your team struggles with the challenge? Absolutely not. As the coach, you must be involved. You just cannot be involved in an overtly direct mode. Telling your team to build a gearbox using a worm gear to do some task is far different than asking them how they could make something go slower.

As a coach you run the fine line between letting your team become so frustrated that they give up, and providing too much help that they simply become your worker drones.

*"Never tell people how to do things. Tell them what to do, and they will surprise you with their ingenuity." - George S. Patton*

## **13 Teaching Phase**

After the first few meetings, you will want the team to enter a teaching/learning phase. Try to put the challenge out of your mind during this time. Try to teach a whole range of skills not just the ones you think are needed to solve the challenge.

### **13.1 Teaching Hard Skills**

As students, we did not learn division by trial and error. A teacher explained the process to us. We then spent the next two weeks doing sample problems to burn the process into our minds. At the end of the two weeks we took a test.

An FLL team works the same way. The coach can and should teach skills. The coach can also expect the team to practice the skill in some way. Just remember that, like the math test you took in class, the teacher did not give you the answers!

Do not worry if you do not fully understand some skill or aspect of the challenge. You can muddle through it just as well as the team members.

*"Creative minds have always been known to survive any kind of bad training." - Anna Freud*

#### **13.1.1 RCX**

For coaches born before 1965, the concept of a computer was foreign to us as we grew up. We learned about computers gradually as they themselves were developed. On the other hand, FLL students were born after the PC and even Microsoft Windows were introduced! They live in a world that is largely controlled by computers. Something as simple as a toaster is now likely to have a tiny computer in it. And yet, it's the rare FLL team member that has a good understanding about the guts of a computer. They can play the games, program the VCR, and teach us new tricks in Word, but they do not understand the fundamental principles. It will be your job to teach them about the centerpiece part of the Mindstorms kit; the RCX.

The RCX is technically not a computer, but rather a microcontroller. A computer is a general purpose device that would have a keyboard, monitor, and mouse. You could play games or store recipes on a computer. A microcontroller on the other hand is an electronic device that is dedicated

to some specific task; a task that requires control and sensing. Your microwave oven has a microcontroller in it. It is used to read the time and power entries you make and then to control the magnetron that generates the microwaves. It also senses when you open the door and how hot your food is if you have a temperature probe. The microcontroller does all these things well and fast. Just do not ask it to run a spreadsheet!

The RCX uses a Hitachi H8 microcontroller. This microcontroller can control and sense external devices, it has computational capability, it can be programmed, and it has memory to remember its program.

What you need to convey to your team is that the RCX can control (talk), sense (listen), and execute instructions. It really has no 'thinking' capability. Thinking requires some degree of consciousness that the RCX does not have. What it can do is follow instructions really well.

When an RCX 'talks', it talks to either a motor or a lamp. It can make the motor run forwards, backwards, or turn off. It can also drive the motors at various power levels. When talking to a lamp, forwards and backwards are meaningless, but the power level will translate to brightness.

When an RCX 'listens', it can listen to a touch, light, or rotation sensor. Other sensors are available (temperature) but are not sanctioned for FLL. Each of these sensors speaks a different language to the RCX. The RCX needs to be told what sensor is connected to which port. It can then listen with the right language interpreter. In RIS, the connection between sensor and port is made from the checkboxes on the various commands. In RoboLab, the association is made by tying the proper modifier port number to the command in question.

### 13.1.2 Programming

First you must choose the software package you will use to program the robot. Please see section 17.1.

The following paragraphs will mostly be geared towards RoboLab. For my team we selected RoboLab simply because RIS could not run on the school's Windows NT computers. Because of the emphasis on RoboLab, not everything will be appropriate to RIS. Sorry!

Until I stumbled on the following idea, I had trouble teaching the basics of programming. What I needed was a demonstration and that led to the 'human robot'. Select a team member to be the *processor*, one to be the *program*, and two to be the *motors*. Write a program, one instruction at a time, in plain English on index cards. Give the stack to the *program*. Ask the *processor* to ask the *program* what the next instruction is. The *program* whispers it to the *processor*. The *processor* must then take whatever action is needed. This might be to tell the left or right *motor* to go forwards or backwards. The *processor* must then ask the *program* for the next instruction.

To work well, the *motors* must not take any action unless the processor tells them directly to do something. The *motors* must not stop doing what they were told to do until the processor tells them otherwise! The players must also link arms to form the human robot. When a *motor* is turned on, that person starts to walk forward/backwards pulling the *processor* and the *program* along. Ask that the *motors* take small steps every few seconds. High speed robots are not welcome!

As you introduce new ideas and commands, add new index cards to support them.

When you get to sensors, add team members to be the *sensors*. The rule here is that a *sensor* may not talk to the *processor* unless spoken to! The *processor* must always query a *sensor*. The

exception to this rule is when using the sensor watchers in RIS. In this case the *sensor* must inform the *processor* of an event.

RoboLab makes teaching simple as it provides four levels of increasing complexity. Each increasing level introduces more complex commands. Start with the lowest level (1) and introduce the commands using the “human robot”. Make the ‘human robot’ perform some task using the new commands. Now give each team member a printed sheet with the new icons (see appendix D). For the first few times, I also gave them a sample program for them to enter. I would work with a few team members at a time to show them how to add, delete, and connect the icons on the page.

After a few meetings at level 1, I introduced the level 2 commands. I gave them a printout of the new commands, and a mini-challenge to solve. I used the same procedure to introduce the level 3 and 4 commands.

After they all had been exposed to level 4, it was clear some team members were happier programming than others. This became the natural split for software and hardware sub-team members. I did make sure that the hardware members where given the chance (or even forced kicking and screaming) to do some programming. I wanted every team member to be able to answer any question a judge might have had.

Make sure that the software team members understand the importance of keeping several versions of the program under development. They should also understand the importance of adding comments to their code. A judge will be impressed by comments that at least comment major sections.

As coach, make sure that you back up the computer(s) that the team is using. There could be nothing worse then losing precious work due to a hardware failure or human error!

### **13.1.3 Locomotion**

The FLL challenge requires that a robot move from place to place to perform some kind of task. Getting the robot to move from place to place is very important. There are numerous solutions to the locomotion problem. By far the most popular is the tank drive. One motor is used to drive a wheel or track on each side of the robot. A caster completes the system. If both motors are engaged the robot moves forwards or backwards. Turns are accomplished by stopping or even reversing the direction of a motor on just one side. Such a robot can pivot in place.

The main drawback to tank drive is that it is sometimes very difficult to get a robot to move perfectly straight. If one motor is a bit stronger the robot will tend to hook to the right or the left. Even if the motors are perfectly matched, uneven friction or weight distribution can also cause a hook.

If the robot hooks, first examine the drive train. Look to insure all LEGO parts are snapped tightly together. Next adjust bushings and gears to insure that each side is the same. A fingernail width between a bushing or gear and its bearing is about ideal.

If this does not correct the hook, the motors may not be matched. Try swapping motors until you find two that minimize the hook. If this still does not correct the problem, apply a little friction to the side that is moving faster. Tightening a bushing is a simple way to add a touch of friction. If this does not work, you can try to drive the faster motor at reduced power using the appropriate software command.

Better yet, you can design in a few features that will help the robot drive straight. The first is to drag the caster behind you rather than to push it ahead. The forces that the caster imparts on the robot act like the rudder on an airplane or the sail on a windvane.

The second is to make the robot wider. The farther apart the drive wheels are, the less the robot will turn even if the motors are missed matched. This is only of limited use as the robot must still fit in the base and travel around a crowded table.

The third is to experiment with the type of caster. Many teams use the round 2x2 'button' as a caster. Other teams will use a swiveling wheel. These solutions all have the problem that they do not impart any directional stability to the robot. A possible solution is to use a fixed wheel or wheels. The stability that a fixed wheel provides is significant. In fact, it might not be possible to turn the robot at all! A good compromise is to remove the rubber tire from a wheel or to use a pulley. The hard plastic wheel provides some stability but not so much as to prevent a turn.

A final possibility to correct a hooking robot is to add a limited slip linkage. A limited slip linkage is a way to loosely couple the drive wheels together. In straight forward or backward operation, the drive wheels act as one being driven by two motors. Only when one motor is stopped or reversed do the drive wheels act independently. The trick is a mechanical device that normally ties the wheels together but can slip in turning conditions. The web document "Building LEGO Robots For FIRST LEGO League" (see section 20.3) has many fine examples of such devices.

Remember if all else fails and the robot still hooks:

*"If you can't fix it, feature it" -- Last Law of Product Design*

Ackerman steering is a possible substitution to tank drive. Ackerman steering is what automobiles use. One motor propels the robot; another steers the wheels. It is a more complex system, but, done properly, will insure the robot drives straight. It does suffer when it comes to turning as it cannot spin in place.

Other drive systems exist as well like the "add/subtract" transmission. A search on the web may lead to many interesting possibilities!

Picking the best locomotion method will involve many compromises like any good engineering problem!

One problem you will find happening fairly often is an apparent reversal of a motor. In software, you will command a motor to go forward, and it will go backwards! What is worst is that using a tank drive, the RCX commands forward on both motors and the robot spins in place with one motor in forward and the other in reverse! This is a simple matter to resolve.

The RCX has no idea what motor direction will make your robot go forward or backwards. The motor/robot directions given in RIS or RoboLab are arbitrary. A solution to the problem would simply be to command one motor to forward and the other to reverse to move the robot forward. YUCK. The real solution is to change the way the motors are connected to the RCX. The connector that plugs onto the RCX (or the motor) can be connected in four different ways. Two of these ways will cause the motor to rotate clockwise for a forward command. The other two ways will cause a counterclockwise rotation. Therefore it is simply a matter of connecting the wires in the proper pattern to have the motors respond appropriately.

### 13.1.4 Odometry

Every car has a speedometer and an odometer. The speedometer tells us how fast, and the odometer tells us how far we have traveled. Odometry is the science of using an odometer to determine one's position. A blind person is very good at odometry. They have memorized a map of their world in units of steps. They can get from place to place by simply counting paces and walking in the desired direction. A Mindstorms robot is mostly blind and is a great candidate for odometry.

The foundation of odometry is being able to measure and travel known distances. A simple solution is simply to turn on the drive motors for a set amount of time. Both RIS and RoboLab have commands that force the RCX to delay a specified amount of time. The idea is to turn the motors on, delay, then turn off the motors. By adjusting the time you can adjust the distance traveled. You can perform turns in a similar way. Simply turn one motor on, delay, and then turn the motor off. See Figure 1.

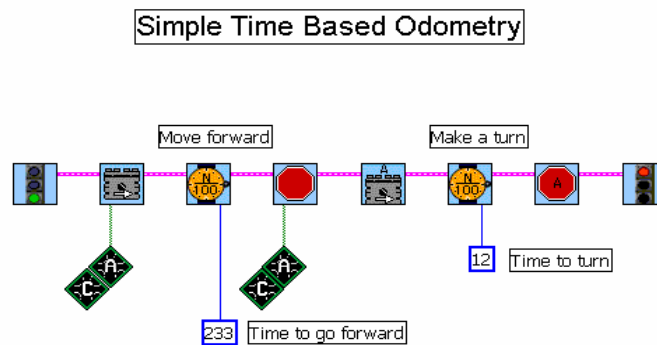


Figure 1

You can do quite elaborate motion profiles using this method, but it does have a few problems. The first is that as battery power falls, the motors run slower. Therefore in the same time duration the robot will not travel as far.

The second is that a time duration cannot be adjusted with fine precision. A robot can move a surprising distance in just a 1/100 of a second!

A solution to these problems can be found in the rotation sensor (see section 17.9). Adding a rotation sensor to the robot's drive train is exactly like the odometer in your car. It will keep track of exactly how far the robot has moved. Each full rotation of the axle that passes through the rotation sensor increases its count by 16 units. You can now program the robot to travel precise distances. In RIS or RoboLab there are ways to pause the program until the rotation count exceeds a given value. The program (see Figure 3) now is as simple as:

- Resetting the rotation counter
- Turning on the motors
- Waiting for the proper count
- Turning off the motors.

Turning a robot using a rotation sensor for odometry is rather strange. Assume you have the rotation sensor on the right wheel. To turn left you would simply stop the left and turn on the right motor. Now you wait for the rotation sensor on the right wheel to gain its count and then stop all the motors. Unfortunately the mirror of this procedure does not work for right turns!

The trouble comes when you stop the right and turn on the left motor. The robot will indeed turn, but neither will the rotation sensor since it's on the right unmoving wheel! The classic solution is either only ever make left turns, or when making right turns, stop the left motor and place the right in reverse.

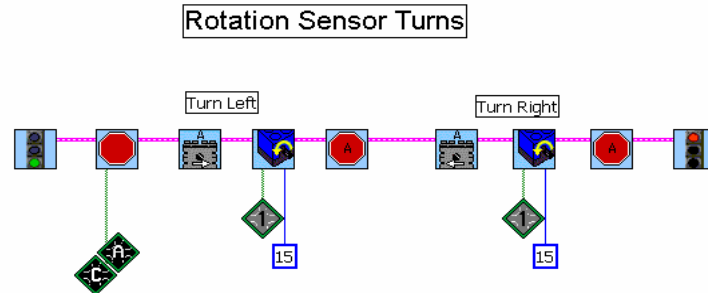


Figure 2

Rotation sensor odometry can have various degrees of accuracy. To increase the accuracy, you want the rotation sensor axle to turn at a faster rate than the wheel. This will give many more than 16 counts per wheel rotation. A gear up arrangement can provide this feature.

