

Unofficial FLL FAQ_{05-a} - Unofficial FIRST LEGO League (FLL) Frequently Asked Questions (FAQ)

[Home](#)

Disclaimer: This FAQ is not an official FIRST document. It is an accumulation of knowledge derived from six thousand messages posted to the FLL forum over two seasons. It has been reviewed by numerous people, but may still contain errors. Use at your own risk.

Specifics for previous challenged (Arctic Impact, City Sights, Mission Mars, No Limits) have been excluded. This information is outdated and will only confuse rookie teams that are not familiar with the previous challenges.

The UFAQ has been updated with the new rules for the 2005 season.

Attribution to questions and answers has not been given as it would needlessly clutter this document. Special thanks to the coaches that answered the lion's share of questions posed by rookie coaches. Notably: Jack Gregory, Dave Kolberg, Team BOB, Peter Vogel, Skye Sweeney, Mark Beitz, Marie Hopper, Courtney Goeltzenleuchter, Jackie Keith, Catlin Gabel, and Ken Streeter.

Special thanks to Steve Dakin for helping to automate these pages. He is responsible for the often requested "Printer friendly" page.

Readers are encouraged to submit errors, suggested wording changes, new topics, or comments to Skye Sweeney at skye@fll-freak.com

Table of Contents

[Printer friendly version \(everything on one page\)](#)

[Summary of rule changes since 2004](#)

[How to Ask A Good Question](#) **Please** start by reading this section!

[Allowable parts](#)

[Rules](#)

[Strategy Questions](#)

[Team Questions](#)

[General Questions](#)

[Competition](#)

[Batteries](#)

[Sensors](#)

[The \(research\) Project](#)

[Programming \(Language independent\)](#)

[Programming \(RoboLab\)](#)

[Programming \(RIS\)](#)

[Programming \(Mindscripts\)](#)

[Robot Questions](#)

[Competition Table Questions](#)

[History of changes](#)

2005 Rule changes

[FAQ Table of Contents](#)

Q) What rules have changed since 2004 (No Limits)?

A) In an effort to better the program, FLL makes small changes each year based on lessons learned. Although many of the rules have been reworded or added to, they seem to provide the same intent as last year's rules. Of course it is possible that they have opened up new or different strategies. Hence the rules should be read very carefully. Here are the significant changes that I am aware of:

1. New sections: New sections (e.g. Autonomy) have been added as a way to clarify the experience.
 2. MATERIALS: The fact that wires are unlimited has now been made explicit.
 3. OPERATION DEFINITIONS; ON/ONTO: The wording is new and different than last year's definition provided in the Q&A.
 4. OPERATION DEFINITIONS; IN/INTO: This definition is new and intriguing!
 5. STRATEGIC OBJECTS: This is possibly the most significant change. Satellite objects are allowed but must be released by the robot outside of the base!
 6. PREPARATION: Explicit permission to handle non robot objects off the table has been given.
 7. ROBOT MUST LEAVE BASE: No scoring is permitted until the robot has fully left the base. Catapults seem to be disallowed via the "pushing, carrying, pulling" clause of the last sentence.
 8. HOUSEKEEPING: Explicit permission has been given for storing non active items outside of base.
 9. INTERFERENCE: This rule seems to explicitly prevent certain types of sabotage strategies.
 10. DELIVERABLES/ROBOT TOUCHED: The rule here has been relaxed to be simply in "contact with".
 11. FULL DISCLOSURE/TRANSPARENCY: FIRST appears to have reserved the right to publish questions and answers gathered via private email to tech support. FIRST will no longer consider emails to be proprietary information.
-

How to Ask A Good Question

[FAQ Table of Contents](#)

Introduction

The quality of an answer depends on how well you ask the question. This guide will teach you how to ask a question so that you receive a helpful answer. This is a good skill to know not just for FIRST LEGO League, but life in general.

The people who answer most of the questions posted to the FLL forum tend to be experienced veteran coaches. We thrive on hard problems and good, thought-provoking questions. Give us an interesting question to chew on and we'll go to great lengths to answer it.

It isn't necessary to be competent to ask a question, but it is necessary to demonstrate the kind of attitude that leads to competence. We were all rookies at one time and understand that you have basic questions. Ask them, but please ask them the right way!

Before You Ask

Do your homework

1. Read the FLL Coaches' Handbook. This is the orange and black paperback book that came with your materials from FIRST.
2. Read the Rules. Available here: <http://www.firstlegoleague.org/default.aspx?pid=15870>
3. Read the Mission Descriptions. Available here: <http://www.firstlegoleague.org/default.aspx?pid=15910>
4. Read the Table Setup instructions. Available here: <http://www.firstlegoleague.org/default.aspx?pid=15880>
5. Read the latest version of the Q&A. Available here: <http://www.firstlegoleague.org/default.aspx?pid=15920>
6. Read the Unofficial FAQ. Available here: <http://www.fll-freak.com/faq>
7. Use the search feature on the forum.
8. Try to search the web using your favorite search engine. Google is a good choice.

Many of the non-mission specific questions can be answered by reading the above documents. It will take some time, but do read the material. At the very least skim the material to know what and where it is available.

Next, try searching the forum for an answer. Scan the appropriate folders for an answer or use the search feature to look for keywords. If that fails, try doing a Google search.

Prepare your question.

Think it through. Hasty-sounding questions get hasty answers. The more you do to demonstrate that you have put thought and effort into solving your problem before asking for help, the more likely you are to get good useful answers.

Make it clear that you are able and willing to learn. "Where can I find age appropriate information on gearing?" is more likely to get a good answer than "Please post links for our research paper".

Choose your forum area carefully

The forum is organized in a directory structure. Each area is targeted for certain types of discussions. Asking your question in the proper place is important for several reasons.

The first is simply one of organization. If all the messages are placed haphazardly, it makes it nearly impossible for the next person to find information.

The second is that veteran users will have set up the forum to alert them of new messages only in areas that they are interested in or competent in. Post a RIS question to the Research area, and you may bypass the exact people you want to target. This is especially true of programming questions. Asking RIS questions under the RoboLab or General sections is confusing to all parties.

Please ask your question in just one discussion area. Posting your question to multiple locations is a waste of everyone's time. If you pick the right area, the right people will see the question.

Be patient. The Forum is not an Instant Messenger type system. When you post a question, it becomes a permanent entry in the forum. Other users will log into the Forum at their leisure, see your question, and answer it if they choose to. Often you will get an answer in just a few hours. At other times you may have to wait a day or two.

Use meaningful, specific subject headers

On the forum, the first few words of the first paragraph become the subject line in the outline view. These initial words are your golden opportunity to attract attention. Don't waste it on "Please help me". Make your subject line reflect your question well enough that the next coach searching the forum with a question similar to yours will be able to follow the thread to an answer rather than having to post the question again.

Etiquette

Asking for a reply by private email is rude. It indicates that you will not be bothered to stay current with the forum to get your answer, but are happy to take up others time. If you want to get an email when somebody replies to the thread, request that the Forum send it to you. Take the Forum tutorial to learn how to do this.

Don't TYPE IN ALL CAPS, this is read as shouting and it is also difficult to read.

PLZ keep IM abbreviations 2 a minimum.

Be courteous. Make it clear that you appreciate the time people spend helping you.

Messages with any foul or abusive language will be deleted as soon as found. Privileges to participate can be revoked.

Be precise and informative about your question.

Expressing your question clearly is important. Spend the extra effort to polish your language. It doesn't have to be stiff or formal, but it has to be precise.

Describe your question carefully and clearly.

When you ask your question, display some of what you have learned from your research. We like answering questions for people who have demonstrated that they have done their homework.

Do the best you can to anticipate the questions that will be asked and answer them in advance. Remember to state your programming language if applicable.

You need to be precise and informative, but this is not an invitation to write a tome.

Describe the problem, not your guesses

It's not useful to describe your guesses. Just provide the raw facts. You may be disguising the real problem.

Poor: The robot is not doing what we want it to do. I think the RCX is bad. How can I test to see if it is still good?

Smart: Using RoboLab 2.5.1 our robot does not stop at the proper rotation sensor count. Could the RCX be bad?

Smarter: Robot does not stop at the proper rotation sensor count using RoboLab 2.5.1. Reading the UFAQ and

a search on the web has not provided any information that seems to fit our case. We have included a sample program. It runs on a typical robot using 24 to 40 gearing and the big bicycle wheels. The motors are on ports A and C. The rotation sensor is on port 1 and is geared at the same speed as the motor. Any hints for what we are doing wrong?

Describe the goal, not your approach

If you are trying to find out how to do something, begin by describing the goal. Only then describe the particular step towards it that you have taken. Without stating the goal, we have no way of knowing if you are even on the right path.

Poor: How can we reduce the power level on the motors below level 1?

Answer: You can't.

Smart: Our robot is moving way too fast. We have tried to lower the power level down to 1 and it is still too fast. A few pointers to get us going would be appreciated.

Answer: Sounds like you need to learn about gear ratios. A great free resource is the chapter on gearing in "Building Mindstorms robots for FLL". You can find it here: [Building LEGO Robots for FIRST LEGO League](#)

The smart version of the question is much better. The true goal (a slower robot) was stated given other Forum participants the chance to address your real problem (gears) not the red herring of power levels.

Be explicit about the outcome you are seeking

You are more likely to get a useful response if you are explicit about what you want respondents to do. Do you want pointers, somebody to check a program, provide a rule interpretation, or something else?

Poor: Our robot never seems to do the same thing twice in a row.

Answer: Sorry to hear that.

Smart: Our robot never seems to do the same thing twice in a row. We know it must be possible to get repeatable results but we can't figure it out. The UFAQ has a paragraph on this issue but shies away from any direct answers. With time running out, we need a kickstart. What would you suggest for experiments that might help us figure out this issue?

Answer: What you are experiencing is very normal for a rookie team. To increase repeatability you need to run the following experiments. ...

Don't bother asking for help on solving a specific mission

The whole point of FLL is to solve the missions in your own way. Asking for help is pointless. At best you will get the "That is up to you" answer or some tips on brainstorming.

This is not to say you can't ask rule interpretation or strategy questions. For veterans these are often the most interesting questions.

Poor: How are we supposed to snag the znorknack?

Answer: Anyway you want.

Smart: Is it legal to snag the znorknack before pushing the flopersnork?

Answer: Interesting idea! Nothing in the mission descriptions or the rules precludes it so it should be permitted. Best of luck.

Follow up with a brief note on the solution

Post a reply after the problem has been solved to let them know how it came out. A short and sweet summary like: "It was a dead battery in my serial tower! Thanks, everyone. - Bill" would be welcome. Say what action solved the problem, but you need not replay the whole troubleshooting sequence.

Besides being courteous and informative, this sort of follow-up will help others searching the forum to know exactly which solution helped you and thus may also help them.

Last, and not least, the follow-up helps everybody who assisted feel a satisfying sense of closure about the problem. Trust us that this feeling is very important to those you tapped for help. Problems that trail off into unresolved nothingness are frustrating things.

How to Interpret Answers

If you don't understand the answer provided, do not immediately bounce back a request for clarification. Use the same tools that you used to try and answer your original question (Team manual, Rules, Mission Descriptions, UFAQ, the Web, ...) to understand the answer. Then, if you still need to ask for clarification, exhibit what you have learned.

Example Questions

Poor: "How does a touch sensor work?"

Answer: "Inside the touch sensor is an electric switch that is connected to the touch sensor plunger. When released, the switch is open and when depressed the switch is closed. The RCX can sense the voltage difference between the two states. A current limiting resistor is added in series to the circuit to prevent damage to the RCX in the case the switch get connected to a motor port."

Here the individual was probably interested in how to use the switch in a program or perhaps how to use it in a mechanism. What they got was an accurate answer to the question asked, but not the information they wanted.

Smart: "Where can we find some examples of using a touch sensor in a bumper? We have seen the ones in the Constructopedia."

Poor: Where can I find out stuff about the rotation sensor?

Answer: Read the UFAQ or try a Google search.

Smart: I used Google to try to find "rotation sensor tank drive" on the Web, but I got no useful hits. Does anyone know where I can find information on using the rotation sensor in a tank drive robot?

Poor: Our program does not work. Is there a bug in the software?

Answer: Yes, you probably have a bug in YOUR software.

Smart: Our RIS program does not seem to work when waiting for a specific rotation sensor count. I read the UFAQ but some of the concepts go over my head. Is there a "Dummy's Guide" or other documentation on RIS

and rotation sensors for the novice?

Poor: Must the robot start inside the base?

Answer: Please read the rules, specifically the one on starting position.

Good: How much help am I allowed to provide to my team? They are stuck trying to get a working robot.

How to Answer Questions in a Helpful Way

Be gentle. The stress of the competition season can make people seem rude or stupid even when they're not.

Take negative comments off-line (private email). There is no need of public humiliation for someone who may have made an honest mistake. A rookie may not know how to search the forum or where the proper documents are located. A pointer in the right direction and a link to this document is all that is probably needed.

If you don't know for sure, say so! A wrong but authoritative-sounding answer is worse than none at all. Don't point anyone down a wrong path simply for the fun of it. Be humble and honest; set a good example for your peers.

Ask probing questions to elicit more details. If you're good at this, the asker will learn something — and so might you. Try to turn the bad question into a good one; remember we were all rookies once.

If you're going to answer the question at all, give good value. Don't suggest kludgy workarounds when somebody is using the wrong approach. Suggest the right path. Reframe the question for them.

Help your community learn from the question. When you field a good question, ask yourself “How would the relevant documentation or UFAQ have to change so that nobody has to answer this again?” Then send an update to the document maintainer.

If you did research to answer the question, demonstrate your skills rather than writing as though you pulled the answer out of thin air. Answering one good question is like feeding a hungry person one meal, but teaching them research skills by example is teaching them to grow food for a lifetime.

Acknowledgements

This document is an altered version of “How To Ask Questions The Smart Way” by Eric Steven Raymond and Rick Moen. The original (age inappropriate) version can be found at: <http://www.catb.org/~esr/faqs/smart-questions.html>

Allowable Parts

[FAQ Table of Contents](#)

Q) What parts are allowable?

