Scripting an iRobot Create

By Yuta Akiya, Evan Friedman, Joshua Feldman, Ethan Duong, Spencer Mayton and Grant Swajian

Scripting an iRobot Create can be difficult, but with the right functions, it is capable of being a defensive nightmare in the perspective of our opponents. Not only can the scripted Create act as a bodyguard for other robots, but is also able to other tasks with simple maneuvering or basic constructions. In this year's game, some tasks the scripted Create can accomplish are collecting samples, botguy, or the red cube as well as knocking rocket boosters off of the skycrane.

Botball has always supplied us with what we know as an iRobot Create and two controllers, this year the KIPR Link, which we use as the brains of our robots. This, understandably, has seemed to limit every Botball competitor to two robots. This year is no different than any other time. However, there is a way to, if you will, cheat the system. Though we are only supplied two controllers, the iRobot Create has the capability to function as its own controller, without reliance on CBC or a Link.

Now, in order for a team to take advantage of this, the team must be willing to step away from our conventional "C" programming on KISS IDE. The programming used to operate the IRobot Create is known as "scripting". However, this scripting can not be done through what KISS has supplied us. The program used to script a Create is called RealTerm. This program is usually used for quote-on-quote, "capturing, controlling, and debugging binary and other difficult data streams." (*Terminal Software*) However, if one reads the manual, he/she would find that if he/she hits the tab on the window that says "send", he/she can input numbers for use of a

Create. Each set of numbers are interpreted by the Create as functions and are carried out once one executes a starting action, such as a bump on the touch sensor of the Create.

The *iRobot Create Open Interface* manual describes all of the functions one may program the Create to execute, and the format with which it must be written into the script. This manual supplies a number that each function is started with. From there, the programmer must fill in variables, such as speed, radius, and others. However, that is where the scripting becomes more complicated. For every value the programmer wants to use, he must first convert it into Hex, and from Hex, he must convert every pair of characters into decimal.

One example that the Open Interface manual gives is with the command for the robot to drive forward. We are told that we must first enter in the value 137, and from there, a value for speed must be entered, and after that, a value for radius. Each value one wants to use must be converted from decimal to Hex. There is no problem with this if the value is positive; however, many decimal/binary/Hex converters do not convert negative values. To convert a negative number, one must first convert the decimal (disregarding the negative, of course) into 16 bit binary (to make a sequence of binary more bits, one can simply add zeros to the beginning). From this point, each digit must be inverted, meaning all of the zeros must be changed to ones, and every one must be changed to a zero. Once this is done, the binary converted into Hex. Once the value is in Hex, every two characters in the Hex sequence must be converted into decimal independently. For example, if one wants to use the value of -200 m/s for the speed of the Create (as the Open-Interface Manual supplies), he/she must convert 200 into binary, which is 11001000. This is an 8 bit sequence, so 8 zeros must be added to the beginning of the sequence. From 0000000011001000, the zeros must be changed into ones and the ones to zeros (to make the value negative). Therefore, the final binary sequence is 1111111100110111. Once this is

converted into Hex, one gets the value FF38. The FF converted from Hex to decimal is 255, and 38 converted to decimal is 56. Therefore, to make the Create drive at -200 m/s, the sequence is 137 255 56.

The numbers we use as the beginning statements are 128, 132, 152, 00, 158, 5, 251. There are two modes that the Create can operate on, 132 being full mode and 131 being safe mode. The 158 in the sequence is interpreted as “wait event” by the Create, which tells the Create to wait until a specific event occurs. In our statement, 5, 251 follows 158, and therefore is the event waited for. The bump sensor on the Create is triggered by 5, 251, so the Create waits until the bump sensor receives a signal and then starts the program. There are more commands that can be used for “wait event”. For example, 17, 239 is the play button on the Create.

In the game of Botball, which is Palm Desert Charter Middle School’s main robotics focus, turning or spinning is obviously very useful. As Anakin Skywalker said in *Stars Wars I: The Phantom Menace*, “...spinning, that’s a good trick!” In order to spin, the programmer first has to type in 137 (which you may recognize from the “drive” command) then the speed, which is limited to 1, 44. After that, 0, 1 must be entered, which the Create to spin counter-clockwise. The next sequences of code tell the create the wait angle. Entering 157 tells the Create that it must spin until it has reached a certain angle, and this must be followed with 0. All that you now have to do is type in the desired angle. An example of this is 137 1 44 0 1 157 0 90, with 90 degrees being the desired angle measurement.

these

To end every program the programmer has to type in two spaces, then 137, then 4 zeros. After, type 153 twice and send the script by clicking “send numbers”. Your create should turn off as you send the command (If it doesn't turn off, either the create was not connected properly to the

computer or the command had an error in it). To run a script, one must perform whatever event the Create was told to wait for, and there you have it!

Palm Desert Charter Middle School, the school we represent, has four teams. All four of our teams competed in the Greater San Diego regional competition. One of our teams did well enough even to get first on Double Elimination. This team is a testimony to the importance of the scripted Create. Although they hadn't used a scripted Create, they had use a Create with a Link, but with nothing built on it, in the final round. This Create was sent to the Sky Crane to knock down the rocket boosters, thus putting it in a position to block the other team. The other team, therefore, was not able to score any points, and those ten points won us the Double Elimination. Through blocking and doing simple tasks, such as knocking into objects or collecting objects, a match may be won with only one simple scripted Create.

In conclusion, scripting a Create is very useful because basically, you can get an extra robot. There are some disadvantages of having a scripted Create, namely, your Create might be desirable to use with other robots. However, overall, there are far more advantages to using a scripted Create than disadvantages. Any team would be better off knowing how to script a Create, because more often than not, there will be many times where one is needed.

Works Cited

"Create Manuals." *Create Manuals*. N.p., n.d. Web. 17 June 2013.

"IRobot - Robots That Make a Difference." *IRobot: Cleaning Robots*. N.p., n.d. Web. 17 June 2013.

"IRobot Create." *Wikipedia*. Wikimedia Foundation, 06 Mar. 2013. Web. 17 June 2013.

"Serial Terminal:." *Terminal Software*. N.p., n.d. Web. 17 June 2013.

Star Wars the Phantom Menace George Lucas