Be aware that the rotation sensor/RCX has a few limitations. It is not possible for it to accurately count ticks at a rate faster than about  $500 \times 16$  ticks per second. The rotation sensor will also give poor results at slow speeds. A good rule of thumb is to have the rotation sensor rotate at the same speed as the motor (about 300 RPM). For a detailed discussion, see [http://www.sjbaker.org/steve/lego/rotation\\_sensor.html](http://www.sjbaker.org/steve/lego/rotation_sensor.html).

Odometry accuracy will also suffer if you use tracks rather than wheels. The scrubbing of the tracks during turns is not very reproducible. Skinny wheels provide the best accuracy.

For RoboLab users, the rotation sensor “Wait For” icons have an interesting limitation. They themselves are implemented by a loop. The loop runs as long as the absolute value of the rotation sensor is less than the absolute value that the user asks for. Within this loop are several instructions. One to get the rotation sensor value, one to take its absolute value, one to wait  $2/100$ 's of a second, and the last to run the loop again based on a comparison of two containers. All these instructions take time and the RCX is not infinitely fast. That means that the RCX can only sample the value of the rotation sensor less than 50 times a seconds. The rotation sensor can accumulate a fair number of ticks in that amount of time! An advanced solution is to write the loop yourself and discard the delay and the absolute value instructions. The loop will run faster and you will get more repeatable and accurate results. The drawback is that you will have to worry about if the rotation sensor is counting up or down and write the loop the proper way.

I teach odometry with a few simple mini challenges (see appendix E). The first is simply to get the robot to move as close to a target as possible from a fixed starting location. We first do the exercise with time than we move to the rotation sensor. Next I introduce the need to return back to the starting point. Finally I set up an obstacle course on the floor that they must navigate.

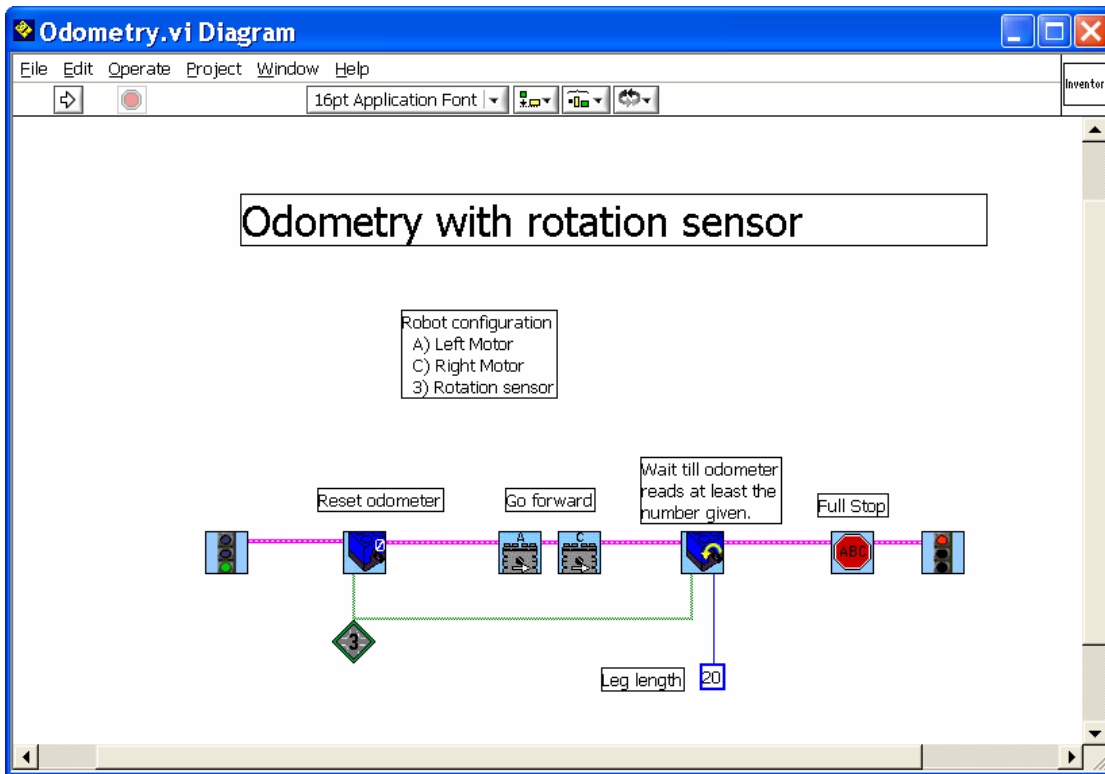


Figure 3

### 13.1.5 Line Following

Being able to have a robot follow a line on the challenge mat can be very valuable. In 2001, most of the major tasks could be reached by following lines on the mat. To follow the line, the robot needs to sense it. This is done with the light sensor. Place the light sensor just above the mat looking down. The software must now use the varying light intensities reported by the light sensor to turn the robot.

There are numerous ways to write a line following program (algorithm). Some will work better in some circumstances than others. The same algorithm executing on different robots to do the same task will operate differently. This is due to the agility of each robot and the placement of the light sensor.

Just like it is not possible to drive a car looking out the rearview mirror, the light sensor must be in front of the robot as it travels. Getting the light sensor properly positioned is also important. You must have a good variation in return values between the black line and white background. About a plate thickness away seems about right.

The simplest line following algorithm, is perhaps the “shades of gray” approach. When we look at the line to follow, we see a very sharp contrast between the white and black. The light sensor on the other hand does not. As the light sensor is moved onto the line, it first sees white, then increasing gray before seeing the final black. This is because the light sensor averages the ‘color’ over a finite area not a pin point. When the sensor is properly placed, this area is about the size of a pea.

This can be used to advantage. Look at figure 1. This shows the transition area on the right side of the line. The numbers given here are samples. Each situation will be different. Our program will

follow the right edge of the line. If the light sensor value is greater than say 43 we must turn to the left. If it is less than about 37 we turn to the right. If the value is anything else (between the two values) we will go straight. This algorithm works only as long you start on the right edge of the line. It also has a hard time following sharp turns.

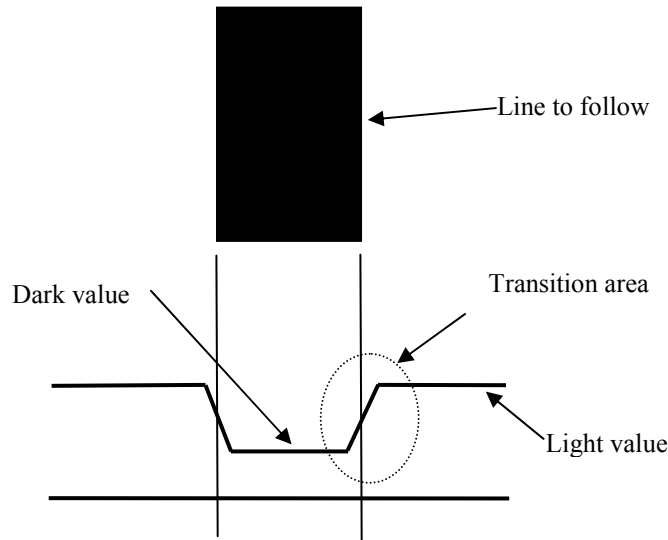


Figure 4

This is the algorithm I teach initially. I spend quite a bit of time to make sure everyone understands how and why it works. I start by drawing a thick line on the whiteboard. We brainstorm a bit on how they could follow the line. After a while, I pull out an index card with a dime sized hole in the middle of it. This I place over the white background. I then ask what the RCX sees. We write down some arbitrary large number. Now I place the hole over the black line. I ask them for a dark value and write that down.

So far so good! Now I move the hole so that a bit of the whiteboard is visible. When asked, some will say it sees white, others will say it sees dark. Now I move the hole so it's 50/50 over the line. At this point someone will start to think in terms of grey. If not, have them run it as an experiment. Use the ability of the RCX to display the light sensor value on the LCD display.

The rest of the algorithm is then fairly simple to explain. Ask them when you must turn right and when you must turn left. Draw on the board a simple flow chart for the algorithm. The tricky part is picking the right values for the "too dark" and "too light" values. My experience is that they should be about at the 1/3 points.

The team will discover that this algorithm has some failings. If they have a good understanding of why it works, they should be able to modify the algorithm to improve it substantially.

To speed you on your way, here is a concrete example. The robot is a tank drive with a front placed caster. The right motor is on port C and the left is on port A. The light sensor is out in front and is attached to port 2. Figure 5 is the program written in Robolab

Also be warned that the values you pick are dependent on the light sensor (each is unique), and the ambient lighting.



I use a mini challenge to reinforce this algorithm. I use the test oval that comes with the Mindstorms kit. They must build a robot to follow the line around. The robot that does it in the least amount of time is the winner.

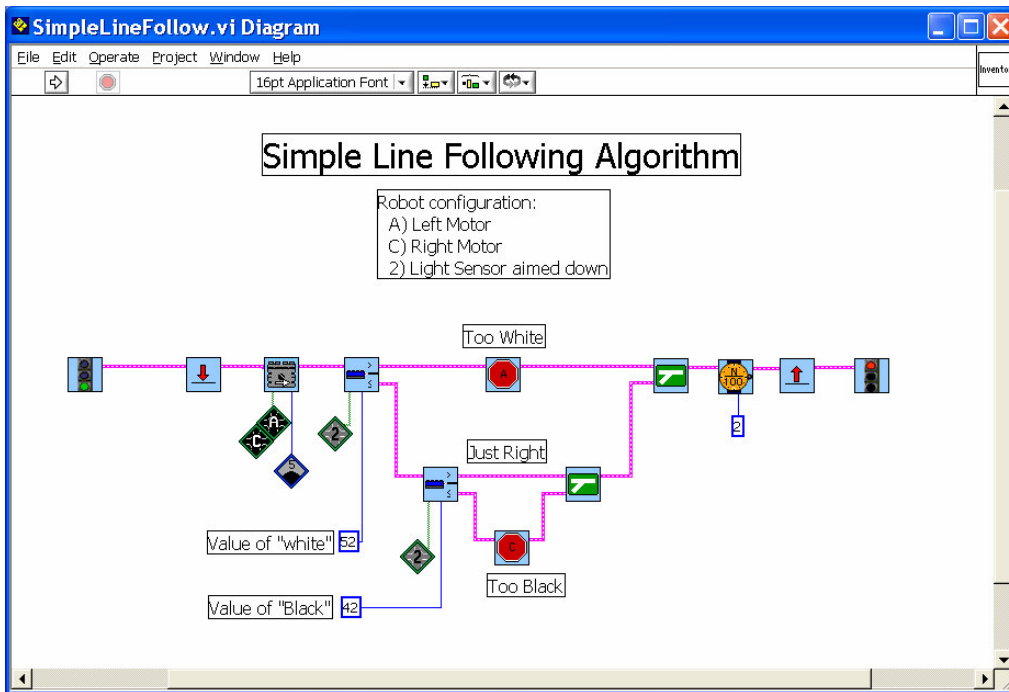


Figure 5

### 13.1.6 Landmark Navigation

Using landmarks is another navigation technique that many FLL teams will use. It is often used in conjunction with line following or odometry, but does need to be. It can be a valuable tool all in of itself.

When we drive our cars, we often use landmarks to navigate. As an example, “Get off the highway at exit 4, turn right at the Shell station, turn left at the next light, ...”. These instructions do not rely completely on distances as needed for odometry, nor do they rely completely on line/road following. They use discernable landmarks (exit 4) to perform an operation (get off).

So what in an FLL challenge could be a landmark? How might you sense it, and how might you use it?

To start with, the wood border makes a very nice landmark. Leaving the base and traveling straight until you hit the opposing wall places you in a fairly well know position. You may have drifted to the right or left, but you will know front to back where you are! To sense the wall you could use a touch sensor in a bumper arrangement, or even perhaps a light sensor. Simpler yet is not to sense it at all. If it takes 5 seconds to drive across the mat, run the motors for 6. You are bound to have hit the wall!

The wood borders can also be used to reorient yourself. After driving 8 feet across the table, you may no longer be perpendicular to the opposing wall. If your robot has a bumper on it, you can

drive against the wall and the robot will square itself to it. This works well if the bumper is as wide as the robot and the robot has fairly good traction.

An advanced technique is to register yourself in a corner. First square yourself to one wall. Then back away slightly and turn 90 degrees. Now proceed forward parallel to the wall till you hit the other perpendicular wall. Now square yourself to this wall. At this point, you know exactly (almost) where you are and can use that to then get back to base or continue on the mission.

In a similar way, the LEGO challenge field pieces can often be used as landmarks. Pushing up against them or having a touch sensor bumper can tell you when you have arrived at some location.

Other landmarks are the black lines on the mat. Rather than follow the lines, you can drive straight till you see a black line, or perhaps till you see the white background. You can also apply the same concept for turns. Turn right/left till you see light/dark are very valuable landmark commands.