A) The wording of the MATERIALS rule is deliberate and clear. You can use any of the thousands of unique parts that LEGO has created over the years subject to the exception list. This includes shocks, springs, pneumatics, smooth tiles, magnets, ...

The exceptions list excludes ALL electrical parts except the following:

- 1 RCX

- 1 Rotation Sensor
- 2 Touch Sensors
- 2 Light Sensors
- 3 Motors
- 1 Lamp
- Any number of electric wires

Q) We were wondering if we can use the LEGO solar panel.

A) No. This is an electrical part and is subject to the parts restriction rule.

Q) We were wondering if we can use Star Wars LEGO parts.

A) Yes, as long as they are in factory original condition.

Q) Can LEGO pneumatics be used in the FLL robot?

A) Yes, they can as long as they are controlled by the robot and not a human.

Q) We were wondering if we can use a LEGO Camera.

A) No. It is an electrical part and not on the list of allowable electric parts.

Q) We were wondering if we can use Bionicle LEGO parts.

A) Yes. These parts were not legal up to the middle of the 2004 (No Limits) season. As per FIRST, these parts are now legal for use.

Q) We were wondering if we can use ZNAP parts.

A) Unknown. Some parts in ZNAP kits are standard LEGO elements. These are legal. The other parts are in a gray area. Either contact FIRST for a ruling, or simply avoid the issue.

Q) Does a 'LEGO element' include the box the parts came in? If so, can the box be cut?

A) The official response is no. LEGO does not consider the box (or plastic bags, twisty, or other packing materials) to be LEGO elements. LEGO Technics, LEGO Systems, LEGO Duplo, LEGO Star Wars, etc., are allowed, but none of the products "XXXX by LEGO" are considered LEGO elements either.

Q) Is the LEGO Graph paper considered a LEGO element for the purposes of the allowable parts rule?

A) No, the LEGO graph paper is not permitted as a jig or robot part.

Q) Does 'factory original condition' preclude cutting LEGO parts to length?

A) The rules do not cover this. Look for this to be addressed in a Q&A on the official FLL website. In the past, you were only allowed to cut string and elastic cord to length without penalty. No other materials including the ribbed tubing and rubber bands could be modified.

Q) How will a referee know that some of our older or unusual LEGO

parts are legal?

A) Often the LEGO logo will be molded right into the part. For those that are not, find the part at www.bricklink.com or www.peeron.com and print the page to show to a referee.

Q) Can we use pneumatics?

A1) Reading the Allowable Parts rule indicates that pneumatic parts are legal. They are LEGO parts and not subject to the electric parts list.

Q) How can you use pneumatics legally?

A) You need to automate the switch either by using an electric motor or some kind of environmental device. Note that a number 0 angle connector (part 32013) fits over the pneumatic switch lever. This should give you a good attack point on the problem.

Q) Is it legal to use water instead of air in the pneumatic system?

A) No. You are trying to use water as a building material. Water is not a LEGO product and hence is not legal. (This answer comes straight from FIRST.)

Q) Is pushing a robot bad for the motors?

A) Perhaps. I cringe when I see a robot being pushed around the table like a matchbox car. The motor is now being driven rather than driving. For the average gear train, being driven in reverse can be damaging. If you must push a robot, do so slowly to keep the stress to a minimum.

Q) Where can we buy spare parts?

A) Both LEGO (<http://shop.lego.com>) and Pitsco (<http://www.legoeducation.com/store/>) sell collections of spare parts. For individual replacement parts, visit myriad of stores that make up BrickLink (<http://www.bricklink.com>)

Q) Why do all LEGO motors not turn at the same rate of speed? The variations make navigation very difficult.

A) LEGO certainly did not do it on purpose! It is nearly impossible to make two identical motors. Even the smallest variations in construction will change the behavior of the motor. And even if you purchased two identical motors, by the time they get played with they will no longer be the same.

Q) When you have finished building the challenge kit are you supposed to have any leftover pieces?

A) It is not unusual to have up to a dozen small parts left over. If you have more than that I would recheck all the models that use those parts.

Q) Can we use a LEGO battery box in addition to the RCX?

A) No. The battery box is an electrical part that is not on the approved list.

Q) Can you literally switch the RCX to a whole new different chassis/robot when we come back to base during the round?

A) Yes you can as long as the SUM of all the parts you bring to the table for the round satisfies the Allowable Materials rule.

Rules

[FAQ Table of Contents](#)

Q) What happened to the Green Button Rule?

A) The Green Button Rule (robot must be started by pressing the green run button) has been deleted. It was replaced with the STARTING TECHNIQUE rule. The new rule is much less restrictive in how the robot can be started. This opens up lots of possibilities. Veteran teams should take the time to unlearn the old rule.

Q) I have a team member turning 15 on X date. Is he/she eligible?

A) The cut of date is January 1 of the year in question. Since the competition is in December, you can certainly have some 15 year olds at the tournament.

Q) What should I do if I have a question on a rule interpretation?

A) The best solution is to craft a carefully worded email and send it to Scott Evans at FIRST (flltech@usfirst.org). Please put the word Challenge in your subject line. Rule interpretations tend to be private affairs since you may not want your teams strategy posted on the web. If you have been granted some special interpretation you should bring a printed copy with you to give to the head referee BEFORE the competition.

Q) Please explain the rule for retrieval and deliverable objects.

A) Retrieval and deliverable object rules are perhaps the most complex of all the rules. A few definitions to start.

A retrieval object is one that starts out on the playing field and your job is to return it to base.

A deliverable object is one that starts in the base and your mission is to deliver it someplace out on the field.

Looking for a new paragraph to explain the new rules!

Q) Please explain the container rule.

A) Rule "RETRIEVABLES, POINTS" states that the container is NOT to be considered when scoring objects. Each object must stand on its own merits. Containers are still valuable in reducing chaos.

Q) Are projectiles legal?

A) Yes. As long as the device is under the robots control. Using a projectile to solve your own mission problems is quite appropriate. Using a projectile to harass the other team is counter productive. a) The other team can ask for the projectile to be removed from the table. b) It violates the "Gracious Professionalism" prime directive.

Q) Once fired, can we retrieve our projectile without penalty to use it again?

A) Not unless the projectile is tethered to the robot. If it is loose, the "LOSS OF CONTACT" rule takes effect and you loose it.

Q) Can you program your robot to return to the base, but while there NOT touch it?

A) Some teams like to save on a programs slot by returning to base, wait, manually placing a delivery object in front of it, then let it continue. This is legal. In the past, due to the "green button rule" this was an important issue. With the new "STARTING TECHNIQUE" rule, this is a minor point. This approach is legal but rather risky. If the first half of the mission fails, the whole mission must be rerun.

Q) Our team is wondering if we could build a robot with two motors attached via long wires to the RCX. The RCX would remain in base, and we would manually control the robot using touch sensors attached to the RCX. We would not be touching anything outside of the base. Is this legal?

A) No. This violates the "FORCED RESTARTS" rule that states all mission ends when any part of the robot is touched by a human. In your scenario, part of the robot is still in the base and would be touched with the first command thus terminating the mission and any chance at a score. Note that a split robot as you suggest is legal if you do not use human intervention during a mission attempt.

Q) Can we create a little satellite robot that attaches to our primary robot using some of the long wires?

A) A satellite robot is legal as long as it is under RCX control. It would not be legal to make a satellite that was controlled by a human back at the base. After the robot is started, no human intervention is allowed without terminating the mission attempt. Of course the satellite is subject to all the same rules as the primary robot. Notice that it would be legal to retrieve the satellite by having a human pull it in on its tether! This would terminate the mission but you would not be subject to a robot return penalty. The reason that this is legal is that part of the robot is still in base. Just remember that retrieval objects must return to base under their own power.

Q) Can we have one end of a long wire attached to a touch sensor in the base, and the other end of the wire on the RCX outside the base to control it?

A) No. This is similar to the above question. The mission will end as soon as you touch any part of the robot including your wired remote control.

Q) Our satellite robot uses a LEGO windup motor to power the vehicle. We understand that the robot has to initiate the satellite robot, but is it against the rules to have the motor pre-wound before we get to the tournament table?

A) A pre-wound motor is legal but you MUST have the robot release the satellite not a human player.

Q) Can we use more than three motors if they are not attached at the same time?

A) No. The robot cannot have more than 3 motors total. The robot is defined as the sum of all parts that are brought to the competition table.

Q) Explain how the robot starts in the base.

A) At the start of each mission attempt, all parts of the robot must start fully in the base. Consider the base to be an imaginary box with a ceiling of 40cm. The robot must start each mission attempt fully inside this box and not just at the start of the *first* mission attempt.

As a concrete example of this, consider a real-life analogy:

* The robot is a big-city hook-and-ladder fire truck.

* The base is the fire station garage.

* At the start of a mission, the fire truck needs to fit inside the fire station garage.

* However, the big-city hook-and-ladder truck probably has a ladder than can reach at least 10 stories high.

Q) When can we touch our robot without penalty?

A) You can touch your robot when it is in the base without any penalty. When returning from a mission, you can touch the robot as soon as the robot breaks the 'imaginary box' over the base without taking a robot return penalty. If you are returning objects, each retrieval object must also cross the 'imaginary box' before you touch the robot to gain points.

Q) Can we use tools or alignment aids inside the base?

A) Yes, but these must be made out of LEGO as well. No paper or other materials. You can also forget the cardboard box the Mindstorms kit came in! According to the rules, the alignment aid must stay completely inside the base.

Q) I have heard that we should cover the RCX port with tape. Why should we and is it legal?

A1) Strictly speaking it is not legal. Some tournaments provide a house rule that allows this. The issue here is to prevent another team from programming your robot by mistake. With careful robot handling procedures, tape is not needed. If you do use tape, make sure it is opaque to IR light. Some tapes including certain black electrician's tape are transparent to IR light!

A2) Forget tape, use LEGO! Make a cover over the port using LEGO elements. This can snap on and off easily and leaves no sticky mess. It has been noted that yellow LEGO bricks are not completely opaque to IR light!

Q) Does the 40cm high base rule apply for returning robots?

A) There are no size restrictions on a returning robot written or implied anywhere. The robot could leave base, grow to 4 feet high and return to base without penalty.

Q) What are strategic objects and how can they be used?

A) Strategic objects are objects made out of LEGO bricks that are used by your robot to accomplish one or more of the tasks. To be strategic objects they cannot be physically connected to your robot. They could be platforms to set things on, jigs, ramps, projectiles, etc.

Q) Is the coach allowed to touch the robot during competition?

A) During the competition (as well as the whole season!) only the kids should be running the robot. The adults should keep their hands off! The only exception I have ever seen was a team that got hit by the flu. Only one student was able to make it to the tournament. During the 'teams' match, the coach stood nearby and handed him parts as he called for them.

Q) Can a robot touch the top of the wood border?

A) When the robot first starts from the base, it must be fully inside the base. The base is defined as the INSIDE wall of the border. This means that initially you can't touch the top of the border. But once the robot has been legally started, it can then extend out over the border and touch its top. Just remember that not all competition tables are built the same! Some may have obstructions in non standard locations to support the lights.

Q) Are attachments to the robot (like an arm) considered part of the robot?

A) The rules indicate that the robot is the RCX and anything connected to it. For example: if a robot has a long arm that crosses into the base ahead of the RCX, then the robot is considered to be in base and can be manually manipulated without penalty. Just remember that retrieval objects must cross into the base before you touch the

robot for them to score. Conversely, that long arm must be fully inside the base to be allowed to start the robot on its next outing.

Strategy

[FAQ Table of Contents](#)

Q) Some of the field models are somewhat fragile. Can we use this to our advantage?

A1) No, the rules explicitly disallow using fragility as part of your strategy.

A2) No, you will upset the referees and judges (they can make comments on your behavior!) if they have to rebuild models after your round.

A3) No, some tournaments glue the field models together to avoid this problem.

Q) How can we adjust for different lighting conditions at a competition?

A1) There are any number of good ways to solve this problem. To list them here destroys your team's chance to brainstorm this issue. Teams should come up with their own solution that matches their overall situation.

A2) A solution is to take light sensor reading of the competition tables prior to the tournament. Most tournaments will allow a visit to the tables for this (and only this) reason.

Q) Are humans allowed to manipulate field pieces when they are in the base?

A) Yes. You may manipulate field pieces that are in the base. These pieces may have originated in the base, or may have been moved to the base by the robot. Once you 'own' them, you can do what you want with them within reason. Stacking, sorting, connecting them, or adding a sub-assembly would be reasonable.

Disassembling them, or moving them (throwing/pushing/shoving) by hand onto the field would be unreasonable.

Q) Are strategies based on sabotage allowed?

A1) The rules have not precluded robot initiated sabotage. However, as a coach you should think real hard about letting a team pursue this avenue. What ethics are you teaching? Does this follow with the FLL "Gracious Professionalism" prime directive? How are you likely to be viewed by other teams, referees, and judges?

A2) The 2005 rules now include an interference rule. This would seem to discount many if not all sabotage strategies.

Team

[FAQ Table of Contents](#)

Q) What is a team banner/poster?

A) Many teams will create some kind of banner/poster to 'mark their turf' in the pit area. The item usually has the team name and number. It might have pictures of the robot and or the kids. There is no right or wrong here, but having one is fairly important to show your team spirit. A 'tri-fold' science fair display is a popular choice

because it is free standing.

Q) Is there a standard size and information content for a team banner?

A1) There is no standard for a team banner. Not all teams will even have one. Some tournaments will provide a fabric backdrop to pin your banner to, others will not. What is pretty sure to work is a tri-fold display board. This can stand on its own on the table you are likely to be provided. You put whatever info you want on it. I have seen pictures of the team in practice, information on research projects, printouts of tricky code, or just fancy art work logos. You should contact your tournament director for specifics.

A2) We've used an extra team t-shirt and PCV pipe to "hang" the shirt from to use as a banner for carrying onto the competition floor before. Just make a T with PVC. We've also used a roll of banner paper, decorated it, and then taken it to a teacher supply store to be laminated to create a banner. It's was very cheap to laminate it there.

Q) Does anyone know any good team names?

A1) Sorry, all good FLL team names are currently in use!