When dealing with black lines, intersections can be valuable landmarks. If you use one light sensor to follow the line and a second off to the side, you can see the intersections pass by.

### 13.1.7 Manipulators

Besides moving from place to place, an FLL robot will have to manipulate objects. Manipulation is perhaps the hardest aspect of the FLL challenge. What looks so simple to us humans can be horrendously difficult to a robot. In the early stages of the challenge I will often hear the team say “We will simply pick up the gizmo and zoom over there and dump it”. Reality sets in soon after the first five “pick it up” ideas fail.

If you distill down the challenge, you are likely to find that you will need several types of manipulation. These might include lifting, dropping, dumping, pushing, dragging, and other actions. These actions are likely to be against dissimilar objects. One manipulator is unlikely to solve all the missions (but would probably be ideal).

Assuming you have used two motors for locomotion, this leaves you with only one for manipulation! How can you solve all the missions with only one manipulation motor? There are several answers.

- You do not have to solve all the missions. It might be better to be able to solve a few of the high scoring missions reliably than all of them 20 percent of the time. Here is a chance to teach some simple probability or game theory! See section 13.2.2.1 for details.
- Not all manipulators will need an electric motor for an actuator. I have seen some deviously simple designs that used purely mechanical devices. Think in terms of a mouse trap or other trigger activated devices.
- Consider a generic motor assembly that various manipulators can be connected to as needed.
- Modify the robot while it is inside the base. I have seen a team rebuild its detachable manipulator during the execution of another mission. I have also seen a modular robot be **radically** reconfigured in the base. A little stressful for my taste, but a workable solution.

The biggest problem with manipulators is the student's lack of experience with mechanisms. Here are some quick ways to get them up to speed.

- Many of the LEGO Mindstorms books have lots of great gizmos to use.

- Search the web for LEGO creations.
- Make a field trip to Home Depot to look at their forklift trucks and scissor jack platforms. If you call ahead, they may give you a demo at a **safe** distance.
- While you are at Home Depot, look for other gizmos to inspect. Wander the aisles with an open mind but your hands safely in your pockets!
- Look for inspiration at cranes, dump trucks, back hoes, and front loaders. If you meet in a library, look through all the books that deal with machines. If not, check out a few of the better ones and bring them in.
- Look at the building instructions for the previous year's field models. These include lots of great ideas.

Remember that engineering is the process of 'gathering' from the best current designs and combining them into your own! There is no shame in building on other peoples work. If this were the case, we would still be huddled in caves fending off the wolves.

After they have researched for a while, brainstorm. Prototype the selected ideas. Do not waste time early trying to get a perfect working model. What you learn from the quick and dirty prototype may completely change your approach! Try to get multiple sub-teams working on various solutions. Internal competition can be an effective expedient.

One technical point to remember is weight. Try to keep the weight of the manipulators to a minimum. Large heavy accessories cause the robot to bog down, eat batteries, and its navigation become less repeatable. Consider building long/tall devices out of axles rather than bricks. Pitsco-LEGO\_Dacta sells bags of axles and connector doodads that are great for this purpose.

Also remember that you may leave manipulators/accessories on the field. They do not have to be attached to your robot and/or returned to the base. Pushing something out on the field and leaving it there is a time honored solution.

Have the team look hard for the simple solutions. These will be the ones that more often than not will win a tournament. This is the perfect opportunity for lots of KISS-ing!

### **13.1.8 The LEGO Modulus**

Modulus is a fancy word for the common denominator we learned in math. It refers to the smallest dimension that all parts are a factor of. And now for the English translations! LEGO was designed so all the parts would interconnect in endless ways. Consider that there are 24 different ways to put two 2x4 bricks together! The way this works is to have all the parts based on the standard dimension between two studs (the modulus).

"The Art of LEGO Design", written by Fred Martin, is a wonderful web document on the LEGO modulus. After reading it, you can demonstrate some of the non-obvious connections to your team.

### **13.1.9 Gearing**

The best way to teach gears and pulleys is by demonstration. Build a few different gear boxes. Have a hand crank on each shaft. Let them experiment to determine the relationship between speed decrease and torque increase. A very nice web page exists that describes LEGO gearing. Find it at

[www.owlnet.rice.edu/~elec201/Book/legos.html](http://www.owlnet.rice.edu/~elec201/Book/legos.html). This address is case sensitive to the right of the tilde (~).

LEGO makes a special gear box part. It allows a worm gear to mesh with a 24 tooth gear in a very simple arrangement. This part does not come in the retail or the FLL Mindstorms kit. It can however be purchased from Pitsco-LEGO-Dacta. (see section 17.10.1). I cannot recommend enough the purchase of at least one per kit.

### **13.1.10 Ganging of Sensors**

It is possible to attach more than one sensor to an RCX input port. If you connect both switches to a port, the port will read 'closed/touched' if either switch is pressed.

A touch sensor can also be ganged with a light sensor. As long as the touch sensor is not pressed, the returned value will be the true light sensor reading. If the touch sensor is pressed, the returned value from the 'light sensor' will be 100. Since the light sensor will never return such a large number (unless pointed at the sun), you can tell the difference between a press and a light value. When ganging a light and touch sensor, tell the software that you have a light sensor.

You cannot gang another sensor with the rotation sensor. Neither can you effectively gang two light sensors.

### **13.1.11 Mini Challenges**

To help you get going, I have included a series of mini challenges that can be used during the teaching phase. These were designed to be similar but not identical to standard FLL missions. This way I could teach the skill without fear of teaching the specifics of how to solve a given mission. You will find these in Appendix E.

## **13.2 Teaching Soft Skills**

### **13.2.1 Brainstorming**

Lots of very good information can be found in the FLL Team Manual. I was pleasantly surprised to find that the school had already taught this skill.

### **13.2.2 Analyzing the Challenge**

Each year provides a new challenge. Each challenge has a theme associated with it. In past years the themes have been cities (2002), the arctic (2001), volcanoes (2000), and the Space Station (1999). Within the theme, the creative minds at FLL come up with a series of tasks for your team's robot to perform. Each task has its own score. Each task has its own unique problems and degree of difficulty. It's up to your team to accomplish as many of the tasks (missions) as possible in the allotted time period.

Deciding what tasks to perform and in what order will be a source of much consternation. Here are some ideas to help your team.

#### **13.2.2.1 Game Theory**

A fellow coach took this opportunity to discuss some simple game theory. He showed that it is more effective to have a robot complete just a few missions reliably than all of them sporadically. He had the team assign success probabilities to each mission. The product of the mission's score times the

probability gives the average score you would expect for that mission. Add up the expected results for each mission and you get your expected total score. Put this into an Excel spreadsheet and you can quickly determine what gives you the most “bang for the buck”.

### 13.2.2.2 Spreadsheet

Get a large sheet of paper and turn it into a spreadsheet. The rows are the different missions. The columns are different attributes. The intersections (cells) have information pertaining to the two.

Here is a small part of our 2001 challenge spreadsheet. Dumper refers to a motor driven dump truck device.

	Tracks	Wheels	Difficulty	Dumper
Medical Barrel	Ok	Ok	Med	Use dumper to push down on spike.
Save Scientist	Ok	Ok	Low	Place Scientists in dumper.
Balloon	Yes	Hard to drive on ice with wheels	Low	NA
Ice Core	Yes	Yes	High	Add a crane to dumper?

Interestingly, the team rarely referred back to the spreadsheet. It was the process of filling it out that was valuable.

### 13.2.2.3 Index Cards

My family thinks I am crazy, but I firmly believe that with enough index cards you can rule the world. They are incredibly useful for so many different tasks. The team used index cards to sequence the tasks during a round. Each card represented a different task. On the top of the card were the task name and the points it was worth. The rest of the card carried important information like how long they took, or special equipment needed. We also added cards like “return to base” and “Yank back to base –X points”. By simply shuffling the cards they could try out different ideas. It did not take them long to create a fairly effective strategy.

## 14 Solution Phase

At some point you will transition out of the teaching phase. The meetings will now consist of a business meeting, time to work on the missions and Research Assignment. Make sure to keep the schedule updated so that all members understand the importance of keeping focused during the meeting.

### 14.1 Sub Teams

A large team of 10 students is very hard to handle. Breaking the team into groups will probably become necessary. During training, splitting the team into sub teams can be very helpful. One team

can work with a coach/mentor on mechanics while the other is learning software. Further splitting is possible if you have access to more than one Mindstorms Kit. Rearranging the composition of these teams from meeting to meeting can also let you see and account for personal dynamics.

When initial training is done, team members will decide for themselves that they want to pursue software, building, research, or miscellaneous career. For the most part this is fine to let them self-pick. Do watch out for the student that might be pushed out of doing what they really want to do.

After your initial brainstorming sessions, the whole team may have several top candidates. If you have access to multiple Mindstorms kits, you can divide the team down further into mini teams. Each mini team can make a prototype during the next few meetings. This allows them to try out multiple ideas in a shorter amount of time. They can then take the better of the two, or more likely, take the best parts of each to make the final robot.

At one stage during the development cycle, the team simply had too few jobs and too many team members. I was forever trying to find stuff to do to keep the others occupied. When I did find things for them to do, they needed my attention or supervision! Had I had a third adult, I could have had the "loose end kids" work on the team poster, song, research or any number of other products.

As it was, I finally gave up and downsized the team for a few meetings. The other members were very important to the team, but were simply a distraction at that point. As a coach, make it clear that there may be times when you may ask some members not to come to a meeting. It is not a punishment, but a way to allow the team to succeed.

## **14.2 Dress rehearsal**

If possible, find a group of adults your team can use as a dress rehearsal before the competition. This will work out some of their fear and jitters. This will be a large benefit to them when they are up against the real judges. If your sponsor is a business, try to let them make the presentation to interested parties at a brown bag lunch. If it's a school, try to get scheduled at a PTO/PTA meeting, look to a parent's company, or some community group like the Elks. Any group of "unfamiliar grownups" will do.

Video tape the dress rehearsal so that the team can watch it later. When reviewing the tape, do not allow the team members to ridicule or mock each other. Here is a good opportunity to discuss "Gracious Professionalism"! What they can discuss are team roles, places to save time or labor, and team appearance.

## **15 The Research Assignment**

The Research Assignment (formerly known as the Hypothesis) is the second main product of your team. The essence of this part of the challenge is a research project. The team must research something and create a report of some kind to present to the judges. The form of the presentation has always been very open. A simple reading, a skit, a PowerPoint presentation, a video, or some other way to present the information are all possible. You are normally given 5 minutes to present to the judges. The length of the presentation may vary depending upon the tournament you are attending. Contact the tournament director for specifics.

If you need special equipment, call the organizers ahead of time. They may already have the equipment you need. If not, make sure to bring everything you need. Also consider that you will not

have a large amount of time to setup. Make sure that the team members can perform the setup and teardown without a lot of fuss and adult help.

The rules and subject for the Research Assignment change each year. Read the description carefully and read the forums at the FLL web page.

## **16 Competition Day**

So your team has built a robot and has its Research Assignment presentation. You are registered for a competition. Now what?

Good question! The competition is the high point of the FLL season. The team will be very focused on this event. You will need to prepare them for it. Nothing can ruin a day faster than bad expectations!

No two competitions are the same. FIRST has a guide/rule book for how to run a local and state competition, but each tournament director has lots of flexibility to tailor the event to their style, capabilities, and facility. What follows is a description of the mythical "typical" competition. Variations are described where appropriate. Your competition will be similar but not identical.

### **16.1 Travel**

If you need to travel to a competition, make sure you follow your school or other sponsor's procedures. They will want to make sure each driver is properly insured. There is often paper work to be filled out. As a coach, make sure that you do not take personal charge of any team members (except perhaps your own child!). There will be too much going on at the competition to worry about other kids.

FLL does have a parent involvement policy. They would really like all parents to attend the competition. Try your hardest to get them all to come.

### **16.2 Setup**

You will have been told what time to register your team. In my experience, arrive as early as possible. Registration is a coach's job. They will have paper work to fill out and most likely a form to sign accepting that your photo/videograph may be made public. They will give you a location to set up your pits and a schedule for your team.

Get your team to set up your pit area. You will not have much space, maybe as little as one table. Try not to bring more than you really need. If possible leave coats and other personal belongings in your car or away from your pits. Make sure your team poster is visible. Set up your computer and try to have a slide show of pictures or a simple picture as wallpaper on your computer.

You may need to bring a good extension cord (heavy duty outdoor type) and a power strip. Most arenas will not have adequate outlets. Some venues will have no power at all other than a few scattered laptop recharging stations. Plan accordingly.

Review the schedule with the students. The competition schedule is VERY tight. If you are not ready at the time given, you may very well miss your round. The judges may have runners to corral the teams in advance of their round. The runners do not know your team or how to find you. They only know where your pit is located. Unless you are competing or being judged, you must have one

team member at your pit area who knows where the rest are located. A pair of Family Radio Service (FRS) walkie-talkies are a good idea! If the team is at a scheduled event, leave a note at your table to that effect.

If the tournament is not using runners, it will be up to you to make sure you arrive at the right places at the right times.

The other reason for leaving a member at the pits is that you never know when one of the walking judges will stop by for an interview! If they can't find you, it makes their job impossible.

At the same time having your team hover at your pits the whole time is not encouraged either! Leave some room at the pits to allow visitors to see your poster and other display items. Have your team stay together and visit the other teams or simply stay out of the way in a corner practicing their cheer!

As coach you will be concentrating on getting to events on time. Give the responsibility to your volunteers to keep your team together. You will have too much to do besides keeping track of who went to the bathroom!

Once you are settled, get your team to test their robot. The arena should have a test table available for use. They should also allow a short visit to the competition tables. If you need to take light sensor values, now is your chance.

### **16.3 Opening Ceremonies/Coaches' Meeting**

At some appointed time, the tournament director will start the opening ceremony. This can be as simple as the playing of the National Anthem or as exciting as the 'Parade of Nations' at the Olympics. Usually at this point, the coaches will be asked to attend a short meeting. They will provide last minute instructions and take your questions. Now is the time to turn in any printed correspondence you have had with FIRST on a special rule interpretation. You might also be allowed a short visit to 'approve' the competition tables. Use this time to look for anything that is not to spec that could hurt your team's chances. Some things to look for are gouges in the wood border, dust/dirt on the mat, mat not in proper place, wrinkles in the mat, models not made properly, and models not in proper positions. I bring a roll of double sided carpet tape to fix wrinkles and black electricians tape to smooth over gouges!

### **16.4 Rounds**

At this point you will descend into organized chaos. Nothing will go quite right! Stay loose and stay close to your pits. The organizers, as good as they are, will stumble on problems that will re-arrange the whole day. Do not complain. Do not allow your team to complain. Be as helpful as possible. Compliment the volunteers at every opportunity. They are volunteers (or draftees) just like you!

During the day you will get normally get at least three rounds at the competition tables against other teams. The organizers may pit you against the same team each round. If they do, they are also likely to schedule you in the same order. It's nice to mix up the pairings, but a massive headache to schedule (from one who knows!). A round or two before your match, a runner may come to the pits to take you to a staging area. Now is a good time to review your strategy for the competition and to practice your cheer.



When the time comes, your runner will walk you to your competition table. Have your two operators move to the table while you get your team settled in the team seating/standing area. Your operators should now follow the table referee's direction. They should also scan the table to make sure it is properly setup. Once the match starts, it is too late to change the table.

When the competition starts make sure you cheer! If possible, have the two operators witness the referee's scoring of the table. If a mistake is made, now is the only chance to correct it. When done, collect all your parts and leave all the table pieces behind. Go back to your pits and review your match. Let everyone know what is coming up next.

At the conclusion of the first three rounds, the judges will select the top teams to compete in an elimination tournament. Normally the first rounds are single elimination followed by a best of three in the final match, but this varies between tournaments. If you have made it this far, you are in for a whirlwind event. At this point, you will not have a schedule for when your next round will be. Stay glued to your pit and be ready at a moments notice. As you survive each elimination, the following one comes sooner. At some point (hopefully), you may not even get back to your pits, so be prepared with all the items you may need.

Having made it to the finals, I know that by the time you have run all the rounds, performed for the technical judges, made and tested any software changes, your batteries may be getting low. During the day, take every opportunity to replace your batteries. But you may find yourself in the situation (a good one!) where you are winning and competing back to back. At some point it's my belief that you are entitled to ask for a short break to replace your batteries. The idea is the best robot wins, not the one who had the fewest matches to run. When you replace the batteries do it fast so that you do not lose the programming. See section 17.2.

Be prepared to have your team shake hands with the opposing team. This is not normally done for the initial rounds but will be expected in the final rounds.

## **16.5 Food**

Food at the competition may be problematic. If any is made available, it is likely to be hot dogs, hamburger, and or pizza. Hopefully, you will have received information in advance from the tournament director telling you what is available. Make sure your parents get this information as many students have dietary restrictions that must be accounted for. If you did not get any information, call ahead.

Also make sure to bring plenty of snacks. Bring good wholesome snacks that will help the kids through a tough physical and emotional day. Since tournaments often start early and run late, make sure to bring more than what you think you will need.

*"An army marches on its stomach" --Napoléon Bonaparte*

## **16.6 Research Assignment**

Sometime during the day, you will be called to perform your Research Assignment presentation before the judges. It may or may not be on your schedule. The judges try to take a team that has just competed at the tables so as not to interfere with their next round. A runner will find you, or the burden may be on you to arrive at the judging area at a set time. If the tournament is using runners, make sure you have all your materials gathered. They will give you some time to gather your team,

but do not make them wait long. If they found you between rounds, check the time of your next round and inform the runner. They may elect to skip you if not enough time is available.

You will normally be taken to a quiet room and given a few minutes to set up. They may not let anybody other than your team and coaches into the room. It is disappointing to parents, but their presence can be distracting. Have your presenters introduce themselves and the team and start the presentation. After the presentation there will be a question and answer period from the judges. Make sure they thank the judges for their time. A quick team cheer may also be appropriate at this time.

Just remember that under no circumstance should an adult help in any way. The coach can teach how to bat, but it is the student that goes to the plate to swing. I avoid the issue altogether by not going in with the team.

### **16.7 Software Interview**

As the day goes on, the team will be interviewed by the software judge(s). This interview may be a scheduled event right at your pit, or it may be scheduled in a quiet room. The judges will want to see and talk to your team about the software they wrote. The questions are addressed to the team members, not the coaches, mentors, parents, or volunteers. The judges want to hear that the students did the work and understand what they did. They want to see and hear about unusual solutions or techniques. Normally an interview and you do not need any kind of presentation. Be advised that some states do require a formal presentation. Call ahead to find out.

Make sure you can show your software to the judges at a moments notice. Trying to find a floppy or waiting for a computer to boot is not helpful to the judges or to your team!

A good hint is to have a sample of your latest software printed out to give to them. Choose a section that is easy to follow but highlights some difficult problem you overcame. Adds lots of comments (you added comments as you went along right?). Make sure your team name and number is on the printout.

### **16.8 Tech Interview**

Also during the day you will have an interview with the technical judges. This interview, like the software interview, may be at your pits or in some quiet room. The judges will want to see and talk to your team about the robot they built. They will want to know about the design process, what are the best part and worst parts of the robot. They will want to know what sensors you used and for what purposes. Just like the software judges, the questions are directed to your team members!

Make sure you can demonstrate the robot. Some tournaments will have a table set up for this interview. The team will be expected to run a mission. Pick a mission that is not too simple, yet you have a good success rate on!

Consider having your team provide the judges with a one page specification sheet. Include a picture, list of sensors used, interesting features, language used, and other critical information. When it comes time for the judges to pick winners, your specification sheet can help jog their memory.

Be advised that many tournaments combine the software the tech interview. Your team may be interviewed all together on the various aspects or they may be interviewed separately but in the same room. Have your team prepared for this so that they can best divide up the team if need be.

## **16.9 Floating Judges**

The sponsors are likely to have several judges walking around just observing. They will not advertise who they are but you can bet their ears and eyes are wide open. They are charged with awarding three trophies. These are the “Against all Odds”, “Spirit”, and the discretionary “Judges” awards.

This is the time to remind your team about Gracious Professionalism. One negative comment from a team member can undo all the effort you make in decorating with balloons, matching uniforms, your precision marching, the human pyramid, and hysterical cheering. One helping hand to a team that forgot to bring something you have in abundance is worth a thousand balloons.

As a coach, the last trophy I would like to get would be the “Against all Odds” award! But I am certain a team that would get this trophy would be delighted. If you see or hear some story that would be appropriate, seek out a runner/judge and inform them. Other than to thank them or to alert them of a fire, this is the only time that you should approach the judges.

## **16.10 Spirit**

Of the many awards presented, one is based on team spirit. “Spirit” is a rather nebulous noun. The judges have little solid criteria to select a team. It could mean a team tee-shirt, a full uniform, a team song, a team cheer, a secret handshake, or a physical stunt. It could mean having a large cheering section for all the teams as they compete. It could mean something completely different. What it will mean is what catches the eye or ear of the judges that day.

Perhaps the team could form a Spirit Committee to dream up ways to show their spirit. Cub Scout web sites are a great resource for spirit related material.

In the senior division (FIRST), a very traditional (nearly mandatory) item is the classic button. Teams make up thousands of logo buttons to hand out to opposing teams. A challenge for each student is to get a button from each other team! Quite a challenge considering the national competition had over 290 competing teams!

Buttons do make an appearance at FLL competitions. This is especially true of FLL teams that have an associated FIRST team. You probably can't afford to make a few thousand buttons, but a hundred might be possible. If you are associated with a FIRST team, they may have the equipment to make buttons, or know the best place to buy them.

## **16.11 Closing/Awards Ceremonies**

During the elimination rounds, the judges are hard at work picking the winners. This is by far the hardest job of the day. Each team is a winner, and yet the judges have to select winners. As a result, this process can take a while. Part of the delay is that the judges must ‘distribute the wealth’ to some degree. FIRST would rather not have a dominant team go home with half the trophies. It is not uncommon that the best team in a category will be skipped if they have won a more prestigious award. This is especially true of the Director's Award winner.

Therefore plan to have some dead time before the closing ceremonies. This might be a good time to pack up your pits and lug all the equipment to the car. It is also a nice time to visit with the other teams and scout for new ideas to use next year. For the coach, it's probably the first time all day to grab something to eat! The organizers may have anticipated this issue and have a planned activity.

Like the opening ceremony, the closing ceremonies can be simple or elaborate. Try to keep the whole team together to show your team spirit to the very end. Cheer for each winner. If you win an award, scream even louder! It normally takes me two weeks to recover my vocal cords after a state competition!

## 17 Technical Information

### 17.1 Mindstorms Software

According to the 2002 FLL rules, you have two choices of software to program the robot. They are:

- Robot Invention System (RIS) and by extension the Mindscript language.
- RoboLab

The commercial Mindstorms Kit includes a copy of RIS. Based on the purchase date this could be version 1.0, 2.0, or 2.5. Any version is legal.

An FLL Mindstorms Kit will include a copy of RIS and well as RoboLab. The versions differ based on purchase date, but all versions are valid for competition.

The choice of what software to use is yours to make.

There are two show stopper problems.

- 1) RIS will NOT run on Windows NT or a Mac.
- 2) The USB tower will not run on Windows NT.

If you use a Mac, your ONLY choice is to use RoboLab.

If your computers run Windows NT, your ONLY choice is to use RoboLab with a serial tower. The USB tower will not work as Windows NT does not support USB even if the computer has a USB port.

If you are not running Windows NT, you can use RIS or RoboLab (version 1.5 or greater) with either the serial or the USB tower.

Having used both RIS and RoboLab, I believe that RoboLab is the better choice for an FLL team. It is only slightly less intuitive than RIS but much more powerful. RoboLab has some features that simply do not exist in RIS. With the introduction of RIS version 2.0, the lacking features are minor. It is my understanding that RIS does not provide for multiple threads of execution (multitasking) nor are it's 'IF' statements as powerful.

Does this mean you cannot win using RIS? No, I have seen teams blow away the competition using RIS. Sometimes a good KISS solution needs hardly any sophistication in the software.

Both RIS and RoboLab are graphical in nature. Users manipulate icons that represent various commands. After the program is created, it is downloaded to the RCX using an infrared tower.

The following table lists some of the pros and cons of each language.

Language	Pro	Con
<b>RIS</b>	This program is very intuitive. Its is very likely that any team member who owns a Mindstorms kit at home is already familiar with this software. More books are devoted to RIS than your other choices.	As of version 2.0, RIS cannot be made to work on Windows NT. RIS is not as capable as RoboLab. Some features of the RCX are simply not available. For a novice team, this is probably not a problem. For a veteran team, it might be a limiting factor.
<b>RoboLab</b>	Allows you to program the full capability of the RCX. You can add comments to your code. Runs on Windows 95, 98, ME, NT, and XP. Has various degrees of difficulty (levels) built in to aid teaching.	Slightly less intuitive than RIS. Not available in commercial Mindstorms kit.

## 17.2 Batteries

Batteries are the bane of an FLL team. If you are not careful, the cost of batteries will eat you alive. A fairly heavy RCX robot can eat a set of six AA alkaline batteries in as little as an hour. A simple robot might keep going for 4 hours. If you multiply this out over your meetings and competition you are looking at serious money!

On a positive note, the tower will run for months on its 9 volt battery!

After a painful learning curve, here are some things to know.

A fresh set of batteries will make the robot behave differently than a weaker set. After spending an evening adjusting time constants in the robot to get it just right, it failed miserably the following night with fresh batteries. At first we thought we had simply lost the 'magic' numbers due to poor software management. But after a few sessions, we figured this out.

Brand new alkaline batteries are 'super-charged' for about 10 minutes of use. At this point they provide the same power for a considerable time before they fall away to nothing. For competition, always replace the batteries with fresh ones after every other round. This is especially true if your robot uses time delays for making turns or moving set distances. See section on odometry.

The team should be aware of this issue and not spend lots of time getting the timing right early on. These adjustments are best left towards the end. Replace with brand new batteries and then adjust the numbers. If you run more than 6 minutes, replace the batteries! Mark the old batteries with a Sharpie marker as 'used' and keep for general use. When an alkaline finally goes dead, mark it as such.