A2) Try a brainstorming activity! Make a list of names and do not stop till you get at least 30. Then go back and talk about them. Delete your 10 least favorites. Add another 10 names to replace them. Sleep on it. Come back and take the top 10. Add 10 more. Sleep on it. Take your top 10. Now say the names like a radio announcer would. "And now at table 2 all the way from Franklin middle school above the Arctic Circle are the 'Insert your name here'". Now yell the names like you are cheering on your team. Now you should be ready to pick your favorite two or three. Sleep on it, than pick the favorite. Try to make sure the vote is not a simple majority. A split decision makes half the team fell left out.

Q) How do you assign roles to team members if all the kids on a team want to be builders/programmers/...?

A1) Each team is different. Some teams assign roles from day one. Other teams assign roles as it becomes obvious who is good at what. Others reject the ideas of roles and have all kids cycle through all roles.

A2) Within reason, let the kids self select.

A3) Let the team brainstorm what roles they see as being needed (Builders, Programmers, Project Leaders, Documentation...). Let them self-selected so that they have a pair for each category with the understanding that everyone gets an opportunity to do a little of everything. The head honcho in each area is there to guide, assign, teach, etc...

A4) What's worked best for us is to divide the team into pairs, with one builder, one programmer in each pair. Each pair works on different tasks. To share the robot, when one pair works with the robot, everyone else works on research, artwork, etc. Another advantage is that at tournament time, the pairs will be used to working together and can tag team for the different missions they've worked on.

A5) Every member on my team participates with every "job". I do not want the year to end and have a child that did not experience every aspect of the event.

Q) How do you keep all my team members busy?

A) FIRST dictates a maximum team size of 10. If you have 10 students on your team and only one Mindstorms kit, you are in for a rough ride. Having six team members with one kit seems to be about right. If you are in the 'danger' zone, here are some ideas:

1) Buy or borrow another Mindstorms kit. This will allow you to try parallel development and keep the kids on task.

2) Have the others work on the research project, team spirit, and other tasks.

3) Have a dedicated testing team besides your hardware and software groups.

4) Have them work on the team banner/poster/thing.

5) Have them work on a hardware and software specification summary sheet to hand to the judges.

6) Ask that some students skip meetings to cut down the numbers (reminding them they are still needed).

Q) What if the entire team cannot make the tournament?

A) FIRST puts a limit on the number of students on a team during the season. The lower limit is ignored at the time of the competition. I have seen a single student at a competition when the rest had the flu (they won the Against All Odds award). Of course, demonstration of teamwork and team spirit becomes problematic.

Q) Why do some teams have more than ten team members?

A) They should not. Sometimes an organization has registered several teams. These teams may all have the same shirt design and travel as a group cheering each other on. Add parents and coaches and it may look like the team is a thousand strong.

Q) How can I increase the commitment of my team? Kids are forever missing meetings in favor of other activities.

A1) Have the students and parents sign a contract with you at the start of the year. A sample contract can be found at www.hightechkids.org/fll.

A2) Make sure the students and the parents know in advance what your expectations will be.

Q) When should I expect my team to be done programming?

A) Never! Changes are likely to occur right up to the competition. Some coaches impose a 'code freeze' a few days before a tournament. Only critical changes are allowed and only if the whole team agrees.

Q) Does my team need tee-shirts?

A) Many teams do make custom tee-shirts. It is a great way to show team spirit. They are certainly not required. Lots of options exist to keep the cost down. You can use ink-jet iron-on transfer paper. You can do each by hand with 'puff' paint. Some tee-shirt shops may have a photographic method that might be cost effective.

Q) How can it be fair to have 9 year old compete against 15 year olds?

A1) In my experience, the question should be asked the other way around. The younger kids are eager and will listen to a coach's instructions. The older kids start to have social issues that prevent them from working together well. Older kids also tend to design overly complex robots that do not perform as well.

A2) Who said life is always fair?

A3) While trophies are a great bonus at the end of a tournament, the mere act of working to develop a robot, its programming, and a presentation leads to some great opportunities for personal development of your team members. Take time with your team members to emphasize on doing their personal best. A FLL tournament, much like a golf game or a music competition, is about striving to do your personal best, overcoming obstacles, working together as a team to solve a common goal, and above all, having fun.

A4) Just remember your 9 year old gets to compete next year as a veteran.

Q) How can it be fair for a homeschool team to compete against a public school team?

A) This comes up each year when homeschooled teams start winning tournaments. People falsely believe that homeschool teams spend every moment of each day working the problem as some kind of long term project. For the most part, this is absurd. The reason they tend to do well is that the team starts as a cohesive group from day one. They share some kind of bond that pulls them together. This might be a religion or simply surviving in an area without a public school. They also have parents that are very supportive. So is it fair? It is not unfair. If you can bind a team together and get the parents to be supportive you are on equal footing.

Q) At a competition it became clear to me that one team's robot and software was not the product of the kids but that of the coach. What should I do?

A) The ideal answer would be that it does not matter. When the team is interviewed by the tech judges they will spot the ringer. I think this is a reasonable expectation from a tournament that has fairly competent judges. A few pointed questions that are met with blank stares speaks for itself. They are out of the running for a tech trophy and probably for the Director's Award as well. Of course we do not live in an ideal world. A private word with the coach to remind them of the FLL values and the FLL Honor Code they signed would be my next step. If the abuse continues at the competition, inform the tournament director. If the abuse continues on a subsequent year, I would alert the tournament director in advance of the tournament.

Q) How can I get girls interested in joining an FLL team?

A) When recruiting, downplay the LEGO aspect. Talk about technology, teamwork, team spirit, and research projects. For whatever reason, PARENTS think boys when they see LEGO. You can also mention that the team will have X girls on it. When a parent sees this in writing, they start to think about their daughters!

Q) I have heard rumors of community service requirements. What are they?

A) There are no requirements. The FIRST high school program does get points for community service. This is probably where the confusion comes in. Of course, a little service would not be amiss. These types of activities go a long way for the FLL Values category for the Director's Award.

Q) How can we raise the money to register an FLL team?

A) You can:

- 1) Try approaching technology companies. They may be able to help you with money or mentors.
- 2) Old fashioned bake sales etc.
- 3) Share the cost across the students.
- 4) Parent teacher organizations may help.
- 5) Small companies may make cash donations or donations of services or goods.

Q) What is the difference between a challenge kit, an FLL kit, and a Mindstorms kit?

A) The challenge kit changes each year. It contains all the LEGO pieces needed to make the various doodads on the playing field with which your robot interacts. An FLL kit contains 'all' the LEGO pieces that you need to build your robot. An FLL kit and Mindstorms kit differ in the number and type of parts provided. Notably the Mindstorms kit only has 2 motors, 1 light sensor, and no rotation sensor. An FLL kit has three motors, 2 light sensors, and one rotation sensor.

Q) I am a rookie coach, of a team that only got started in November. We dont really know what we are doing. How will we be viewed at a competition?'

A) I would be shocked if you were not well received even celebrated. FLL is all about the kids doing the best that they can. Being a Tournament Director is not a paying job. They do it because they want to give kids a positive experience. The judging will be conducted in private. The judges are instructed to question at the level of the team. In just a few moments the judge team can figure out the level of the team and ask questions at their level. Of course, the final score will vary based on the degree of difficulty. Treat this year as a building year (you are coming back next year right?). Learn the tricks. Watch the other teams compete at the tables. Ask the

judges/director for a copy of your judging sheets to know where you need work. Stress the importance of all aspects of the challenge not just the robot performance. Remember the robot score is only worth about 25% of the total score. Team spirit is worth the same amount. How about a team stunt/song? There is a wealth of information on the web. Talk to your state Tournament Director about possible classes and training. If none are available, consider trying to coordinate some yourself. You may not be able to teach a class, but you might be able to find space and a willing veteran coach.

Q) My team will not be ready for our tournament. Should we still go?

A1) If you have qualified for the tournament then you can go regardless of what you can or can't do. I would highly encourage your team to go and participate in the tournament, regardless of their state of readiness. The coach should prepare the students to win no awards, and have scores which are far short of other teams. However, there is no substitute for going to a tournament. Even if the robot has only one mission it can do, just doing that is a significant accomplishment!!! The attitude of all FIRST tournaments I have attended is that all participants should be treated as WINNERS! Having an opportunity to demonstrate what the team has learned (even if that is not a lot in an absolute sense) is a valuable experience. Good judges will be encouraging to the students, regardless of their level of readiness.

A2) Before canceling, I would highly suggest that the team coach talk to the Tournament Director and get their perspective. The goal of going to the tournament is not just winning, but is to have an opportunity for the students to demonstrate what they have learned, have others appreciate their work, and be a part of the FIRST Lego League experience. I have seen teams with a robot which could only do two missions, but with students who were completely elated that both of their missions worked! As long as the expectations and attitudes are set appropriately, even the last place team should have a lot of fun at an FLL tournament! That said, if you decide not to compete, going as spectators is an absolute must!

Q) Our team won a LEGO trophy at a competition. Is it possible to order more so that each student can have one?

A) No. The LEGO trophies are only provided to official regional tournament partners. A solution is to make your own mementos to hand out. Perhaps distribute the field models on plaques, or build your own custom trophies from scratch. For No Limits, I was able to build five "mini stairs" from the parts of the stair model. These I mounted on plaques with an engraved plaque.

Q) What does a Documenter do?

A) A Documenter's job (if you even have them) is to document! They take pictures, take notes, take video, and otherwise record what the team has done.

General

[FAQ Table of Contents](#)

Q) The PDF documents we print from the MN FLL site (www.hightechkids.org/fll) have strange characters in them.

A) Get the latest version of Acrobat reader. If the problem still persists, see if there is a later version of your printer driver.

Q) What is a programming garage?

A) It is an IR opaque box in which you program your robot. It can be as simple as a cardboard box. This prevents your programming tower from programming other robots in the vicinity. If you keep your robot inside

this box when not needed, it helps to prevent your robot from being accidentally programmed!

Humor

In relation to the catapult supplied with Mission Mars

When our MAV launcher was newly built, it was capable of sending the sample canister on a trajectory that left the vicinity of Gusev crater. Now that the re-useable space vehicle has been flown one too many times by the space agency, the launcher is incapable of injecting the canister into even a low orbit and the canister always comes crashing back down onto Gusev crater. Moreover, the canister usually lands directly on top of the Great Wall of Mars and ricochets into the ice drilling operation, creating much havoc. Has anyone else experienced this performance degradation and what have you done about it? I believe that it may be a thrust issue with the launcher's energy storage devices but I was wondering what sort of launch performance everyone else was experiencing before my team performs a complex repair EVA?

In relation to the loss of the Spirit rover

January 22, 2004 NASA scientist revealed that they have lost communications with the Spirit mars rover. It has been over 24 hours since any non garbage data has been received. They are investigating what type of hardware or software failure could explain the problem. It is rumored that thousands of FLL team members around the world are tying up the phone lines at NASA. They are all trying to tell NASA that a Martian just took a flash photo of the rover. Dr. Justin Case is reputed to be on route to NASA to help them download new firmware. Just before he was whisked away by helicopter he was overheard saying "Make sure you have the latest version of RoboLab waiting for me when I get there".

In relation to the number of balls needed for the No Limits competition

Q) How many balls does each team have at the beginning of the tournament?

A) "Eight shall be the maximum number of balls, and the maximum number of balls shall be eight. Nine balls shalt thou not permit. Seven is right out. Once the number eight, being the number of balls counted, be reached, then no further balls shall be permitteth..."

Poetry

FLL Poem for 2001 Challenge:

Twas the night before FIRST Lego League, and all through the house,
Not a computer was stirring, not even a mouse.
The children were nestled all snug in their beds,
While visions of robots danced in their heads.
The coaches were sleepless and pacing the halls
Wondering if they had taught their kids anything at all.
Oh med supply, oh weather tower, can you be deployed?
If our robot malfunctions we'll be so annoyed.
Oh instrument, oh scientists in need of a rescue,
Back at the ship we will have a big barbecue.
Oh ice core, oh red flags, oh ten brown fuel barrels,
When our programming works, we break out in carols.
Oh weather balloon, oh ice floe, quite a challenge to us,
With very low clearance, we never got across.
To all of the teams, we wish you good luck
And hope that your robot does not run amuck!

FLL Poem for 2002 Challenge:

Tw'as the night before FIRST Lego League, and all through the house,
Not a computer was stirring, not even a mouse.
The children were nestled all snug in their beds,
While visions of robots danced in their heads.
The coaches were sleepless and pacing the halls
Wondering if they had taught their kids anything at all.
Oh food loops, oh building supplies, can you be delivered?
If our robot malfunctions we'll be in a tither.
Oh modular housing, what shall we do?
No matter how hard we try, we can only stack two.
Oh bridge repair, oh windmill, how can it be?
We must be successful with a percent higher than 50.
Oh gathering of toxic barrels, don't be our disgrace,
If only our robot can get back to base.
If we can clear the soccer field of those pesky old rocks.
We'll take off our shoes and dance in our socks.
In the midst of competition we don't want to hear "whoops"
When our robot can't gather all the food loops.
And what about research, you want a solution?
How can a robot reduce pollution?
Good luck to y'all - when it's all said and done,
it doesn't matter who won,
but that our cities will flourish for many years to come.
Humming quietly, In the garage,
A team member clicks rapidly Through a barrage
Of Lego programs that fly past the screen
To pilot a robot (that sometimes acts mean)
All over a 4 * 6 table Accomplishing tasks
That were a challenge at first; But now a team member asks:
"How can we make This bot more robust?
'Cause surely we don't want the tourney... A bust??"
So then he grabs Bricks, beams, axles and more,
In the hopes of creating Attachments galore.
Programs and attachments that work-All the time!
Maybe a robot That turns on a dime!
So grab the Ritz crackers,
Get Mountain Dew-We'll be here a while
Creating a new Robot that works so well
Every team member stands, shaking,
From pride and from joy, Even the coach is quaking!
The coach praises the team And says "Good job everyone!
But one more thing- It still is not done –
You need research to present!" The coach says frantically,
So the team gets to work (Of course, very coolly)
The team makes a video To raise awareness of the need
Of accessible places and things So take heed!
So with the research all done And a great working robot,
The team heads to the tourney With only one thought:"
We did the work, Not the coach, not a friend;
And we're SURE that We didn't bend
Any important FLL rules, But we're not here to win,
Because we already have –
We learned, worked hard and had fun,

And that's really what counts
 (though it is nice to take home a trophy)
 Remember that, if your robot screws up,
 And faces are red, angry and poofy.
 Remember, it's not any special award;
 It's what's inside... What you learned, what you did.
 So remember, when things seem daunting next year,
 Don't. Fear. Just remember the FLL values-
 Teamwork, cooperation and learning;
 And maybe, just maybe,
 If you show the judges, you might be earning
 The coveted Director's Award.
 But if not, it's okay... You can always try again next year, anyway!