A reasonable replacement for alkaline batteries are Nickel-Metal Hydride (NiMH) rechargeable batteries. These are quite expensive, but will pay for themselves over time. You can purchase

NiMH batteries and chargers at Radio Shack. Buying 12 AA NiMH batteries and a charger will cost about \$90.00. The parts numbers I purchased were:

Part Number	Description	Cost	Quantity	Total
23-525	4 packs of NiMH AA batteries	\$18.00	3	\$54.00
23-422	Fast charger for 8 batteries	\$35.00	1	\$35.00

NiMH batteries are no substitute for alkalines when it comes to competition. An alkaline battery has a higher voltage and more power. The NiMH batteries are great for everyday use.

Do not try to use Nickel Cadmium (Nicad) batteries. Their ability to supply the proper voltage and power is poor.

If possible find a place to recycle old batteries. Alkaline and especially NiMH batteries can pollute ground water very nicely.

Have your team learn to change the batteries. They will learn two things. The first is not to build a robot that must be dismantled to gain access to the battery compartment. The second is how to do it fast enough that you do not lose the firmware or the stored programs.

Changing batteries is a two person job. First open the RCX to expose the batteries. Next take a Sharpie and scribble over the batteries to mark them as used. One person will remove and add the batteries, the second will manage the new and old batteries.

Change the batteries one at a time. Remove the old and insert the new quickly. Now wait a few seconds before doing the next. If you leave any one battery out for longer than about 15 seconds you risk losing the firmware and programs. Waiting a few seconds allows a capacitor to charge inside the RCX that will give you the power to run for the next 15 seconds.

If you confuse the batteries during the change, do not panic. First get any battery into the RCX. Now it's a simple matter to look for the Sharpie scribble to determine the old batteries.

### **17.3 Competition Mats**

In 2001, the shipments of the FLL equipment were delayed due to 9/11. Rather than wait for delivery, I made a paper mat as best I could from the pictures on the web. It allowed the team to get started thinking even if they could not count on it. When our mat did arrive, we made some paper copies that were used by some team members at home with their own personal Mindstorms kits. These were made by simply tracing the design onto large sheets of paper taped together. Lines were filled in with black magic marker.

This greatly enhanced our ability to multitask the problem. One problem we did discover with the paper mats, was that the surface friction was different. This made odometry programs work differently on the paper versus plastic mat. This was especially true of robots that have high friction tracks.

### **17.4 Competition Table**

Instructions for making a competition table are available at the FLL web site. I would only offer the suggestion of using MDO plywood rather than standard plywood. MDO is plywood that has a paper like surface. It makes the surface very smooth and perfect for robot operation.

The mats are prone to delamination. Rolling out the mat each meeting was taking its toll on the mat. I elected to keep the mat on the table permanently. Because I had to store the table on its side, I had to figure out how to hold the mat to the table. I tried double sided tape but it peeled off in a day or two. I tried spray contact cement and the mat peeled off as well. I then used industrial contact cement for counter laminates. That was bad news. The solvents attacked the plastic and we had a real mess. I am not sure what the proper adhesive would be.

In 2002 I did not use any adhesive. I simply stapled the mat down along edges and under the stationary field models. This worked very nicely. In future years, I moved the team to my home and the table and mat were out permanently.

### **17.5 Programming garage**

Imagine the pit area at a competition. It will be some kind of large room with teams all busy downloading programs to their robots. It does not seem like a problem because humans can't see the infrared light from the towers. If you could see infrared light, you would be blinded by all the towers sending out programming sequences. Unless you protect yourself, you will be the subject of a disaster. If your robot is turned on and sitting on your table, a team 30 feet away may not only program their robot but yours as well. The solution is to bring a programming garage. A simple dark opaque box. Put your tower and the robot inside the box when programming. Make sure the robot is turned on **ONLY** when it is inside the garage. Start practicing this procedure early so that it becomes a habit. Many teams have been shocked to find their perfect robot running their competitors programs at mission time! This is of course the reason that the rules state you may not program inside the arena area.

I have heard of some competitions allowing a small strip of plastic tape over the programming port to prevent this problem. Check with the sponsors for this possibility. Also check that your tape is infrared (IR) opaque. Some of the strangest things are opaque to visible light but transparent to IR. Some black electricians tape is transparent!

New in the 2002 rules is a requirement to program at 'short range. This lowers the power of the infra-red transmitter to minimize (but not remove) this problem. This can be done in RoboLab from the Administrator button on the main page.

Remember, gracious professionalism prevents you from performing an infrared assault on your rival!

### **17.6 RIS "My Block" issues**

For RIS users My Blocks can be an important feature. Once mastered, their use can speed up software development. Due to a less than ideal implementation, My Blocks can be very confusing and painful. The following section hopefully will get you up to speed quickly. First we need a definition for Subroutine and Macro.

**Subroutine** - a subroutine is a group of commands that is run each time its name appears in a program. Subroutines save space in a program by listing the group of commands (which can be quite long) once and then referring to those commands with a one line command. This also makes the program logic easier to understand by removing repetitive details. Standard subroutines require additional time to refer to the commands, run them and then return.

**Macro** – a macro is a group of commands that are run each time its name appears in a program. Macros do not save space as the sequence of commands is placed in the program for each instance of the macro in the program. Macros are similar to subroutines in that they are a set of commands that are referenced by a single name.

My Blocks are macros. The compiler may at its choosing implement a My Block as a subroutine. The user does not have any control over this and therefore should consider the My Block as a macro. My Blocks are stored in a file called MBFile.ini and are located in the vault directory (typically c:\program files\LEGO MINDSTORMS\RIS2.0\users\''user name''\vault\). Each time the GUI interface is started all the my blocks in the MBFile.ini are loaded and available in the menu for placement in your program. A copy of the My Block is stored with your program so that the program is transferable to others without passing along the MBFile.ini.

My Blocks are used to 1) Allow modular programming and to make the code more readable by placing groups of commands inside a My Block and giving it a label. The commands inside the My Block can be hidden or expanded as needed. Using names for groups of commands allow for more readable code. 2) Allow for groups of commands to be moved and “shared” with other programs. A My Block created in one program will be stored in a common set of My Blocks and are then available to be placed in other programs (new or existing). This allows “sharing” of the My Blocks but since each program maintains a copy of the My Block within the program itself, changes in one program are not automatically picked up by other programs. A manual effort is needed to update the My Blocks in other programs.

Each time a program is loaded within the GUI interface with the pull down menu, a check is made to compare the My Blocks in the loading program with the My Blocks in the MBFile.ini. An error message pops up asking for the My Block in the loading program to be changed so that two different copies of the My Block do not exist. This check can be bypassed by loading the program from the vault outside the GUI interface. Some people prefer to bypass the check and verify the integrity of the My Blocks manually and others prefer to let RIS verify the integrity of the My Blocks each time they are loaded.

## **17.7 RoboLab Tricks**

The documentation that comes with RoboLab is fairly poor. The printed manuals are barely an introduction to the program. On the disk (as of version 2.5.1) you will find a huge reference manual. This is a nice document, but as a reference manual and not a tutorial, it only is of help to the more advanced user. For the beginner, there are some nice tutorials at the Minnesota website (see section 20.4). What follows here are some tricks that are not found in the reference manual or the tutorials.

- 1) Use spacebar to select between the pointer and thread tool.
- 3) Use the tab key to select between pointer, hand, thread, and text tools.
- 4) Press control-b to remove all the broken wires.
- 5) Press control-e to toggle between the front panel and the diagram.



- 6) When editing the icons for sub-vi you can use Paint! Open Paint and under Image-Attributes select 32\*32 pixels. Now draw your icon. When done, press control-a to select all and control-c to copy. Now jump back to RoboLab's icon editor. Click on Edit-Select all. Now press control-v to paste. Voila! This sounds hard, but Paint is a much nicer editor. This hint is likely to work with other graphical editors.
- 7) RoboLab icons can have transparent areas. Unfortunately, there is no transparent color to select from the palette. Transparency occurs when the background color is in contact with the outside border of the icon. In this case, RoboLab will then flood fill the background to a transparent mode. A good example of this is the motor power level icon(s).
- 8) The following is very dangerous! You can understand how many of RoboLab's icons actually work by looking at their implementations! Double click on an icon, now press control-e. The sub-vi for the implementation will pop up! DO NOT EDIT!!!!!! Although you certainly can copy interesting sections! The interesting ones to look at are the Wait-fors. They are implemented as busy loops with 2/100 second delay!
- 9) Unpowered light sensor (Ken Streeter)  
You can place the light sensor into unpowered mode if you upgrade to 2.5.2 and apply the patches. You will now be given new sensor adapters for an unpowered generic sensor. Running a light sensor in unpowered mode will turn off the illuminating LED making it useful for determining ambient lighting. The patches are available from:  
<http://www.ceeo.tufts.edu/RoboLabatCEEO/Resources/downloads/newsensors.htm>
- 10) Raw mode (Ken Streeter/ Skye Sweeney)  
You can place the light sensor into raw mode if you upgrade to 2.5.2 and apply the patches. You will now be given new sensor adapters for a powered generic sensor. Running a light sensor in a raw mode provides a greater range of return values. The light sensor will return numbers from 0 to 1023 rather than 0 to 100. There are some mathematical advantages to using these larger numbers. In percent mode, a one unit change is equivalent to a 102 unit change in raw mode. This makes the sensor considerably more accurate (but no more sensitive). The patches are available from:  
<http://www.ceeo.tufts.edu/RoboLabatCEEO/Resources/downloads/newsensors.htm>
- 11) A tip to save time transferring programs to your robot. (John Wanninger)

The following hint uses a little know fact about the RCX firmware. True subroutines (not just sub-vi) are persistent. That is, once defined on the RCX they do not disappear once the program has stopped. In fact, the only way to get rid of a subroutine is to 1) remove the batteries for a few minutes to clear the firmware and memory or 2) explicitly delete a subroutine using the "delete subroutine" icon. This means that once you create subroutines and load them into a program "slot" (1 of the 5 on the RCX), they can be used by any future program uploaded to that slot. Each program "slot" can have up to 8 subroutines and these subroutines are not visible to programs in other "slots".

So if you have a large subroutines or a large number (up to 8) of subroutines, simply load them once into the RCX with a vi that creates them, and then only re-load them if they change. Now you can make vi's that call subroutines that don't need any "create subroutine" functions. This can save lots of upload time.

Each of the 5 program slots requires it's own initial set of subroutines. Suppose you want to use 4 subroutines: Sub1 for Forward, Sub2 for Reverse, Sub3 for Left turn, and Sub4 for Right turn. Let's create these in a vi we'll call "DefineSubs.vi". This vi has a start(green stoplight) , 4 'create subroutine' function blocks, each with it's own associated subroutine branching off, and then an end (red stoplight). Let's pick program slot 3 on the robot, and now upload the "DefineSubs.vi" program to it. The four subroutines are now available to any program that gets loaded into program slot 3.

Now create another vi called "new.vi" containing several "run Subroutine" functions to run any or all of the subs Sub1, Sub2, Sub3 or Sub4, and upload it to program slot 3. Note that this vi does NOT need to contain the 'create subroutine' functions. Those previously created subroutines are persistent and will always be available in program slot 3. Each upload will overwrite existing subroutines and tasks with new subroutines and tasks. However, if there are no new subroutines in the upload, the existing subroutines are not overwritten and are therefore persistent.

12) Using the pointer tool, you can single click on a wire to select just that segment. Double click on the wire and you select the full connection from source to destination. Triple click on a wire and you select the entire tree (one source to many destinations). Quadruple click on a wire and your finger gets tired!

13) Commenting in RoboLab using text boxes: (Mark Beitz)

- If you right click a text box, a pop-up window opens. Uncheck the "Size to text," then drag a corner of the text box to increase the size of the text box.
- The text box inside color and the text frame can be set to colors other than white and black using the colors palette. The lower right color box controls the text box color. The upper left (and more forward) color box controls the frame color.
- Changing the text box color to 'T' for transparent and then increasing the text box size can allow a certain section of code needing a comment to have a box around the code and the comment. Look for the 'T' in the lower left hand corner after left-clicking the color box.
- If you change the text box background color to something other than transparent, the text box may cover any coding that is behind it. To correct this problem, select the text box and "Move to Back" This command will place the selected item behind other graphical elements. This 'Z axis' reorder command can be found on the toolbar to the right of Align and Distribute icons. It can also be executed using the highly obvious control-shift-J combination!

14) Motor power levels can be selected one of several ways.

- You can wire a plain numeric constant to a motor command. The RCX firmware provides for a total of 8 power levels using the numbers 0 to 7. If you wire a numeric constant use a number from 0 to 7. Interestingly, a value of 0 is not a stopped condition, but rather simply the least of the eight power levels. Motor commands will also accept a power level that is stored in a container.

- You can wire a RoboLab modifier to a motor command. RoboLab has created 5 'pre-canned' icons they call power levels 1 to 5. There is no direct correlation between the RCX (0-7) numbers and the RoboLab (1-5) numbers. The table below is the arbitrary association that RoboLab created.

RoboLab Value (1-5)	RCX Value (0-7)
1	0
2	1
3	3
4	5
5	7

15) There are several ways to place a sub-vi onto a diagram. You can:

- Drag and drop from the File Manager.
- Drag and drop from another VI's diagram
- Drag and drop from the current VI. Use CTRL to get a duplicate.

## 17.8 Lighting

One of the most used sensors in a Mindstorms robot would have to be the light sensor. This sensor can be used for a whole lot of things. One traditional use is the sensing or the following of a line on the competition mat. In the Arctic Challenge, black leads (ice cracks) could be used as roads to most of the major tasks. Pointing a light sensor downward allowed the RCX to 'see' the line. When over the ice, the RCX read a large number, over a crack a smaller number. This could be used then to sense or follow the line.

The trouble is that the values that the RCX would return would vary depending on ambient lighting. A line following robot would work great sometimes and not at all at other times. In an attempt to minimize this problem, FLL has standardized the lighting of the competition tables. As per their web site, they place a florescent light fixture a certain distance from the playing surface centered over the mat. This goes a long way to fixing the problem but has not removed it.

Ambient lighting from high pressure sodium vapor (HPSV) lights in a gymnasium will be quite different from the florescent lights at your school. Even with the competition lights directly overhead, the light intensity at the table surface will differ from venue to venue. It is also likely to vary within the same venue. If one competition table is directly under an HPSV and another is not, the light values returned by the same sensor will differ by a few counts.

As a coach, you need to explain this issue to your team. They need to figure out how to solve the problem. Solutions I have seen are:

- 1) Abandon the use of the sensor (Follow the KISS principle).
- 2) Shield the sensor from as much ambient light as possible.
- 3) Select average values and hope for the best.

- 4) Carefully calibrate the software to the conditions at the venue by taking readings at each table.
- 5) Have the robot self calibrate. (An advanced solution for sure!)

## **17.9 The Rotation Sensor**

The rotation sensor is a 16 count optical encoder. This simply means that for every 22.5 degrees (360/16) that the axle is rotated, the brick sends a pulse to the RCX. Furthermore, the type of pulse it sends is different if the shaft is rotated clock or counterclockwise. The RCX counts these pulses. With one type of pulse it increments its internal count. With the other type it decrements the count. The running count therefore gives a measure of the number of times the rotation counter has rotated away from its starting position.

If the rotation sensor is on the same axle as the drive wheel, the count gives an indication of the distance traveled. Every count of the rotation sensor will indicate the robot moved 1/16 of the wheel circumference forward. The rotation sensor is therefore an accurate tape measure of sorts.

It can be made even more accurate by gearing up the rotation sensor relative to the wheel. If the rotation sensor is placed on an axle that rotates faster than the wheel, it will generate many more counts per wheel revolution. This idea can be taken to extremes. At some point, the RCX is simply not fast enough to count all the pulses being generated. Connecting the rotation sensor directly to an unloaded motor is marginal. Add some friction to the motor, and the sensor can keep up.

There are several commands that deal with the rotation sensor in RoboLab. These are shown in Figure 6. The reset command sets the internal count to zero. This is like resetting the trip odometer in your car. The next three commands are "wait for" commands. These cause the execution of the program to halt until the condition is met. The first of these three is a reset and wait. Its first action is to reset the count and then wait for the count to exceed the number given. In this case exceed can be both in the positive or negative sense.

The second wait command does the same as the first but does not reset the count. The 'A' in the icon refers to 'absolute' from the original reset not just the last 'relative' wait.

The last two rotation sensor commands are 'if' or conditional commands. The program will follow either the top or the bottom thread based on the current count. Neither of the two 'if' commands reset the counter.

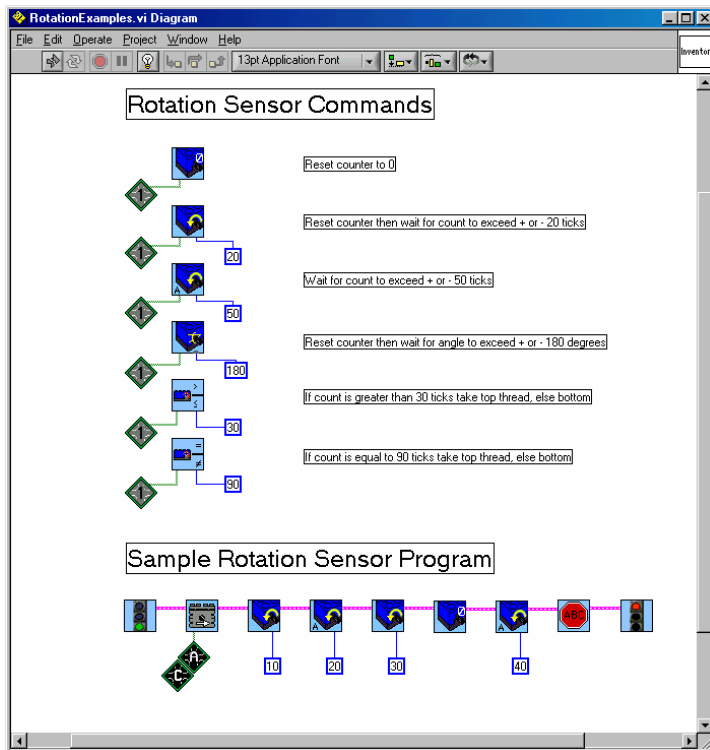


Figure 6

## 17.10A Primer on Mindstorms

LEGO's Mindstorms kit is simply fantastic. They were initially made with the child in mind, but LEGO has found that at least half the kits are owned by adults. These adults use them as a hobby (read toys!) or as a development tool. The fact that you can prototype an idea so quickly has many companies using them during product development. Companies and universities use them for teaching electronics and control skill.

### 17.10.1 Variations on the Mindstorms Kit

Not all Mindstorms kits are the same. Over the years, the commercial kit has changed slightly. The quantities and types of the plastic pieces are different. The RIS software as also changed over time. The kits also started out using an RS-232 serial interface for the IR transmitter that connects to your computer. Today the IR transmitter has a USB interface.

To complicate matters, the FLL version of the kits is different from the commercial version. The FLL version has an extra motor, an extra light sensor, a rotation sensor, and a copy of RoboLab software. It may also have a slightly different collection of plastic parts. In past years, other interesting parts were included with the FLL version. Notably among these parts, was the "gear

block". This part was a gear box to allow the worm gear to mesh perfectly and efficiently with the 24 tooth gear.

It is possible to upgrade a commercial version to a pseudo FLL version. Pitsco-LEGO-Dacta sells the key parts that differentiate the kits. They can be reached at 1-800-362-4308 or [www.pldshop.com](http://www.pldshop.com). The parts in question are:

Description	Part Number	Price
Motor with gear reduction	W775225	\$16.75
Rotation Sensor	W079891	\$16.95
Light sensor	W779745	\$19.95
Gear Block	W779918	\$2.95
RoboLab	W900059	\$59.00

### 17.10.2 Working with Mindstorms

*"You can't fall off the floor" -Paul's Law*

Mindstorms kits have lots of tiny and expensive parts. The team had to respect the kits or lose the privilege of having more than one kit. During the whole 2001/2002 season, we only broke one axle and that was by a simple mistake. I will admit that I had a group of angels. To keep from loosing and breaking pieces, I instituted a work on the floor policy. Small parts can bounce huge distances and fragile motors break when they get dropped from table heights.

Teach the students how to use the brick separator. It will save your fingers.

Read "The Art of LEGO". This great web document talks about the basics of building.

### 17.10.3 Storage of Mindstorms Kits

The boxes that the Mindstorms kit comes in are not adequate for continuous use. Being able to store a kit and keep it sorted is of prime importance. A box with all the parts mixed up is very frustrating. You need to invest in a better storage device. I think that a large fishing tackle box is best. There are lots of little compartments to hold the various parts as well as a large open area in the bottom to store a robot.

My favorite tackle box is a Plano model 8606. The only modification I make is to hot melt glue the dividers in place. It can be purchased at Wal-Mart for \$17.00. A recent trip to the store had them on sale for \$7.00. Perhaps this is an indication that this model will be discontinued?

Another good box is the Plano 1234. It is available at Wal-Mart for \$25.00. A bit more expensive, but the individual covered trays might make it a better choice.

Of course other models of tackle boxes or even other storage systems may work well for you. The prime consideration is allowing a student to see all the parts at once in a sorted fashion. The container should also have a good latch to prevent accidental spills.

#### **17.10.4 Multiple Mindstorms Kits**

As mentioned several places in this document, having more than one Mindstorms kit is a HUGE benefit. This is especially true for larger teams (6 and up). Of course there is a down side to more than one kit. The first is having to purchase more batteries and another tackle box for the kit.

The real downside is having to keep the kits separated. Ten students can completely mix two kits in under an hour and it takes an adult 4 hours to restore them. I quickly instituted a rule to fix this problem. Kits had to be used in opposing corners of the room. Moving from corner to corner with parts in your hand was a "Mooing" offence. A Mooing had the entire team make a mooing noise at the offender. At the elementary age, this was sufficient punishment (and great fun to boot).

I found it helpful to mark the various major parts of each of my kits. With a fine Sharpie, I initialed the motors, sensors, RCX, and tower of each kit. When more than one robot was in use, the markings differentiated them. Strictly speaking, marking the LEGO makes them no longer in "Factory Original" condition as per the rules. I believe that if you keep the marking small you are unlikely to be called on a violation.

Extra Mindstorms kits do not have to be FLL versions with the extra motor and sensors. They can still help in prototyping mechanisms and learning about programming.

One Mindstorms kit for a team of more than 5 will be a difficult challenge for the coach. Too many hands, not enough parts; a recipe for mischief. At the start, when there is a lot of learning to do, the turns on the RCX are short with long pauses between with little to do. Team members get distracted and loose focus. If at all possible, beg, borrow, or steal some more kits. I was able to borrow two kits and kick in my own. We had four kits total for nine members. This was ideal at the beginning. As time went on, we focused on two design approaches and only needed two kits. When we went to competition, we used the second kit to build an exact duplicate of the final robot as a hot spare.

I would discourage using a team member's kit. The issues involved with ownership rights and lost parts are complex. A kit owned by a parent would probably be fine as long as they realize that the kit is likely to get some serious use and lost parts.

## **18 Books**

*"A room without books is like a body without a soul." - Cicero*

To date, I am not aware of any books dedicated to FLL. There are several good books available that deal with LEGO and LEGO Mindstorms. Here is a partial list.

Building Robots with LEGO Mindstorms. Mario Ferrari, Giulio Ferrari, and Ralph Hempel. ISBN 1-928994-67-9

Creative Projects with LEGO Mindstorms. Benjamin Erwin. ISBN 0-201-70895-7

This is a great book to have in your arsenal. The book is a good mix of hardware projects and software. It is the only book I know that has a section on RoboLab. It also has a short chapter on the author's experience with the pilot program (1998) of FIRST. The book comes with a CD-ROM that has project instructions as well as the code samples.

Definitive Guide to LEGO Mindstorms. David Baum. ISBN 1-893115-09-7

Extreme Mindstorms. Dave Baum, Michael Gasperi, Ralph Hempel, and Luis Villa. ISBN 1-893115-84-4

This book has little value for the average FLL team and coach. It has chapters on pbForth and legOS; two non-FLL approved software languages for the RCX. It also has chapters on the design and use of homemade sensors (another FLL no-no). If you truly want understand the RCX and all its capabilities, this book is a great resource.

Jin Sato's LEGO Mindstorms; The Master's Technique. Jin Sato. ISBN 1-886411-56-5

Jin Sato is one of the true LEGO gods. This book details several of his amazing creations including Mibo. Mibo is a two RCX pet that sits, walks, and moves its head. Mibo has no direct use to FLL, but the building techniques are exquisite. Also detailed are walking and wheeled robots. The book also includes chapters in RIS, RoboLab, and NQC software. The last chapters include information on LEGO drawing tools that can be used to detail designs.

Joe Nagata's LEGO Mindstorms Idea Book. Joe Nagata. ISBN 1-886411-40-9

This book is, as its title suggests, a construction idea book. Very little software is provided. The book details nine different projects including a clock, a centipede, and a train. Not one is a 'traditional' FLL robot. As a consequence, this book is not directly applicable to FLL. As a source of some wonderful construction techniques and 'out of the box' ideas it is a valuable addition. 192 pages.

The Ultimate LEGO Book. David Pickering et al. ISBN 0-7894-4691-X

This is more of a history book on LEGO. It has nothing directly on Mindstorms. Never the less, the book is fascinating. It talks of the development of LEGO, the manufacturing process, and is full of fantastic LEGO creations including a Star Wars Storm Trooper helmet.

## 19 Publicity

To further the goals of FLL, you may want to pursue some publicity. A great place to start is your sponsor's newsletter. Your school, PTO, company, or civic organization is bound to have a newsletter. Write an article or better yet have the team write an article for the newsletter. If you have the authorization from your parents, take and include pictures.

Try to approach your local newspaper as well. It often takes a long time to find and talk to the right people so start early. If this is the area's first FLL team, you may get a general information article. What will perk up a newspaper will be a competition. They can get better pictures, lots of excitement, awards, a veritable feast.

If you have a community TV station, you might try them. You might get a spot to announce a competition. You might also get some footage to show.

A sample press release is included in the Coaches' Handbook from FIRST.



## **20 Resources**

### **20.1 FIRST**

If your community has a FIRST team at the high school, you may have access to their resources. Seek out the director of the program.