FLL Poem for 2004 Challenge:

*Twas the night before tourney and all through the house
 Not a computer was stirring not even a mouse
 The robots were tucked all snug in their case
 In hopes of great missions launched from home base
 The kids were in jammies the coach, she was too
 Wondering if their hard work was enough to get through
 When out in the garage there arose such a clatter
 Coach jumped from her bed to see what was the matter
 Away to the door she flew like a flash
 Turned on the light tripped on a skate. CRASH!
 The fluorescent lights shimmering down on the mat
 Gave a sight to behold for coach and the cat
 For what to her wondering eyes should appear
 But a bevy of elves with a cold case of beer
 More rapid than DSL those little guys came
 They whistled and shouted and called out their names
 Now! Codeman! Now Sensor! Now, Tester and MayGo!
 On, Robust ! On, Wheeler! On, Teamwork and Lego!
 To the top of the stairs and open the gate
 Now test away, test away fix that wheel mate.
 With gracious professionalism and a wink of their eye
 They met each obstacle with a cheer and a cry
 So up to the table those little guys flew
 The robot fed dinner for elves and pets too
 And then they jumped down no one got hurt
 Robo pushed in the chairs and swept up the dirt
 One was dressed in red another is blue
 But most were in black with a few in grey, too
 A bag full of beams they flung on the table
 They engineered, fixed and mended where able
 Their eyes how they twinkled as they sucked back the drinks
 All over 21 clearly, me thinks
 They chattered and laughed until the wee hours
 Working marvels and wonders on that robot of ours
 Suddenly and sadly as quickly as that
 They undid their work and tidied the mat
 Then they hit the white flag and jumped in car

Obviously arriving from somewhere afar
Driving out in the night a chorus of elves sang
"Sorry sweet children you must do it yourselves"
Written by Rookie Coach Tracey Ananmalay
With deference and thanks to Clement Clarke Moore

Inspirational

I am not sure why I am writing this message. Perhaps it will have a cathartic effect on me. Perhaps it will make other coaches understand that what they are going through is normal. It is 10:00PM on Friday. In little over 10 hours from now, my team will be competing. We are not prepared. We normally do not meet on Fridays, but we held an emergency meeting after school till 7:00 PM. We made some good process, but we only got the ball halfway up the hill. Our best score is perhaps 150 points if all the planets (including Mars) are aligned and we have a following wind. Our research project is a fantastic idea, but the execution is just passable. Our team spirit is high, but the tee-shirts we ordered (in plenty of time) will not be ready. But (here comes the cathartic part) it does not matter. The kids will compete, they will get to see the other bots, they will do their thing in front of the judges, and yell their heads off. The coach will do the same. And at the end of the day, if they do not get to the podium, it will be disappointing but not devastating. In fact I hope it stiffens their resolve to work harder and better next year. For those competing this weekend, have the best of FUN. That's what it is all about.

I just loaded the car. Hope I didn't forget anything. If we're lucky we will run some of the missions and they will go as planned. Our robot is much too sensitive to positioning in the base. However, it has been a great rookie year no matter what we score. My kids have learned to think out of the box and build without directions. They can think in programming terms and break a problem down step by step. They are starting to understand how to work together and not worry about whose idea it was. So, we're going to be at our tournament at 8:30 tomorrow morning, unprepared to handle whatever comes our way. But handle it we will. My biggest thrill this year was when one of the kids looked at me and said "But it looked so easy on television. Reality is so much harder!" I told him that he had nothing left to learn, my job was done!

It is 9:00PM and the last of the team has just left my home. They had arrived here after school. I worked them to near death, feed them dinner than worked than to death again. Going into tonight's meeting we had all the parts but we needed polish. Tonight they sparkled. This is a 4th year team that is mostly original members. They have worked together well from day one. This year I was a bit worried in that our qualifying competitions are VERY early. So I changed my strategy with them. From the second meeting we held, I forced them to run at the table (a solid 16 points – the bonus objects!) and give me their research presentation (pulled out of thin air). We have been doing that each meeting. I think it has made them realize what was still left to be done, and gave them a great boost when their score started to climb up. Interestingly, the presentation they created on the third meeting is the one they went back to a month later after they tried a bunch of other ideas. The off-the-cuff presentations I forced on them was a good idea because they learned they can discard work. Before, if they had invested even a minute of effort, they would not toss it. So tonight they alternated going to the table, doing the research presentation, and making small corrections to the presentation. We agreed not to touch the robot software but just work with the devil we know rather than the devil we do not. As it is they have a very reliable and solid score. If nothing else, we still know how to get a solid 16 points! Tomorrow we compete. Win, draw, or loose we will have a great time, and that is what matters most. Good luck to you all.

Just a word to all you teams competing this weekend. I am sure you are all as tired as I am. It has been a real push these last few weeks to get everything done in time. I wish you the best of skill. I will not wish anybody luck. You all have plenty of skill. Just assembling the mission models took skill. Hands up; how many of you got it right the first time? So at the very least we learned to follow instruction. In the grand scheme of things that is not a bad skill to have! We have also learned just a little bit about those of us that have disabilities. Some may be visible and some may be hidden, but we all suffer from some limitation compared to the "average". Anybody want to take a stab at defining the 'average'? So win, lose, or draw, the season has been a success.

Competition

[FAQ Table of Contents](#)

Q) What should new teams expect when they attend a tournament?

A) A long exciting day full of ups and downs. The big thing to remember is to just RELAX! Tournaments are a lot of fun. At the end of the day even if your team is not top scorer, they will have learned and participated in something really good and had fun doing it. A description of a 'typical' tournament can be found in the [Coach's Handbook](#) .

Q) Is there a National competition for those that place at the State level?

A) At the conclusion of the 2001, 2002, 2003, and 2004 seasons, the Director's Award winners at the state level were invited to the "Invitational" tournament. This tournament is held at the same time and place as the FIRST robotics (high school program) national competition. The invitational events have not been publicized in advance and therefore should not be counted on in the future. Check annually with FIRST. Be warned that the cost of sending an FLL team is anything but trivial and that each team is responsible for its own costs.

Q) What are the criteria used to advance a team from local to state?

A) It varies from state to state and even within a state. A small local competition may only get to advance a few teams. A larger competition may get to advance more. The actual criterion is also variable but is most often based on the overall score used to select the Director's Award. This score is a weighted combination of the individual scores. The weights for each competition are likely to be different. The following is a sample:

Robot Design	25%
Robot Performance	25%
Project Presentation	25%
Teamwork	25%

It is interesting to note that only 25% of the overall score is from the performance score! FLL is about being good at all aspects of engineering/technology.

Q) Can we attempt the same mission more than one time?

A) Yes. If you failed to complete the mission the first time you may try again, and again, and again.

Q) Can we complete the same mission more than one time?

A) Yes but why? If you completed a mission, you could still do the mission again, but you will only get points for doing it once. The table is scored by examination after the time has expired. The fact that you did the flobersnack mission ten times in a row will not be apparent to the score taker by looking at the final state of the table.

Q) How can we solve mission XYZ?

A1) This is the heart of the problem! If you ask you may get answers, you may get hints, you may get wrong answers, or you may get "gently flamed" (toasted?). The key is to have the team brainstorm the problem. Specific questions ("Can we whack the gizmo before touching the thingy?") are welcome.

A2) Try not to be swayed by what other teams post. Each team should do their own thinking. Following the crowd will not lead to revolutionary ideas.

Q) Is it possible to bring a TV tray table to put next to the competition table with our accessories on it?

A) This is tournament dependent. Within reason most will allow this. But be aware of the dangers. Beside the two operators there will be a referee, and possibly an MC and camera operator. These individuals all have the possibility to knock over a table. A plastic bucket on the floor is perhaps a better approach.

Q) Will we be subjected to a robot return penalty if we yank the robot back before it fully leaves the base?

A) No. The BONUS OBJECTS/LOSS rule states that you will not get a penalty in this case.

Q) Can our robot leave litter on the table?

A) There is no rule that prevents you from littering. A fairly traditional strategy is to discard appendages you do not need anymore. The downside to this approach is that a referee will not allow you to retrieve a jettisoned item (unless the robot is in possession of it) on a failed mission.

Q) Can we use the LEGO remote control during the competition?

A) Good try, but NO! It would not only control your robot but the other team's as well. It also violates the AUTONOMY rule and the spirit of FLL. A remote has benefits during testing in the pit area, but should be left there. Do not even bring it into the competition area. Remotes make referees very nervous. Nervous referees may confiscate the remote for the rest of the competition.

Q) Can we "pre-containerize" our own team's objects and substitute these when we come to the table?

A) No. You cannot come to the table with your own team's objects all built into a container and use that instead. All prep of this kind must be done with the tournament objects. Keep this in mind if you are planning elaborate containers.

Q) Do we need to prepare a formal presentation on our software and mechanical design?

A) Not usually. The judging is normally an interactive session between the judges and team members. Nothing says that a team cannot have a short overview to warm the judges up. In fact it could "peel back the onion" faster allowing the judges to see more detail and ask better questions. Just make sure it is SHORT. Be aware that some states (TX?) do require a formal presentation. Contact the regional organizer in advance.

Q) During a mission, part of my robot breaks off. Will the referee return the piece if I ask him or do I have to wait till the mission run is over to get the part back?

A) Most referees are instructed to hand broken robot pieces back to the operators as soon as possible. Parts that are meant to come off like projectiles or delivery containers are not returned. It is usually easy to tell if the separation was deliberate or not. The referee will have the final decision. Of course, a robot that falls apart a lot is not likely to win the "Robust Design" award!

Q) Can the robot touch the 2x4 wood borders without penalty?

A) Yes. Many teams use the wall as a navigation aid.

Q) Can a team request that the referee return the robot to base for

them if they can't reach it?

A) Absolutely, but they will be subject to the robot return penalty. In fact, before the match even begins a good referee will ask the team the best way to pick up the robot without damaging it just for this eventuality. A quick word with a referee that does not bring this up on their own might be a good idea. Your other option is to bring a stool.

Q) I read that programming is not allowed in the tournament area. Does this mean that there is time between rounds to return to the pits and make adjustments?

A) Perhaps. Besides the performance rounds, your team will be busy with their research presentation and technical judging. If these events do not interfere, you are likely to have time to make adjustments during the preliminary rounds. If you have made the finals, all bets are off. You may have as little as the time it takes to reset the table before you must compete again.

Q) What happens if our robot is not in the base when the time expires?

A) There is no penalty for not being in base at the end of the round. If you have a object still on board the robot, it will be scored as such. Some missions may actually give points for being someplace other than the base at the end of the match but this changes year to year.

Q) What do I need to bring to a tournament?

A) Bring:

1. A computer (laptop recommended) with whatever software(ROBOLAB or RIS) you are using to program the robot.
2. Extension cord and power strip.
3. All your programs pre-loaded on the laptop, robot, and on two floppies.
4. Printouts of your code if possible.
5. Your robot!
6. Your Mindstorms kit.
7. A programming garage.
8. Research presentation materials.
9. Team banners, tee-shirts, etc. to show your team spirit.
10. Money and/or food for lunch and snacks.
11. Camera and/or camcorder. Remember a) disable the flash b) do not use if the camera uses IR focus.
12. Spare batteries.
13. Parents to keep track of the team and to cheer.
14. A cell phone to call wayward parents.
15. Something to keep siblings busy. An FLL tournament is often a full day event.

Q) Can a team bring note cards or some kind of check list to the table?

A) Yes, the prohibition on non-LEGO parts does have to stop someplace! As long as the check list is not used in any way to directly affect the robot it is legal.

Q) Do we need to bring our table to the tournament?

A) Normally no. The tournament Director is responsible for having the right number of tables available. Of course, you may be asked to lend your table for the event. You may also be asked to lend your field pieces. But normally you can leave your table at home.

Q) Will there be any place to practice at a tournament?

A) Normally yes. Most competitions will provide one or more practice tables in the pit area. Some teams bring their mat and try to find a place on the floor to lay it out to practice. Some tournaments forbid this due to safety issues.

Q) I am wondering how the rounds are structured. The manual states that there will be a minimum of three rounds. Does that mean we will have three 2½-minute periods of time to accomplish all the goals or are we trying to achieve all the goals in each 2½-minute round?

A) Each of the three rounds is scored on its own merit. Try to accomplish as much as possible in the 2½-minutes provided.

Q) Does the 2½ -minute round include time at base changing parts etc?

A) Yes. There is no timeout mechanism. Parts changing and robot action all are charged against your 2½-minutes

Q) After being called to the table, how much time if any does the team get to setup before the match starts?

A) This varies between tournaments but you are likely to get a minimum of one minute. You may or may not be allowed to manipulate field pieces that start in the base during this time. You certainly may not touch anything outside the base.

Q) Our robot breaks at the competition table. Can we send up our mechanical guru to fix it?

A) Yes. You must tell the referee you are declaring an emergency and call up the extra people. The robot cannot be operated again until only two operators are present.

Q) Can my operators play "tag team"?

A1) Probably. Although the Participation rule specifically allows it, some tournaments shun the practice because of the physical layout. Find out in advance. If it is allowed, let the referee know in advance. As long as only two students are at the table it should be fine. Be aware that your approach may be problematic in a hectic environment.

A2) This great description is from Scott Kukshtel.

Last season, our team used the tag team strategy with much success. The year before we had used the 2-person strategy (also successfully), but there was some strong desire by other team members to participate, so we decided to try the tag team approach.