A FIRST team will have a corporate sponsor. This may be a source of funding. The FLL program was designed to prepare students for the FIRST program. Tell the sponsor the benefits of training their future team members. Make sure you include the FIRST director in any conversations with the corporate sponsor.

Since the season for FIRST and FLL do not overlap significantly, you may find that FIRST team members can act as mentors on your team.

A FIRST team may have a location that you can use to work at if a school cannot provide you space.

A FIRST team may have button making equipment if you go that route.

### **20.2 PTO/PTA**

Many schools have a parent teacher organization or association of some kind. Some will be much more active than others. Try to tap into this resource if possible. A well run PTO/PTA can offer you many services. They may maintain a list of volunteers that you can use. They may help to offset some of your expenses or even pay them all. They may know the best way to approach the staff for unusual requests. They can offer you a group of adults that can witness a dress rehearsal (see section 14.2). They can be a wonderful resource.

Of course, this is a two way street as you now become a resource for them! They may want you to demonstrate a robot, or talk at a meeting about FLL. They may want you to keep track of volunteer hours for various types of awards and matching gifts that are available to them.

### **20.3 Documents**

“Building LEGO Robots For FIRST LEGO League” by Dean Hystad. Available at the Minnesota web site.

Building instructions for the “FIRST Scooter”. Available at the FIRST web site ([www.usfirst.org](http://www.usfirst.org)) under “Team Resources”.

### **20.4 Web Sites**

FIRST main web site: <http://www.usfirst.org>  
This is the main site for the FIRST organization.

FIRST LEGO League (FLL) main web site: <http://www.firstlegoleague.org>  
This is the main site for FLL. All the official rules, registration, prices, and newsgroups are located here.

Minnesota FLL: <http://www.hightechkids.org>

An amazing resource for FLL. They have a coach's guide, an example syllabus and a ton of other information. This site is a must read.

The LEGO company: <http://www.lego.com>

This site has all kinds of LEGO information as well as a link to their online store.

The Official Mindstorms page from LEGO: <http://mindstorms.lego.com>

An official LEGO site that specializes in Mindstorms. They have challenges, competitions, newsgroups, and a picture gallery.

The LEGO Mindstorms Internals page: <http://www.crynwr.com/lego-robotics>

The best site for learning about the guts of the RCX. This is mostly a collection of links to some incredible sites.

The Tufts University Engineering Outreach site: <http://www.ceeo.tufts.edu>

A site dedicated to using Mindstorms and RoboLab for youth engineering outreach. The best site I have found for RoboLab. You can find a RoboLab tutorial and sample programs here.

My own home page: <http://www.fl-freak.com>

A site dedicated to the FLL mission movies, the unofficial FAQ (UFAQ), and this very document.

## Who has this Skill?

- Find anyone (adult or child) that has the skill.
- Get them to write their name on your paper.
- No one may sign your paper more than once!

1. Who has sung in front of a group of people?

---

2. Who can play a woodwind instrument?

---

3. Who can play the piano?

---

4. Who has built something out of wood?

---

5. Who has made something out of clay?

---

6. Who knows how to do a magic trick?

---

7. Who has baked a cake?

---

8. Who has played on a soccer team?

---

9. Who has gotten a strike in bowling?

---

10. Who can count to 10 in a foreign language?

---

11. Who has gone camping?

---

12. Who has a pet cat?

---

13. Who has a pet dog?

---

14. Who has had a haircut in the last week?

---

15. Who has had a broken bone?

---

16. Who has been in the White House?

---

17. Who has more than 3 children?

---

18. Who can stand on their head?

---

19. Who is wearing a pair of black shoes?

---

20. Who has read all the Harry Potter books?

---





### FIRST LEGO League (FLL) STUDENT/PARENT CONTRACT

This contract is to insure that parents and students understand the responsibility and commitment needed by each FLL team member.

### STUDENT CONTRACT

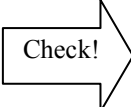
(please check each item you can honestly agree to)

- I realize that no FLL problem has only one solution, and that a successful team is one which cooperates by considering EVERYONE'S solution and ideas. I will not criticize anyone's ideas.
- I agree that my behavior at meetings and tournaments will be constructive
- I agree that each FLL team meeting is valuable and will attend each meeting as best as I can. If a conflict arises, I will notify my coach in advance.
- I agree to cooperate on whatever solution the team chooses, even if it is not my first choice.
- I agree that all solutions will be done completely by me or a member of my team
- I understand that the FLL program recognizes all teams that bring a solution to the tournaments are considered 'winners'. I agree to show other teams the utmost respect and good sportsmanship.
- I agree that the goal of my team should be to do our best to solve a challenging problem.
- I agree to stick with FLL all season, even if I get frustrated sometimes.
- I agree to HAVE FUN!

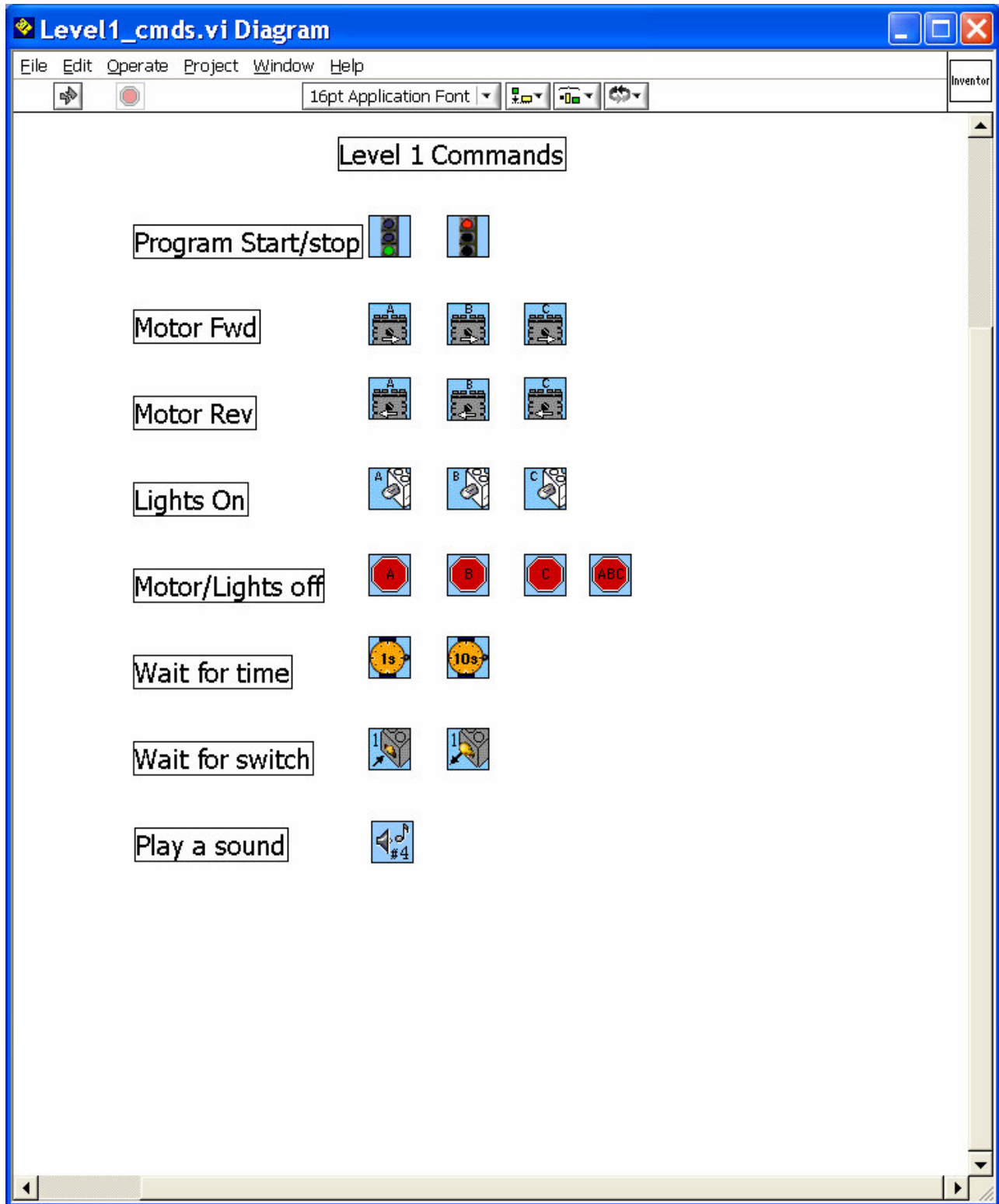
Signature of team member \_\_\_\_\_ date \_\_\_\_\_

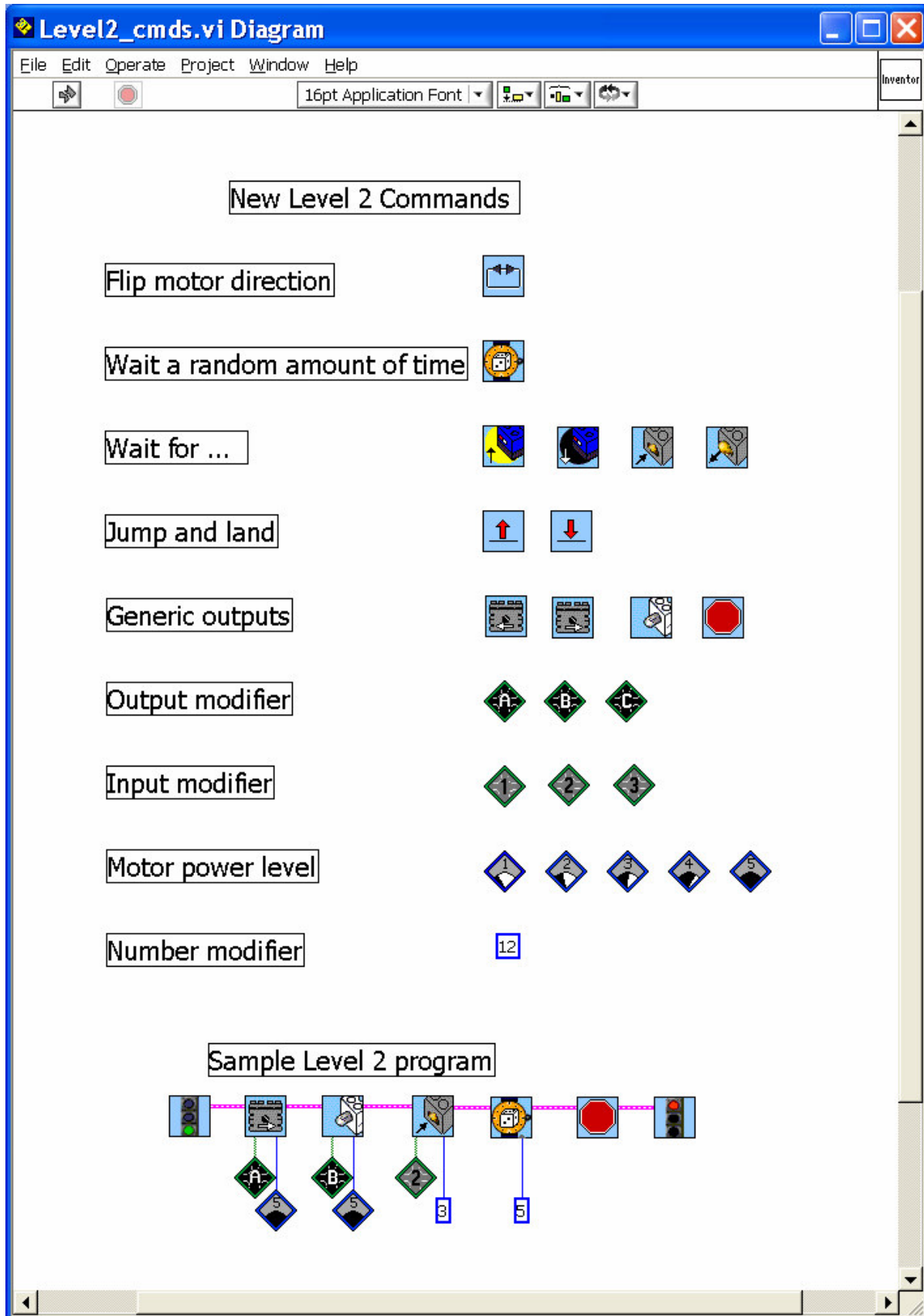
### PARENT CONTRACT

- I agree, in the proper spirit of FLL, not to interfere with the team's solutions. All creations, inventions, ideas, must come from the team members.
- I agree to make every effort to have my child attend every team meeting. If there is a conflict, either my child or I will notify the coach as soon as possible. I understand that my child has a commitment to his/her team.
- I realize that my child's coach will be contributing a significant amount of time and effort to provide a rewarding experience for the team. I will cooperate and be as helpful as possible.
- I understand that team volunteers must be a "Designated Volunteer". This will require a visit to the district office to fill out a form, allow a police background check and be fingerprinted at the school's expense.
- I agree to discuss all items listed above with my child.