Out of our team of 10, only 2 decided not to participate, so we used four teams of two. Each team was assigned a mission "interval", in which they were responsible for bringing any necessary robot pieces, deliverables and/or jigs to the table, removing pieces no longer required, adding the new

pieces, and starting the mission. They would then leave the table and tag the next team in.

This approach also had the advantage of not cluttering the base with extra robot pieces/deliverables/jigs. All team members lined up by the table just out of the immediate competition area (there was a tape line on the floor) so that they could watch the action and be ready to go when it was their turn. The judges could also see all parts in their hands at all times.

I admit that it did take some practice to get it right, but no more than would have been required if it was just 2 team members doing the whole thing. It's also much better than having multiple 2-person teams where each team does everything for one round. This way, each team member could concentrate on 1 or 2 simple tasks, and you never get into the issue of one 2-person team being better than the other.

It also promoted much more of a team mentality and really helped to boost our team spirit. And with all team members watching, they all learned all the intricacies of all the missions. Which brings up another advantage of this approach (which we fortunately didn't encounter): With a single 2-person team, if one or both happened to be sick on tournament day, you are in trouble, whereas with a tag team approach, it's relatively easy to substitute.

To start, we created a detailed flowchart on a white board so that everyone could remember their roles, but after a few practices, it was all in their heads. Each member knew exactly which parts and deliverables they were supposed to be handling at all times, and they were responsible for those parts' integrity and correctness (ever seen a jig fall apart just at a critical moment as it is being fished out of a box on the floor?)

For a few more complex missions, we ended up not doing the 2-in/2-out switch during the mission, and instead left in one team member as a "crossover" in case anything went wrong during the mission and it needed to be restarted.

This year, our team is down to only six, but we'll probably still employ the tag team approach. Last year, we received several complements from table judges and other observers for our "choreography". I highly suggest trying it - it's really not as hard as it may seem!

One suggestion: if you take this approach, make sure that the team mentions it to the floating "teamwork judge", as sometimes they are too busy to actually see the team in action at the table, and this essential feature of the team may be overlooked.

Q) Can you modify your robot during a round?

A) Yes, as long as the work is done in the base, and the sum of all parts meets the legal parts rule.

Q) Can you change your robot between rounds?

A) Yes. In fact you could alternate between two designs if you wanted.

Q) Are the missions to be accomplished in any set order?

A) By default there is no set order that the missions must be done in. In some years a mission was given a special order. Either the robot had to do a set mission first, or points were given for a special final position. Look at the mission specifics for these overrides.

Q) What should we do if we see a referee not following the rules

properly?

A) This will happen at times. Referees are often volunteers that were only trained for a few hours and perhaps even that morning. It is therefore very important to raise rule issues with the referee immediately after the round. Calmly explain the situation. They will usually call the Head Ref over and get it straightened out. Usually you have enough chances to score well that the occasional referee mistake won't cost you. But I have had great teams completely hosed by table or referee flaws, and it is just part of what the kids have to learn to "take".

Q) What is a bonus object and why would they be taken away?

A) A bonus object is an item on the table that at the start of the match is worth points in its current location. If your team finds it necessary to manually return the robot to base, then the referee will remove a bonus object from the table. This will therefore lower your score. Bonus objects are a simple way for the referees to keep track of the number of robot return penalties.

Q) Can you please explain the issues behind scoring?

A) Your team will be evaluated in several categories. These may differ somewhat on the tournament you enter, but they are likely to include Robot Performance, Innovative Design, ???, and Teamwork. Only the Robot Performance score is an objective score. The others are subjective due to the human factor in judging. At the end of the tournament, the judges will deliberate and pick winners for each category. Their job is simple for the Robot Performance score having only to pick the best score. Their job is also fairly simple for the Director's Award (overall award). Here they sum the scores in each category and pick the top team. At this point their life becomes tough due to FIRST's policy that no one team should walk away with all the awards. The judges must then look carefully at the scores and team rankings to best allocate the awards. They must take into account numerous factors and special circumstances. This is not an enviable job.

As a result of these deliberations, a team that was ranked third in a particular category may well take home the first place trophy. As a result, tournaments will often not post the results for fear of opening Pandora's Box. The team that won may feel cheapened when they learn they were actually the third place team. A team that was second in one category and third in another may wonder why they were given the third place trophy and not the second.

But conversely not publishing the scores leads to hurt feelings when teams can't make sense of who won what and why. These are issues that FIRST and their tournament partners struggle with.

What does make a great deal of sense is for the tournament partner to release each team's score sheets. These sheets can be very valuable if written comments are included. They can be even more powerful if the tournament publishes the average score to allow some numeric comparisons.

Q) What amount of talk is allowed between teams during a match?

A) As much or as little as your want. FIRST is big into "competitive cooperation". They encourage it as much as possible. That is the whole reason for a shared mission. Be aware that the music may be so loud as to prevent any communications during a match. Talking to teams back in the pit is a great way to pass the time and share information. Of course heckling would be severely frowned upon.

Q) What kinds of questions do they ask the kids in the technical judging phase of the tournament?

A) What mission are you most proud of? Who figured out to do that? How many sensors do you use? Why are you using this sensor? How did you come up with your strategy? Why are you doing this before that? What is the best feature of your robot? What is the worst? What software are you using? Who wrote the software? How did you learn to do that? Who built this gizmo? What are some ideas that you tried that didn't work? How did your robot change from when you started to what it is now? Are there any features of your robot that you think are completely different from anything else that other teams have done? What is the triple integral of $y = \sin(2\pi \cdot f \cdot t)$ relative to time over the interval 0 to π ?

Q) Is it possible to solve all the missions in the time allowed?

A) Yes. Perfect scores are rare, but do happen. It is extremely rare to see a perfect score in November. In December/January a few perfect rounds will have been recorded world wide. By March, three or four teams attending the World Festival may be able to have a perfect score. You must temper this answer with the knowledge that the degree of difficulty changes each year.

Batteries

[FAQ Table of Contents](#)

Q) Can I use rechargeable batteries rather than alkaline inside the RCX.

A) Yes with a few warnings. Use batteries that are rated for 1.5 volts a battery or less. Anything more can/will damage the RCX. Most rechargeables have lower voltage than alkaline batteries. The performance of the RCX will therefore be different. Another issue with rechargeables (especially Ni-Cad) is that they provide constant voltage and then over a short period die quickly. Alkaline batteries die slower giving you some early warning.

Q) I have an RCX with an external power jack. Can I use it to recharge the batteries inside?

A) NO.

Q) We have heard of some teams buying cases of batteries. How many batteries will we need?

A) Depending on the robot, a set of batteries will last from 1 to 20 hours of use. A heavy treaded robot will gobble batteries at an alarming rate. A light wheeled robot will last much longer. Do not be alarmed at 'buying a case' of batteries. A case is normally 144 batteries or just 24 changes. A large team with several robots can easily use 24 changes in a season. See the following coach's manual for more details: [Handbook](#).

Q) Are the kids required to change the batteries or should/can the coaches change the batteries?

A) It's not very hard to change them, so let the kids do it. In fact, by having them change the batteries, they will consider twice before burying the RCX deep into the robot.

Q) What is the proper way to change the batteries in an RCX?

A) It is possible to change the batteries without losing the firmware if you use the following procedure.

1. Remember that they all face the same direction.
2. Use a magic marker to scribble over the old batteries while they are still in the RCX. This allows you to keep track of them should they get mixed up during the change!
3. Remove only one battery at a time.
4. Replace with a new battery as soon as possible.
5. Wait 5 seconds before changing the next battery. This allows the RCX to recharge its internal parts.

Having two team members work together to change the batteries may help keep the new and "used" batteries separate, as well as encouraging teamwork.

Q) How can you test your batteries?

A) In RIS, you can check the battery level by going to the settings area. In Robolab, open inventor 4, click on the project button at the top of the screen in the Robolab program, and then click on the interrogate RCX button on the drop down menu. The RCX battery level is displayed on the left side of the screen. You need to have your RCX on and in front of the serial tower for this to work.

Q) Can cold affect the performance of a robot?

A) Anything and everything can affect an FLL robot! Certainly the cold will greatly reduce the performance of a battery. Consider keeping your robot and batteries warm on that long car drive to a competition in cold climates.

Sensors

[FAQ Table of Contents](#)

Q) What is the touch sensor good for?

A) Lots of things. The simplest is a bumper sensor. Other more novel uses for the touch sensor are best left to be discovered by your team. Have them research on the web or look at the numerous Mindstorms books available. The Minnesota FLL High Tech Kids web page (<http://www.hightechkids.org/fll>) is another good resource.

Q) We have heard that resetting the rotation sensor is valuable. Please explain!

A) The rotation sensor can be reset to zero just like the trip odometer in your car. Being able to reset the rotation sensor can drastically simplify software. Both RIS and RoboLab have reset commands.

Q) We have connected a light/touch/rotation sensor to the RCX, but using the LCD display in 'view' mode cannot see any variation in the numbers.

A) The RCX probably does not know that you have a light/touch/rotation sensor attached to that port. Each sensor requires that the RCX 'listens' to it differently. Download a simple program that uses the proper sensor on the given port. Run the program once. The RCX should now know the proper configuration and give you the right results in View mode.

Project

[FAQ Table of Contents](#)

Q) Is the Project what was formally known as the Research Project?

A) Yes.

Q) How can I motivate my team to work on the Project?

A1) Do not allow them to work on the robot until certain milestones are met on the Project! Put the LEGO away and make the whole team work the Project.

A2) Remind them that they must do well on the Project to have a chance at the Director's Award.

A3) Stress the integration of the Project topic with the challenge.

Q) How are we supposed to present our Project findings to the judges?

A1) Any way you want! Some do a skit, others do PowerPoint presentations, tri-fold 'science fair' type exhibits, pre-recorded video, game show, and the list goes on. Just remember your time constraint (the new standardized time is 5 minutes) and bring everything you need with you.

A2) To give you specific ideas might be to limit your creativity. You could stand in a kiddie pool full of lime Jell-O in a bunny suit and sing a song if that gets the point across. Overly traditional methods are PowerPoint presentations and a skit. Personally, the Jell-O sounds more fun!

Q) How much time do we have for the Project presentation?

A) In 2003, FIRST standardized the time to 5 minutes. Most tournaments will probably comply, but there might be a few that do not. A very large tournament may only give you 3 minutes; smaller ones might allow you more than 5 minutes. Plan accordingly. When the tournament information packet is distributed (a few weeks before the tournament), the time allocation should be provided.

Q) Do we need to have a Project presentation to compete at a tournament?

A) No, but you will be out of the running for the Director's Award and of course the Project presentation awards. Although the kids may see the robot performance as the top prize, it really is not. The overall winner (Director's Award) is based only 20% on performance scores. The rest is teamwork, presentation, technical interviews and the like.

Q) Who can/will/should be present for the technical interviews and Project presentation?

A) This varies from tournament to tournament. Normally, because of room size issues, only the team and the coach are allowed in. Sometimes the parents are allowed to watch as well. In rare cases, only a part of the team will be allowed. Contact the tournament director for specifics. Regardless, only the team members can interact with the judges.

Q) Does anyone have any grading rubrics on how the Project will be graded?

A) Scoring rubrics are in the back of the Coaches Handbook (black and orange book from FIRST). But think carefully if and how you want to use them. You do not want to find yourself "teaching to the test".

Q) 'Do we need to build a robot for the research presentation?'

A) Unless the instructions specifically call for it, you don't have to build a robot for the research project.

General Programming

[FAQ Table of Contents](#)

Q) Should I use RoboLab or RIS for programming the robot?

A1) They each have their own advantages and disadvantages.

A2) Use RoboLab if you are using Windows NT or a Mac (RIS does not work on NT or Mac).

A3) Most coaches who have used RIS and RoboLab agree that RoboLab is more capable but a little less intuitive. Neither should prevent you from attaining high scores. If you have the time, try both and see what your team thinks. If most of your team already is used to one, go with that.

Q) What is Mindscripts?

A) Mindscript is the foundation of RIS. It is a text based language that the RIS user interface reads and writes. Your programs are stored as Mindscript files. A 'compiler' turns these files into code when you ask to download your program to the RCX.

Q) Can we use Mindscripts to program out robot?

A1) Using Mindscripts directly as your programming language is legal. Mindscripts is not called out as a legal language directly, but as it is the foundation of RIS it is allowable.

A2) Consider introducing Mindscripts to a veteran team in the off season to teach a 'traditional' text based language. Be warned that the error messages are not very good and it will take some time to learn.

Q) No matter how hard we try, we can't get a program to load into slot 1 or 2. What are we doing wrong?

A) By default, slots 1 and 2 are write protected. During firmware installation two generic programs are downloaded to slots 1 and 2. You can free these slots for your own programs by:

- In RoboLab: Start RoboLab, select the "administrator" button, go to "RCX settings" tab, select "unlocked" for program slots 1+2. Make sure that the RCX is on and in range of the IR tower when you do this.
- In RIS: Start RIS, choose "Settings" then click the "Advanced" button and unmark the "X" in program slot 1, indicating that slot 1 is locked. Apply the settings and return to the RIS top-level menu. The RCX needs to be on and in range of the IR tower.

Q) Which is better RIS or RoboLab?

A) This is a hard question to answer. There is no one right way to go. If there was, we would not have two choices!

There are three versions of RIS. 1.0, 1.5, and 2.0. There is little argument that RIS 2.0 is far better than 1.0 and even 1.5. If you have an old copy of RIS, getting an upgrade to 2.0 is highly advised unless you want to consider RoboLab.

There are several versions of RoboLab as well. These are 1.5, 2.0, 2.5, 2.5.1, 2.5.2, 2.5.3, and 2.5.4. For owners of 2.5, the patches to make it 2.5.3 are free and available on the web

(http://www.cceo.tufts.edu/robojabatceo/resources/Service_Pack/default.asp). Version 2.5.4 is not a patch and requires a purchase from Pitsco. The advantages of 2.5.x over 2.0 are the introduction of the automatic wiring feature, support for the USB tower, and support for the new firmware features.

Here are comments from users of the forum. Some editing has taken place.

I think that RIS 2.0 has a definite advantage over RoboLab due to the simplicity of the programming. When I competed in Volcanic Panic last year I found that it can take days of practice to master RoboLab. I highly recommend RIS 2.0 over RoboLab and I hope you will use it.

-Thoughts from a 10 year old RIS user

RoboLab 2.0 is far more powerful than RIS, though it lacks the capability of some of the new firmware. RoboLab 2.5 supports the new firmware and the USB tower. (Both versions are legal for FLL). From my perspective as a programmer for over 15 years, RoboLab offers a better analogue to "real" programming than RIS. I am also annoyed by limitations of RIS that force multiple tasks, each watching a different sensor, etc. than straight-forward sequential programming that allows "if/else" type constructs, etc. using the sensors. I firmly believe RoboLab will leave students better prepared for programming with more traditional languages than RIS. There are FAST line following algorithms that can be coded in RoboLab that cannot be coded in RIS.

-Peter

RoboLab is available from Pitsco through their latest catalog. Major differences: support for the USB tower, programming the RCX, the Scout, and the Dacta control lab, auto-wiring, support for the new RCX firmware functionality, better Internet + direct control, support for a camera as a sensor (via infrared messages and a PC interface to the camera). The functional icons have been more hierarchically arranged and the interface is a little "cleaner" w.r.t. picking commands. All in all, a nice improvement, but not critical to FLL unless you need the USB tower (a big help for me since my laptop handles USB better than serial).

-Peter

If you are Macintosh based, RoboLab becomes a requirement as RIS does not run on a Macintosh.

-Mark

RoboLab is a vast improvement over RIS. I have programmed for the past two years with the RIS programming, and this year we bought the Tech upgrade just to get RoboLab. It is vastly more powerful, but harder to use. You have to learn how to string things together (it doesn't attach programming blocks together automatically like RIS), and figure out how to program some of the sensors, but it much more productive. I recommend RoboLab, but allow some training time for your programmers.

RIS was designed to make it easy to program, make it impossible to create a "compile error", and to minimize technical support. As a consequence of this design criterion, some programming tasks are more difficult to program using RIS, and some are downright impossible. This competition is for the kids, and my opinion is let the kids choose the programming environment. My team chose RoboLab because they understood it, which is more important than any theoretical "programming power".

RoboLab is more powerful and versatile, but it takes longer to learn, if you have gifted / talented kids or older kids that can work on it by themselves - go for it.

-A second year coaches opinion

RoboLab is my choice. RoboLab is dual platform, and more powerful and versatile. My recommendation is buying the teacher manuals from LEGO. I have the kids begin by reading and working through examples in the manual. They quickly become familiar with the interface and logic. There is more room to grow with RoboLab. There is a reason LEGO markets RIS to the toy stores and RoboLab to the educational community.

Our kids are coming along with RoboLab. We used it this year because one of our kids uses a Mac and he was the most interested in the programming, so RIS was not an option. It does take a bit longer to figure it out, but now that they have been using it for awhile, they like it. On the average our team has older kids than the team we coached last year, so RoboLab hasn't been a problem.

Q) How can we make our robot follow a black line?

A1) This question is beyond the scope of a FAQ. Following a line is more art than science. It is best to have the team work this issue early to come up with an algorithm that works on your robot. A search of the web will turn up lots of raw information you can use to get started.

A2) Have you read "Toeing the Line" or the "Coaches' Primer"? Both documents are available for download at <http://www.fll-freak.com>.

Q) How big can each of the 5 programs be?

A) Individual program slots do not have a size limit. Memory management inside the RCX divides up the total memory on an as needed basis. There is no simple answer to how large all programs can be. It would depend on the firmware version and other factors. The bottom line is that it would be fairly unusual for an FLL team to exhaust available memory. What is possible is that the memory becomes fragmented to the point that there is no one chunk large enough for your program. The solution is to delete all programs in all slots and start fresh. Also note that in RIS there is a maximum size for each stack. This is not an RCX issue but a limitation of RIS. There are ways around this problem.

Q) How can you program a "break" to prematurely abort a loop?

A) Use the traditional "abort flag" mechanism. This is hardly "simple" but it works. Here is the concept in text form. Converting to RIS or RoboLab is left as an exercise to the reader.

```
Set "container/variable of choice" = 0
Loop while "container/variable of choice" = 0
|   (Body of loop goes here)
|   IF on 'normal exit criteria'
|   |   Set "container/variable of choice" = 1
|   End IF
|   IF 'abort exit criteria'
|   |   Set "container/variable of choice" = 1
|   End IF
End loop
```

The flag ("container of choice") starts out cleared. When either the normal or the abort exit condition occurs the flag gets set, and the loop exits.

Q) How can you program a loop to terminate on one of two conditions not just one?

A) Look at the above question and think "second exit criteria" rather than "abort exit criteria".

Q) What should be in the program's pre-amble?

A) The preamble is a section of code that you execute first before you start in on your mission code. Not all teams will use a pre-amble. The preamble would set up variables and sensors. Other things include:

- 1) Display the team number or some other number to the display. This lets you know it is YOUR code and not injected by mistake by some other team.
- 2) A 1/2 second delay. This allows fingers to get out of the way before the robot lurches off.
- 3) Initialization of needed constants.

Q) Our robot is not stopping at the specified rotation sensor count. Why?

A) To understand this problem, you need to understand that the RCX will not be able to evaluate the stop

condition for each and every rotation sensor count. It may only see counts of 97, 98, 101, 102, 105, ... If you wait for the rotation sensor to equal 100 it will never stop! What small change to the stop condition can you make to compensate for the occasional skipped number?

Q) We can't download programs anymore!

A) Look for these issues:

- Ensure the RCX is turned on (you would not believe the number of times this has happened).
- Ensure the tape or LEGO cover on the IR port is removed (if you had it covered in the first place!).
- Reboot the computer.
- Run the laptop from its power cord not battery.
- Ensure that the short/long range switch (RS-232 tower) matches the option set in the software.
- Check that the serial cable is plugged in ok.
- Check that the 9 volt battery in the serial tower is in properly.
- Replace the batteries in the RCX.
- Replace the battery in the serial tower.
- Check for the green light to come on in the tower.
- Check that the right serial port is selected.
- Remove a battery from the RCX for 2 minutes and try to re-install the firmware.

Q) My RCX seems dead. What can I do?

A) On occasion the RCX gets very confused and needed to be 'rebooted'. Simply remove a battery and wait 2 minutes by the clock. Reinstall the battery and download the firmware. This normally takes care of most problems. If this fails, you may need to replace the RCX. LEGO is very good about sending replacements.

Q) Repeated insertion and extraction of the serial tower plug is damaging my computer's serial port connector. What can I do?

A1) The molded plastic connector on the serial tower is fairly tight. It can do damage to a weak computer connector. A solution is to replace the cable with a standard 'straight through' 9 pin female to 9 pin female serial wire available at most computer stores. Bring the tower with you to make sure the new cable will fit properly.

A2) Purchase a serial extension cable.

A3) Replace with a USB tower.

Q) Can my team use NQC, Java, or ...

A) No. The only legal languages are RIS, and RoboLab of any version.

Q) The RCX sometimes does not behave properly.

A) One possibility is mismatched firmware. If you are using RIS and RoboLab or different versions of RIS you may experience this problem. You may have downloaded an older less capable version of the firmware into the RCX from one program and trying to use advance features of the other. The solution is to pick one language and reload the firmware from that program only.

Q) FLL allows 2 light, 2 touch, and 1 rotation sensor, but there are only 3 input ports on the RCX. What gives?

A1) Who said you have to use all the sensors? You can pick what you want.

A2) A trick allows you to use all the sensors at the same time. It's called piggybacking. Some sensors can be connected to the same port on the RCX. You can connect both touch sensors to one port and the RCX will

report 'touched' when either switch is depressed. You can also piggyback a touch sensor and a light sensor. As long as the touch sensor is not depressed, the light sensor will report as it should. When the touch sensor is depressed, the RCX reports that it saw a light value of 100. Unless you are working close to the sun, you can therefore tell the difference between a light sensor value (<100) and the switch being pressed (=100).

Q) Why does our robot not stop when it reaches the end of our program?

A) The RCX does exactly what you tell it to do. If the last motor instruction was to turn in on, it will stay on even after the end of the program. The solution is to stop all motors at the end of all programs.

Q) This year's competition has more than 5 missions but the RCX can only hold 5 programs? What are we supposed to do?

A) You can:

1. Only run 5 missions.
2. Combine 2 missions into one program. Do ABC and XYZ before returning to base.
3. Have one program slot run two missions based on some external input. Hint: Remember a program can branch as a function of a sensor input. Just remember not to violate the green button rule.
4. Brainstorm the problem and come up with an even better solution.

Q) Is there such a thing as a global variable/container that can be used across all 5 program slots?

A) Variables/containers are a function of the firmware and are both global and persistent. Persistent indicating that they keep their values from one run to the next. In RoboLab the containers support both global and persistent features.

In RIS, variables are reset to zero before your program starts rendering the firmware global and persistence useless. You can modify this default behavior by modifying the default macro in RCX2DEF.h located in the script\include directory of the RIS program directory. The default is to define:

```
var MyVariable = 0
```

Which both defines the variable and initializes it. If you change it to:

```
var MyVariable
```

It will only define the variable and not initialize it. Now you may use this variable as a global variable. Just make sure you initialize it someplace, typically in program 1.

RoboLab

[FAQ Table of Contents](#)

Q) My new computer hangs when I use a USB tower with RoboLab!

A) The new "hyperthreading" technology in the new batch of Intel processors is not compatible with the tower's USB driver. The solution is simply to download the latest driver from LEGO at this link

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

Q) What versions of RoboLab exist?

A) RoboLab has had many revisions. Some revisions have been patches to a previous version and others have

had to be purchased. This table details the principle changes.

RoboLab Versions	
Version	Comments
1.0	Disk
1.5	Patch from 1.5 available from LEGO
2.0	Disk
2.5	Disk
2.5.1	Patch.
2.5.2	Patch. Fixes know vision, data logging, CLI, and other problems...
2.5.3	Disk or patched from 2.5.2. Fixes known bugs in 2.5.2
2.5.4	Disk. Directly supports Mac OS X. Can read old files.

Q) RoboLab hangs up at times. What is wrong?

A) It has been reported that on some systems RoboLab will hang up. The problem may occur when the system goes into a powersave mode. If you are running a laptop, turn off hibernation and/or powersave modes.

Q) What documentation is available for RoboLab?

A1) Visit the Minnesota FLL web page at www.hightechkids.org/fll. They have several good tutorials on the use of RoboLab.

A2) LEGO Dacta sells a series of manuals aimed at teachers using RoboLab in the classroom. In the past, the first manual came with the copy of RoboLab you get when purchasing the technology upgrade kit from FIRST. This manual does have some useful information but may not be worth the hefty price asked.

A3) Tufts University was the creator of RoboLab. They have a huge reference manual available online at <http://www.cceo.tufts.edu/robolabatceeo/Resources/documentation> It is a reasonable reference guide but cannot be considered a tutorial or user's guide.

Q) I need documentation beyond what comes with RoboLab. Are there any tricks to help us?

A) The following tricks are available:

1. Use spacebar to select between the pointer and thread tool.
2. Use the tab key to select between pointer, hand, thread, and text tools.
3. Press control-b to remove all the broken wires.
4. On occasion you will see array tools pop up in some menus. These have no benefit in RoboLab. They are artifacts of the underlying LabView language.
5. Light sensors can be used in raw (unpowered) mode. You will need the latest 2.5.1 version or the 2.5.1 patches from Tufts (<http://www.cceo.tufts.edu/RobolabatCEEEO>) The specific patch needed are the "New Generic Sensors". According to FIRST, the patches are legal to use.
6. Using the pointer tool, you can single click on a wire to select just that segment.
7. Double click on the wire and you select the full connection from source to destination.
8. Triple click on a wire and you select the entire tree (one source to many destinations).
9. Quadruple click on a wire and your finger gets tired!
10. 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.
11. 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.

12. 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.
13. You can move graphic items forward or backwards relative to each other using the tools on the menu line.
14. You can arrange icons using the "align and distribute" commands.
15. Clicking your right mouse button while holding down the shift key, gives you a menu of cursor tools. This provides easy access to the eye-dropper and paint brush tools.

Q) How can we speed up downloading our programs?

A) If you use RoboLab and use subroutines, you have an opportunity to speed up your downloading. First you must understand that each program slot has space to define 8 subroutines. These subroutines are available to be called by the program (and tasks) loaded into the associated slot. Next you need to understand that the subroutines are persistent. That is, they stay defined in memory until you delete them or you remove the batteries from the RCX. With that said, you can download your subroutines to the program slot in one download session and your program in another session. The following description of this most ingenious idea comes from John Wanninger.

"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 "subs.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 download the program to it. 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 download it to program slot 3. Note that this vi does NOT need to contain any 'create subroutine' functions. Those previously created subroutines will always be available in that program slot (#3 in this example) in the RCX until they are either deleted (with the 'delete subroutine' function), written over(redefined by a new 'create subroutine' of the same number), or the RCX batteries expire."

Q) What happens to a task that you stop and then restart?

A) According to the Mindstorms documentation, the stopped task will be restarted at the beginning and not where it left off.

Q) Is it possible for a subroutine to call another subroutine?

A) No. Subroutines can only be one deep.

Q) Why are there only 5 power levels in RoboLab when RIS has 8?

A) The icons for the power levels in RoboLab are nothing but a sub-vi wrapper around a numeric value. The legal numeric values are 0 to 7. Notice that power level 0 is not off but rather the lowest power level. This means you can wire a numeric value (0-7) when asked for a power level, or a RoboLab icon (1 to 5). The mapping between RoboLab power levels and the low level numeric constants is:

RoboLab Icon	Numeric value
1	0
2	1
3	3
4	5
5	7

Q) Why do our programs go nuts when we add a jump into a sub-vi?

A) A program cannot have more than one set of each colored or numbered jumps and lands. When you place a jump/land combination into a sub-vi and use that sub-vi multiple times, you actually have multiple jumps and lands of the same color. The programs therefore get very confused.