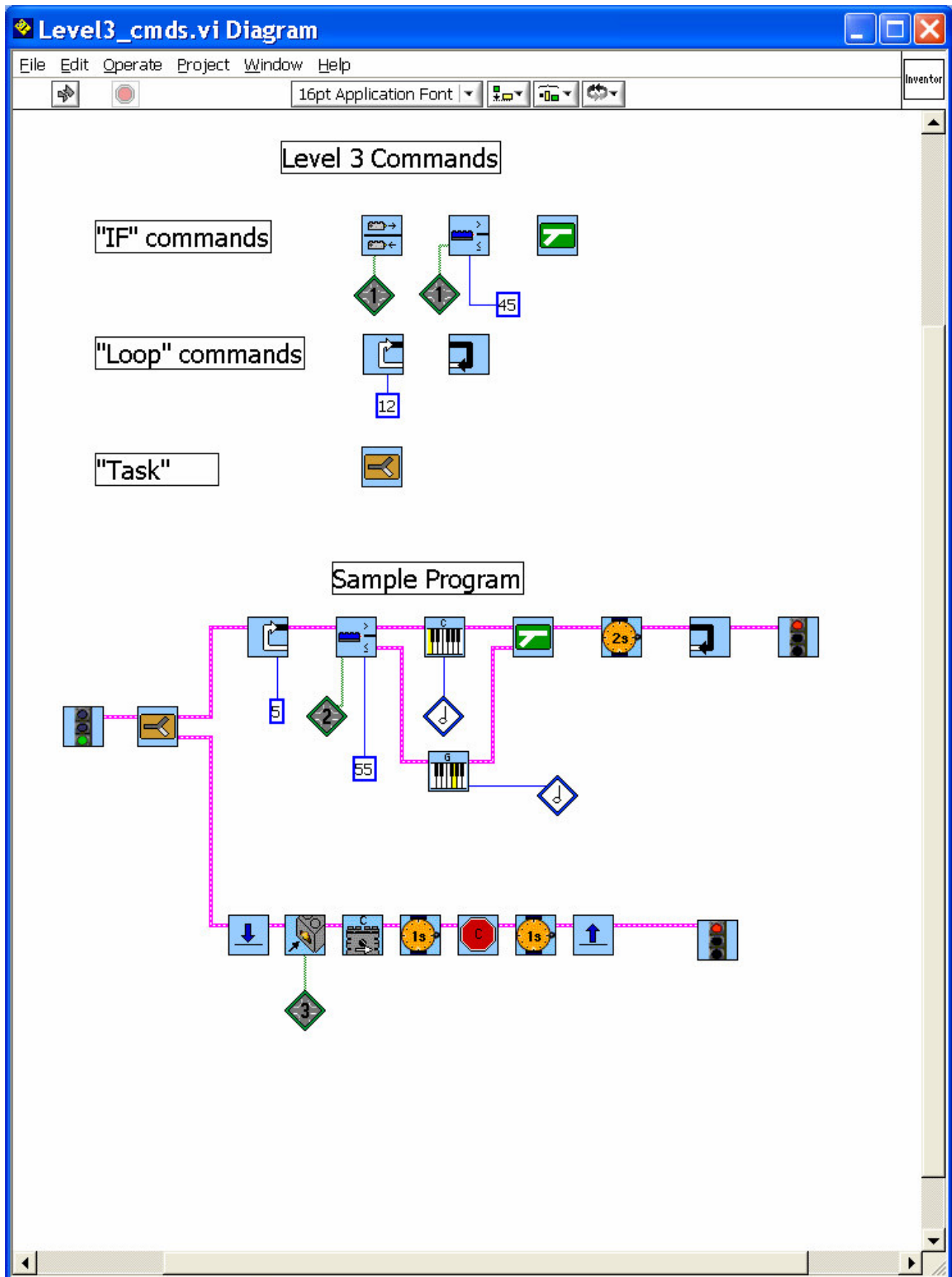


Signature of team parent \_\_\_\_\_ date \_\_\_\_\_









## Mini Challenge 1 (Level 1 commands)

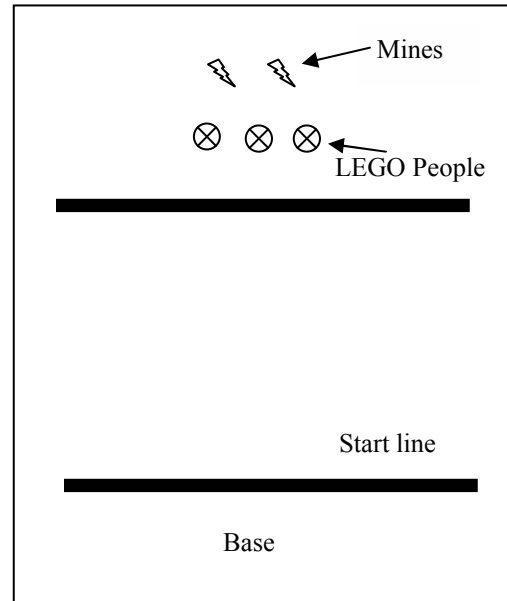
### “Rescue the People-1”

Your mission must:

- Robot must start behind the ‘start’ black line.
- Robot must cross the second black line.
- Wait two seconds for people to climb into robot.
- Do not touch a mine.
- Return to base crossing the black line.

Your program must:

- Be written using RoboLab level 1 commands.



You may not touch the LEGO or computer for the first 15 minutes. Use this time to figure out what you are going to do and who is going to do what! Try a little brainstorming.

Setup: Use black electrical tape for the lines. Lines should be 2 or more feet apart. LEGO people (or just bricks) should be about 6 inches from the black line. Mines (LEGO bricks) should be another 6 inches back.

## Mini Challenge 2 (Level 1 commands)

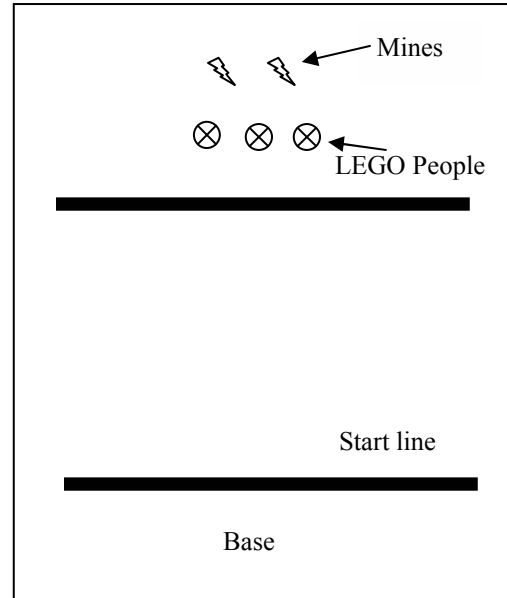
### “Rescue the People-2”

Your mission must:

- **Only start after you press a touch sensor.**
- Robot must start behind the ‘start’ black line.
- Robot must cross the second black line.
- Wait two seconds for people to climb into robot.
- Do not touch a mine.
- Return to base crossing the black line.

Your program must:

- **Be written using RoboLab level 1 commands.**
- **Include a “wait for” command.**



You may not touch the LEGO or computer for the first 15 minutes. Use this time to figure out what you are going to do and who is going to do what! Try a little brainstorming.

Setup: Use black electrical tape for the lines. Lines should be 2 or more feet apart. Lego people (or just bricks) should be about 6 inches from the black line. Mines (LEGO bricks) should be another 6 inches back.

## Mini Challenge 3 (Level 2 commands)

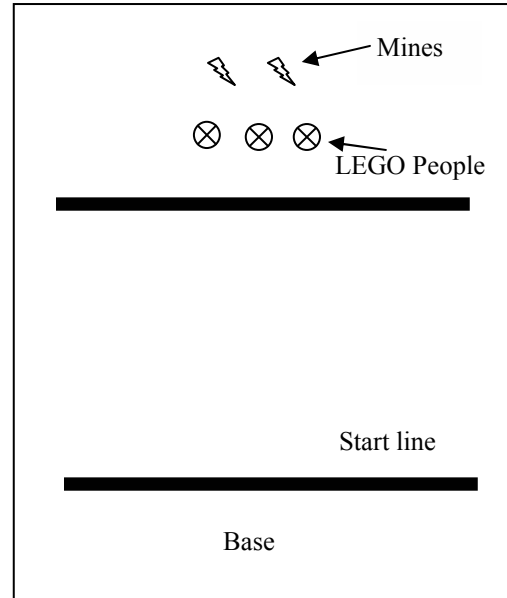
### “Rescue the People-3”

Your mission must:

- Only start after you press a touch sensor.
- Robot must start behind the ‘start’ black line.
- Robot must cross the second black line.
- Wait two seconds for people to climb into robot.
- Do not touch a mine.
- Return to base crossing the black line.
- **Do it all again with a touch sensor press.**

Your program must:

- **Be written using RoboLab level 2 commands.**
- **Include a motor reverse command.**
- **Include a jump/land command.**



You may not touch the LEGO or computer for the first 15 minutes. Use this time to figure out what you are going to do and who is going to do what! Try a little brainstorming.

Setup: Use black electrical tape for the lines. Lines should be 2 or more feet apart. Lego people (or just bricks) should be about 6 inches from the black line. Mines (LEGO bricks) should be another 6 inches back.

## Mini Challenge 4 (Level 3 commands)

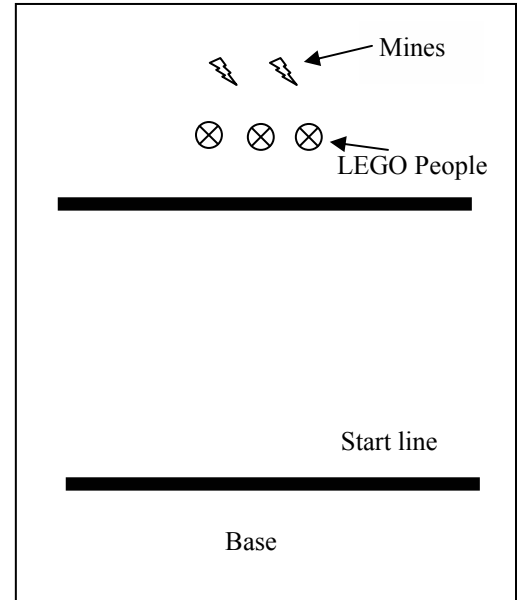
### “Rescue the People-4”

Your mission must:

- **Robot must start when you shine a flashlight on it.**
- Robot must start behind the ‘start’ black line.
- Robot must cross the second black line.
- Wait two seconds for people to climb into robot.
- Do not touch a mine.
- Return to base crossing the black line.

Your program must:

- **Be written using Robolab level 3 commands.**



You may not touch the LEGO or computer for the first 15 minutes. Use this time to figure out what you are going to do and who is going to do what! Try a little brainstorming.

Setup: Use black electrical tape for the lines. Lines should be 2 or more feet apart. Lego people (or just bricks) should be about 6 inches from the black line. Mines (LEGO bricks) should be another 6 inches back.

## Mini Challenge 5 (Level 3 commands)

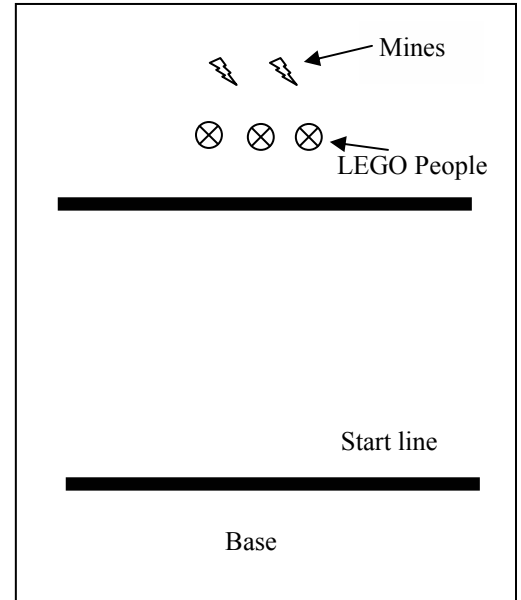
### “Rescue the People-5”

Your mission must:

- Robot must start when you shine a flashlight on it.
- Robot must start behind the ‘start’ black line.
- Robot must cross the second black line.
- Wait two seconds for people to climb into robot.
- Do not touch a mine.
- Return to base crossing the black line.

Your program must:

- **Be written using RoboLab level 3 commands.**
- **Use a rotation sensor.**



You may not touch the LEGO or computer for the first 15 minutes. Use this time to figure out what you are going to do and who is going to do what! Try a little brainstorming.

Setup: Use black electrical tape for the lines. Lines should be 2 or more feet apart. Lego people (or just bricks) should be about 6 inches from the black line. Mines (LEGO bricks) should be another 6 inches back.

## Mini Challenge 6 (Level 3 commands)

### “Follow that line!”

Your mission must:

- Be able to have a robot follow a line.

Your program must:

- Be written using RoboLab level 3 commands.
- Use a light sensor.

#### Details

Your program should be simple and based on the “shades of gray” algorithm. See picture at bottom.

The values you will need for motor off times will be very different for different types of robots. A slow moving robot will be easier to get working.

#### Extra Credit

Modify a COPY of the program to follow the opposite edge with as few changes as possible.

What did you have to change?

Could you make one program to handle both cases? (Perhaps based on a switch?)

Is it possible to follow the line in reverse to get back to base? Why or why not?

Test your program on various lines. Can it follow all the lines? Even very sharp corners? Why or why not?