Q) How do you change the background color in RoboLab 2.5+?

A) Bring up the palette of cursors. Pick the eye dropper. Use the eye dropper to pick up some white from one of the icons. Now switch the paint brush. Pick on the background and voila!

Q) Why would you want to change the background color in RoboLab anyway?

A) Because you can(!) and because:

1. A pink background on Tuesdays and orange on Fridays is about as much fun as is legal for a minor!
2. Changing the color to blue is always a good laugh since you can't see half the wires!
3. A white background uses less ink on a color printer.

Q) In RoboLab, what is the difference between a subroutine and a sub-vi?

A) A sub-vi is a shorthand method to make your code more readable and re-usable. Code that is defined in a sub-vi is expanded each time it is called. For those in the know, these are macros. There is no economy of code size.

A subroutine shares the trait that a collection of code is aggregated into one place. But unlike the sub-vi, the aggregate code is not expanded. The subroutine code only exists once in the program. When the program goes to execute the aggregate, it remembers where it was, jumps to the subroutine, executes it, and then jumps back to the starting point. There is economy in size.

Q) In RoboLab, what is a sub-vi and how do we create them?

A) A sub-vi is a short hand notation. It allows the user to represent a series of icons with a single icon. A sub-vi drastically reduces the number of icons displayed on the top level diagram. Sub-vis are not subroutines in the true programming sense. They are equivalent to a macro. Each time a sub-vi is used in a program, all the icons it represents are substituted before the program is downloaded. You can create a sub-vi quite simply. Use the selection tool to highlight all the icons that you wish to aggregate. Next select "Create Sub-vi" from the edit menu. The icons will be replaced with a new icon. Double click this icon to see the original icons. Make sure to perform a "Save As" on the sub-vi to save it. This sub-vi can now be used many places in your own program. Be aware of a minor bug in RoboLab. The sub-vi diagram will have begin and end terminals. The terminal labels are swapped, but the program will function properly. Use the text tool to rename the terminals. A sub-vi can accept parameters. When creating the sub-vi, do not select numeric constants that you wish to be parameters. When the sub-vi is created, connectors will be generated that you can tie to.

Q) Why can't we seem to be able to create a subroutine using RoboLab 2.0?

A) Subroutines are part of the 'extras' that can be separately activated. Look under the 'Project' menu for this option. Activating the 'extras' will give you lots of new icons to explore.

Q) How many containers are available in RoboLab? It looks like only three (Red, Blue, and Yellow)!

A) Yes there are quite a few available but you will need to use the generic container and tie a container number to it. If you are using RoboLab 2.0 you will need to first install the extras to gain access to the generic containers.

Q) RoboLab keeps crashing after installation.

A) Make sure you reboot the computer after the installation and before use.

Q) I have a copy of RoboLab 2.0. Should I upgrade to 2.5+?

A) An upgrade might be in order. Some new features were added to 2.5+ that make it very nice for a veteran team. The first is the 'autowire' feature that automatically wires icons together when they get close to each other. 2.5+ also supports the USB tower. Besides this 2.5+ has a host of new command icons that might be valuable. 2.5+ also supports sensors in un-powered mode and other advanced commands.

Q) In RoboLab, how can I tell what an icon does?

A1) Double click the icon and the front panel for that icon should appear. Often, help information will be provided.

A2) Enable "Show Help" from the menu help command. Hovering the cursor over an icon should pop up a help window.

Q) Can RoboLab be run under OSX on a MAC?

A) Yes. RoboLab just needs to open in Classic (which happens automatically).

Q) How does one have a fork within a fork without getting programming errors?

A) You must open and close the forks in a certain order. The last fork opened must be the first fork closed. In ascii art:

```

-----<-----<----->----->-----
.....|.....|.....|.....|
.....|.....|.....|.....|
.....|.....+-----+.....|
.....|.....|.....|.....|
+-----+

```

The < and > are fork opens and closes. Ignore the dots, they are needed to keep the picture lined up. If the vertical line to not match up, cut the text and past it into NotePad or other editor that uses a fix width font!

Q) Why does Robolab only have a few colored containers?

A) The colored containers are just shortcuts for generic numbered containers. If you use the generic container and tie a unique number to it, you can have many more containers. The first few numbers (1, 2, 3, 4) correspond to the colored variables. But you can use variables up to something like 35 without problems. Just be careful. The higher variable numbers are special purpose variables. They will tell you what iteration your loop is in and other interesting things, and hence using them as a variable would be bad news!

Q) Is it possible to add comments to a RoboLab program?

A) Yes. You can use text boxes. Once created, 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. - I haven't found a way to set the text box color other than transparent and still see any underlying code in Robolab 2.5.

Q) Once a subvi is created is it possible to adjust the connectors?

A) Yes it is with some dangers. It is often just as easy to start fresh use copy and past functions to pull from the old subvi. But for the brave:

1. Opened Sub VI in question.
2. Right clicked on Sub VI icon in upper right corner of the Panel window and selected Show Connector.
3. Right clicked on Sub VI icon (popup-menu) again and picked Disconnect All Terminals.
4. Selected (Popup-Menu)/Rotate 90 degrees until the terminals were in the correct orientation.
5. Left clicked (using the wiring tool) on each terminal and the appropriate component in the panel window in turn.
6. Right clicked on the Begin and End terminals and made them required rather than recommended.
7. Edit the text boxes from the default "Numeric" type text to something useful like "Power" or "Light Level" or "seconds".
8. Saved the Sub VI. Using similar ideas you can even add new terminals. This is left as an exercise for the reader.

Q) Why does the value of our blue container get destroyed when using a loop inside a subroutine (not subvi)?

A) You have found a fascinating bug in either the firmware or in Robolab 2.5.2. The loop counter is normally assigned to container 47 when used inline or in a subvi. For some reason it gets assigned to container 1 which is the same as the blue container when used in a subroutine. Either to not use loops inside subroutines or stay away from the colored containers.

Q) RoboLab hangs up each time I try to download a program.

A) The old tower device driver that comes with RoboLab is not compatible with the new Hyperthreading technology. You can download a new driver from here:

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

RIS

[FAQ Table of Contents](#)

Q) How can we overcome the stack size limit in RIS?

A1) Here is a relatively easy way to overcome the stack limit size. You create multiple stacks that trigger each other using a variable. The example here is for three stacks. You want the main stack to run and have it switch to Stack1 at the end, then at the end of Stack2 you want the program to terminate. At the beginning of the main stack (the one that you normally program in) set variables jmpStack1 = 0 and jmpStack2 = 0. Now create 2 variable sensor watchers. One will watch for jmpStack1 = 1 and the other jmpStack2 = 1. At the end of the main

stack, set jmpStack1 = 1. This will trigger the sensor watcher to run its stack. The first thing you should do in Stack1 is set jmpStack1 = 0 so you don't forget. At the end of Stack1 set jmpStack2 = 1 and set jmpStack2 = 0 at the top of Stack2. The effects of sample time using this method are not known, but assumed minimal. Test it before doing anything critical.

A2) Create your programs with a hierarchical structure using Myblocks. Break the mission down into sections and program each section as a separate MyBlocks.

Q) We can't get RIS to recognize that we have a rotation sensor!

A) You must enable it in the setup under advanced.

Q) RIS 2.0 programs that rely on small times do not work after being saved.

A) A bug in the software converts all times of 0.01 seconds to zero seconds! The only know workaround is to edit the times each time the program is loaded.

Q) Should I upgrade from RIS 1.0 to 1.5 or 2.0?

A) Strongly advised. The new version is much better. RIS 2.0 is more powerful and is well worth the upgrade. MyBlocks are stronger. Use of variables is great.

Q) How can we get a printout of our program in RIS?

A) There is no simple print capability in RIS, but you do have three options:

- 1) Text printout: Drill down the directory structure into the RIS program directory. Continue into 'user', then into the directory with the name used to login, and finally into 'vault'. Here you should find files with your program names. These files can be opened and printed from any text editor.
- 2) Graphical printout: Use Control - Print Screen to capture the display with your program in it and paste into a word document (or other graphical program) and print them from there.
- 3) Hybrid printout. The web page http://www.inpharmix.com/cgi/lego/lego_listing.pl has a program to convert RIS programs into printer friendly web pages!

Q) When we make a change to a MyBlock in one program, then load another program that uses the same MyBlock, we must rename the MyBlock. How can we solve this problem?

A) This is a bug/design error in the RIS software.

Q) Our programs get corrupted when using small numbers.

A) RIS 2.0 allows you to specify a time constant of 0.01 seconds for motor duration. If you use this value, the program is no longer editable. If you are in this situation, you can change the time constant to something else while you edit the program than change it back to 0.1 just before you download the program.

Q) How can we add comments to our code?

A1) You can use MyBlocks to combine code into a single labeled icon.

A2) Yet another good reason to upgrade to RIS 2.0. Under 2.0 there is a comment feature under the Edit menu.

Q) Using RIS 2.0, my kids created a "WAIT UNTIL" block waiting for the rotation sensor counter to be greater than a value. It works fine with a fixed number. However when using a variable, one time in

five the loop will not terminate.

A) An answer to this question is still needed. Authors welcome.

Q) Does RIS 2.0 have jumps?

A) Jumps or gotos are not part of the RIS language. But you can use conditionals and repeat blocks to do the same thing.

Mindscripts

[FAQ Table of Contents](#)

Q) Please tell me something about Mindscripts!

A) Mindscripts is the foundation for the Robot Invention System (RIS) software. Under the pretty pictures, RIS uses Mindscripts to store its programs and to convert them into code that can be downloaded to the RCX. It is possible to bypass the pretty pictures interface of RIS and program directly to Mindscripts. In doing so, you gain access to much more capability at the expense of a more complicated text based language.

Q) Is it legal?

A) This legality of Mindscripts has been debated in past years. The final verdict from FIRST is that it is legal. It is not called out explicitly as a legal language due to it being a part of RIS that is listed.

Q) What resources are available?

A) The following section is courtesy of Jon Gilchrist

I've been playing with Mindscript since last FLL season. Although I like the text-based language, it is not very user friendly. The kids on my team have decided to stick with RIS code.

In searching for resources, I have found precious few sources of info about Mindscript. The best by far is the documentation included with the SDK. You can get the SDK here:

<http://mindstorms.lego.com/sdk2point5/default.asp>

Be sure to read both the pdf docs and the help file.

One other source is from a group in the UK called Robofesta. They have a tutorial, though it is incomplete.

<http://robofesta.open.ac.uk/lsc/>

There is a website aimed at Spybotics that also includes programming in Mindscript. It doesn't like browsers that aren't IE.

<http://users.ncable.net.au/~blane/smartParts/>

Other Tools:

An essential tool, and one that is useful even if you don't use Mindscript or NQC, is BricxCC. The virtual remote and virtual joystick are indispensable, and the piano is great fun.

<http://bricxcc.sourceforge.net/>

That is the bulk of the information that exists. Here are a few tidbits that I have learned:

1. If you write something in Mindscript, make sure you have a backup copy before you open it in RIS. RIS will re-write your code, renaming everything and basically hosing up formatting. It will also wrap everything in "try..." commands.
2. If BricxCC or the compiler is giving you an error, look at the line after or before where it says the error is. Sometimes a typo will trigger an error on a different line.
3. Learn to use Macros and #include files. They will make your life easier and your code much easier to debug and understand.
4. Once you get a macro or code block that works, don't mess with it.
5. Create a macro that initializes everything, and use it at the beginning of your program.
6. You can write Mindscript code in any text editor and transfer it to BricxCC using any reasonable means. I program on my iPaq, and store programs on CF and SD memory cards. I can program in line at the grocery store, soccer practice, just about anywhere.
7. Learn to use Datalogging. It is about the most useful debugging tool that I have found. You can record sensor values at different times, and then examine them later. Figure out exactly what your light sensor or rotation sensor was reading!
8. Use lots of comments. Use a block at the beginning to explain what the program does and what uses and what dependencies it has. This will be important 3 months later when you try and figure out what the program does.

Robot

[FAQ Table of Contents](#)

Q) We have switched from time delays to using the rotation sensor, but have still not improved our accuracy. What are we doing wrong?

A) When properly understood the rotation sensor can greatly improve your accuracy. Unfortunately there are lots of pitfalls. Here are some things to consider during your experimentation.

1. Rotation sensor deficiencies. There is a known problem with the rotation sensor. If the axle rotates too fast or slow, the rotation sensor will lose counts. The best range is from 50 to 300 RPM. 300 RPM is the speed of an unloaded motor.
2. Speed. Assume you are driving at 60 MPH down the highway and slam on your brakes when your odometer clicks over to some magic number. Because of the car's inertia, it will take you some distance to stop. The same experiment at 5 MPH and you stop much sooner. The same thing holds true for a robot.
3. Battery power. Closely coupled to speed is battery power. As batteries die, the robot will start traveling slower and will affect the accuracy.
4. Stopping method. Using the car analogy, the difference in accuracy is much better if you slam on the brakes rather than just coasting to a stop. Make sure you brake the motor and not 'float' or coast it.
5. Sample rate. The RCX cannot watch your rotation sensor all the time. It has other jobs to do. It may be doing something else the instant the rotation reaches the terminal count. When the RCX 'finally' does look at the rotation sensor count, the robot will have traveled some distance. This will cause the distance the robot travels to vary by some small amount each time.
6. Peeling out. Robots that start and stop quickly may actually lose traction on the mat.
7. Non deterministic effects. Friction is not deterministic. Each run, the friction gods roll their dice. This

- causes fluctuations from run to run. This is especially nasty during turns when wheels scrub sideways.
8. Resolution. The basic resolution of the rotation sensor is 16 counts per revolution of the axle that pierces the sensor. If the wheel is on the same axle, your best possible accuracy is 1/16th of the wheel circumference. Not too bad if you are using the tiny balloon wheels, but fairly bad if you are using the large 'bicycle' wheels. A trick advanced teams use is to gear up the rotation sensor so that its axle rotates more often than the wheel axle. This increases the basic resolution of the robot. Just be careful not to rotate it faster than 300RPM.
 9. Full halt. When programming multi leg paths, place a short pause (1/10th to 5/10th sec) between legs. This allows the robot to come to a full rest and makes it more repeatable.
 10. Slowing down. What would you think of a driver that went the speed limit until they hit a red light and then slammed on the brakes?

Q) What is the LEGO lamp used for?

A) Many things including:

- 1) Decoration.
- 2) Debugging aid. Much like you would use a beep in your program to indicate whether the code actually got to a particular point, you could tell the light to light instead.
- 3) Used in combination with a light sensor and pulley wheel you can have a makeshift second rotation sensor.
- 4) Something nobody has thought of, but will revolutionize FLL.

Q) Are lubricants allowed?

A) No. It is quite explicit in the rules that lubricants are NOT allowed. Be aware that many lubricants will attack the plastic used in LEGO elements, rendering your robot a horrible mess. If you are trying to reduce friction on an axle bearing, design the support properly. An axle supporting the weight of a robot should not pass through just one beam.

Q) There is a rule against stickers and the like on the robot. What can I legally do to mark ownership of major LEGO parts?

A) The ban on stickers was instituted to prevent their use as structural elements and to keep the robots looking LEGO-ish. Although it is technically illegal, a small sticker with the team name on an RCX is not likely to cause problems. Make it small and obvious that it is not aiding in the structural design of the robot. Better yet (and now even legal!, see rule 5), use a Sharpie pen to write the identification mark in a hidden spot. Make it obvious that the mark in no way is affecting the behavior of the LEGO or the robot.

Q) Our robot does not drive straight. What is the matter?

A1) This is a constant irritation with FLL teams. In fact, it is very rare for the average robot to drive straight. A full answer to this question is beyond the scope of the FAQ.

A2) Nothing is the matter. Remember that no two motors are alike! Try to find the two motors that are most similar.

A3) Work with what you have. "When life gives you lemons, make lemonade".

A4) Brainstorm with your team to determine what all the factors might be that would cause the hook. Then run some experiments to isolate the problem.

A5) Some information is provided in the FLL Team Manual.

A6) Read the document [Building LEGO Robots for FIRST LEGO League](#)

Q) I understand that I can minimize the 'not driving straight' problem by matching motors. What is that and how do I do it?

A) Matching motors is simply trying to find the two motors that are the most similar in behavior. To find a

matched set, place the same sized gear on each shaft. Turn the motors on in the same direction. Now face them towards each other and let the gears mesh. If the motors match, there will not be any movement between the motors. By movement I mean that one motor will not try to walk around the other. The more motion between the motors, the more difference between speeds.

Q) We're having trouble with the rotation sensor binding up the motor. What are we doing wrong?

A) Don't snap the rotation sensor into your robot. Allow it to float on its axle. The wire should be enough to prevent the sensor from moving with the axle. Snapping it into place seems to cause binding, probably because the hole isn't in precisely the right place. If the wire is not enough to constrain the sensor, try a belt or rubber band it gently into place. A rotation sensor that flops around will not be as accurate.

Q) Why does our robot travel different distances for the same amount of time?

A) Because of many reasons. The largest being that your battery level is dropping and the motors are not working as hard or fast. Switching from time to using the rotation sensor is one way to minimize this problem. Other issues are varying friction or other effects that change from run to run.

Q) What is better, wheels or treads?

A) It all depends. Each has pros and cons. Consider making a single robot that can switch back and forth between wheels and treads. Run various experiments to see what makes sense. All things being equal, treads require more power than wheels and tend to kill batteries faster.

Q) What is a light sensor good for? Why do we need one or even two?

A) The prime use for the light sensor is to follow or sense lines on the mat. So far, each year the primary tasks have been located on or near lines that can be followed. The light sensor is the device that lets you do this. Some teams will even use both light sensors to optimize their line following algorithm. There are other ways to use a light sensor and that is why you get two.

Q) We have commanded both motors to go forward and yet the robot spins in place. What's wrong?

A) Nothing is wrong. Forward and Backwards are very arbitrary. How the wheels turn for a forward command is dependent on the mechanics and the wiring. To get the robot to go forward, rotate the connector on either the motor or the RCX port 90 degrees on the offending motor.

Q) Why does our robot not stop at the proper rotation count?

A1) The ability of the robot to stop on the exact count is primarily a function of how fast your robot can stop. If you have a fast robot, it may take a second for it to come to a stop. During this time you may travel several inches and continue to increase your rotation sensor counts. As batteries die, the robot slows down and now the robot will not travel the same distance during its stop. Slow robots are not affected as much. Another issue is how often the RCX checks to see if your rotation sensor count has been met.

A2) For those using RoboLab there is a 'feature' in the Wait For icons that introduce some latency and hence add to the overshooting problem. If you double click the icon, it will open up a subvi. Do NOT edit this subvi. If you do, it will affect all your programs and can only be fixed by reinstalling the software from disk. This subvi is what RoboLab does to accomplish the wait. It is a 'spin loop'. In other terms, it sits in an infinite loop testing the sensor for a particular condition. When true, it bails out of the loop. The 'problem' is the authors put a 2/100 second pause inside the loop. This means that you will not sample the rotation sensor as fast as possible. The solution is to make appropriate modifications to copies of these icons. This works great for the light sensors and touch sensors. The rotation sensor has the added complexity that it can go both positive and negative.

Q) What type of navigation do teams use?

A) Having a good way to navigate to the missions is one of the key aspects of the table performance rounds. Different teams use differing approaches that include:

- "Dead reckoning" with the rotation sensor to help provide more accurate odometry can work very well for robots that are precise in their movements. Through careful construction and appropriate gearing, some teams manage to build robots with a high degree of precision and can use dead reckoning with the rotation sensor and alignment jigs.
- "Physical orientation" is another approach. The robot can bump itself up against walls and/or mission models in order to align itself in a known location. This can be used to "reset the error" that has built up after doing a few turns or traveling a long distance by dead reckoning.
- "Feedback orientation" uses the external sensors (touch sensor and light sensor) to interact with the environment and figure out where one is by finding features such as lines, craters, or light spots on the mat, or by finding where physical objects are (the walls, mat, or mission models).

Winning teams will likely make use of all of these navigation approaches to some extent.

Q) Our program works great, but it requires the handler to position the robot correctly within about half a millimeter of the right way. Otherwise, the rest of the round is hopeless. Any suggestions, please?

A1) Find a repeatable way to position the robot by alignment with landmarks on the mat, or through use of a jig or similar aiming device. This approach tends to be pretty straightforward to implement, but requires extra time for the handlers, and is still error-prone.

A2) Change the programming so that the handler doesn't need to position the robot 100% correctly. This can be pretty difficult.

A3) Always start the robot from the same place in base. Perhaps right up against the wood borders in the corner of the base.

A4) Look at your robot design and see if it can be made less sensitive to exact positioning. What is the failure mode and can a mechanical rather than programming solution fix it?

A5) Have your kids think of other things that are aimed like compasses, telescopes etc. How are these things aimed? Is there any part that is used for aiming? Maybe they could add that to the robot to increase the accuracy.

Q) What type of driving base is best to solve all the missions?

A) It may well be the case that one base may not be able to solve all the missions. Many teams will make modifications to the robot as the match progresses to do various things. I saw a robot once at the end of a match that looked NOTHING like how it started off.

Table

[FAQ Table of Contents](#)

Q) My shop light fixture has a plastic diffusion lens. Should I use it?

A) No. The specification for an FLL table uses a shop light without a lens.

Q) How can I be sure the lighting on my practice table will be the

same as at competition?

A) You can't! You must allow for variations in your mechanical or software design.

Q) Using construction 2x4 for the table borders seems 'rough'. Would it be better to make the borders from a better wood?

A) Yes and No. A beautiful border of ebony would be very pretty. But at competition you will be using tables of rough and crooked lumber. Best to practice and learn to deal with those problems early! I would take a little sandpaper to the edges just to keep the splinters down.

Q) Now that I have made all the LEGO field pieces how do they stay in place?

A) You should have received a sheet of 3M Dual Lock self adhesive material. This material is similar to Velcro. Pieces of this material are applied to the field pieces as well as to the mat to hold certain elements in place. Detailed instructions will be provided by FIRST at kickoff. Not all elements are held in place. Some may be loose to allow the robot to manipulate them.

Q) What color should the table border be painted?

A) The border should be painted black. FLL does not specify the type or manufacturer of paint nor the texture (gloss, matte, ...). This choice is up to you. Consider carefully relying on an exact color as the tournament tables may differ. In fact, some local tournament tables may not be painted at all!

Q) How can I make a duplicate of the mat for off site practice?

A) This is a tricky problem. It will be nearly impossible to make a true duplicate. The texture of the mat is bound to be different on the copy. For some robots this may not be an issue. Most robots though, will behave differently since the friction will be different. Some teams paint the features on plywood, some transfer to paper. Some teams cover the paper copy with clear adhesive shelf paper. Others find that Tyvek and magic marker works well.

Q) I have heard that the mat is fragile. Could you explain?

A) The mat is made from some kind of vinyl with a clear plastic coat over the top. The bond between these layers is what is fragile. A sharp bend or kink in the mat will cause the clear plastic to delaminate and bubble. A dropped robot can also dent the mat and start a bubble. Bubbles in the mat change the surface color and may render the mat useless for line following. I have found that gentle heat and firm pressure can temporarily repair a bubble.

Q) How am I supposed to attach the mat to the table?

A) You probably do not need to attach the mat at all. After a few days, the mat will flatten out and stay in place. If your mat needs a little convincing to stay flat on the ends, try double sided carpet tape. If you need to store the table vertically, remove the mat each time you use it, or use a staple gun to tack it down along the edges and in out of the way places.

Q) Any suggestions for a detachable light assembly?

A1) Use wing bolts. A2) Use the preformed metal table assembly parts that fit 2x4s. They are sold at Home Depot and Lowes and are less than a dollar. They mount to the table and provide a place to slide the 2x4 in.

Q) Our mat does not fit in our table!

A) Make sure your table dimensions are correct. The inside dimensions should be 45 +/-1/8 inch in width and 93 +/-1/8 inch in length. Then trim the mat to fit. FIRST normally provides advice in trimming oversize mats. If

not, the guiding principle is to leave the edges adjoining the base alone and trim the opposite edges.

Q) Will the tournament tables be exactly like our table?

A) Tables are likely to vary considerably based on the skill of the builder. As such, tournament tables vary in quality. I have seen tacky paint, massive unfilled knot holes, light supports in the wrong locations, warped surface and borders, and out of spec dimensions. Come to a tournament prepared to make last minute corrections. For veteran teams, focus on ways not to rely on exact placements. Missions should work even with placement errors of up to 1 cm.

History

[FAQ Table of Contents](#)

I compiled the original UFAQ in the spring of 2003. FIRST was kind enough to provide me with a text dump of all 6000 plus messages from the 2001 and 2002 seasons. After compilation, I sent out the UFAQ to the top forum contributors for review.

The bulk of the comments I received were spelling/wording/ type problems with just a few technical comments. I also received from a few individuals the concern that I was giving away too much information. That teams should learn the answers to these questions on their own.

These comments sent me into a period of philosophical upheaval. How far is too far when you teach a team a new skill? Was I providing too much information in the UFAQ? Was I providing too much information to my own team? Was I violating the FLL code of ethics? I went back and read all the information from FIRST that I could find to try to answer these questions. I found little to give me guidance.

So I arranged a meeting with the kind folks at FIRST and asked them for guidance. Just how much information is too much information? Had I gone too far? They had no simple answer as they themselves are grappling with the issue. Their current rule of thumb is not to provide information on a silver platter but make it available with just a bit of digging. They reviewed the UFAQ and gave me suggestions.

So I went back and edited the UFAQ with this in mind. I either deleted answers, minimized the answers, or turned the answers into thought provoking questions. Some will think I have not provided enough information, others will think that I have gone too far. I am open to your [point of view](#).

In the fall of 2004 I update the UFAQ to reflect the new rules. Luckily, the rule changes were minor, so the UFAQ survived mostly intact. Numerous updates were made during the season to account for various rule interpretations.

In the fall of 2005 I update the UFAQ to reflect the new rules. Again, the rule changes were minor.

Change History		
Version	Comments	Date
1.0	Original "over the top"	5/03
2.0	Initial web release	9/03
2004-a	Updated for 2004 rules	9/14/04

2004-b	Added RoboLab versions, new USB IR tower driver	9/16/04
2004-c	Fixed section on batteries. Deleted incorrect section on deliverable and retrieval objects.	9/16/04
2004-d	Added a description of tag-team and added Mindscripts as a legal language.	9/17/04
2004-e	Added the printer friendly all in one page courtesy of Steve Dakin.	9/21/04
2004-f	Added a section on Mindscripts courtesy of Jon Gilchrist.	10/1/04
2004-g	Corrected the link to the program that prints RIS programs.	10/7/04
2004-h	Removed references to documents at www.HightechKids.org. All links are broken.	10/7/04
2004-i	Added the "How to ask a Question" page.	1/14/05
2005-a	Initial 2005 version	8/17/05

Counter History		
Page	9/04	9/05
index.htm	4202	4200
faq_robolab.htm	1461	3684
faq_prg.htm	1263	1763
faq_parts.htm	999	881
faq_robot.htm	858	839
faq_strategy.htm	759	514
faq_rules.htm	706	426
faq_competition.htm	623	414
faq_ris.htm	617	721
faq_team.htm	583	479
faq_general.htm	581	384
faq_table.htm	547	366
faq_research.htm	538	413
faq_sensors.htm	517	429
faq_2003.htm	484	0
faq_2004.htm	0	299
faq_batteries.htm	462	476
faq_history.htm	177	120
faq_printer.htm	0	1659
faq_mind.htm	0	278
faq_question.htm	0	203

[UFAQ Table of Contents](#)

[Home](#)

Disclaimer: This FAQ is not an official FIRST document. It is an accumulation of knowledge derived from six thousand messages posted to the FLL forum over three seasons. It has been reviewed by numerous people, but may still contain errors. Use at your own risk.

Readers are encouraged to submit errors, suggested wording changes, new topics, or comments to Skye

Sweeney at skye@fll-freak.com

Copyright 2005 Skye Sweeney; Last Updated on 7/20/2005; Counter:

0