

Controlling RC Vehicles with Your Computer Using LabVIEW and the USB-6008 Interface

Operates Many—But Not All—4-Function, 27-MHz RC Vehicles

NOTE: The instructions here are a rough draft, designed to aid those who want to experiment with using National Instruments' LabVIEW 7.1 software and the USB-6008 or USB 6009 interface to operate low-cost RC Vehicles. More complete programs and instructions will be available in early 2006.

Contact fred.thomas@mathmachines.net with questions or suggestions.

Preparing Your Computer

You must have LabVIEW 7.1 and the USB-6008 or 6009 interface installed on your computer prior to using the Electronic RC Controller. Students and faculty may purchase the student version of both at a discounted price. Additional information is available on National Instruments' website at

<http://sine.ni.com/nips/cds/view/p/lang/en/nid/14681>.

Follow the instructions provided by National Instruments to install LabVIEW, the USB interface and the NI-DAQmx drivers before attempting to use the Electronic RC Controller.*



Attaching the Electronic RC Controller

The Math Machines which inserts directly into the digital side (pins 17-32) of the USB 6008 or 6009 interface. The RC Controller can then be attached to any of the 3 modular jacks on the adapter using a standard 6-conductor phone extension.

Alternatively, you can use a standard 6-conductor phone line and wire it directly to the screw terminal provided with the USB 6008 or 6009. If the phone plug is attached to the phone wire with "tab up, white on the right," the wires attach to the USB 600x as shown below.

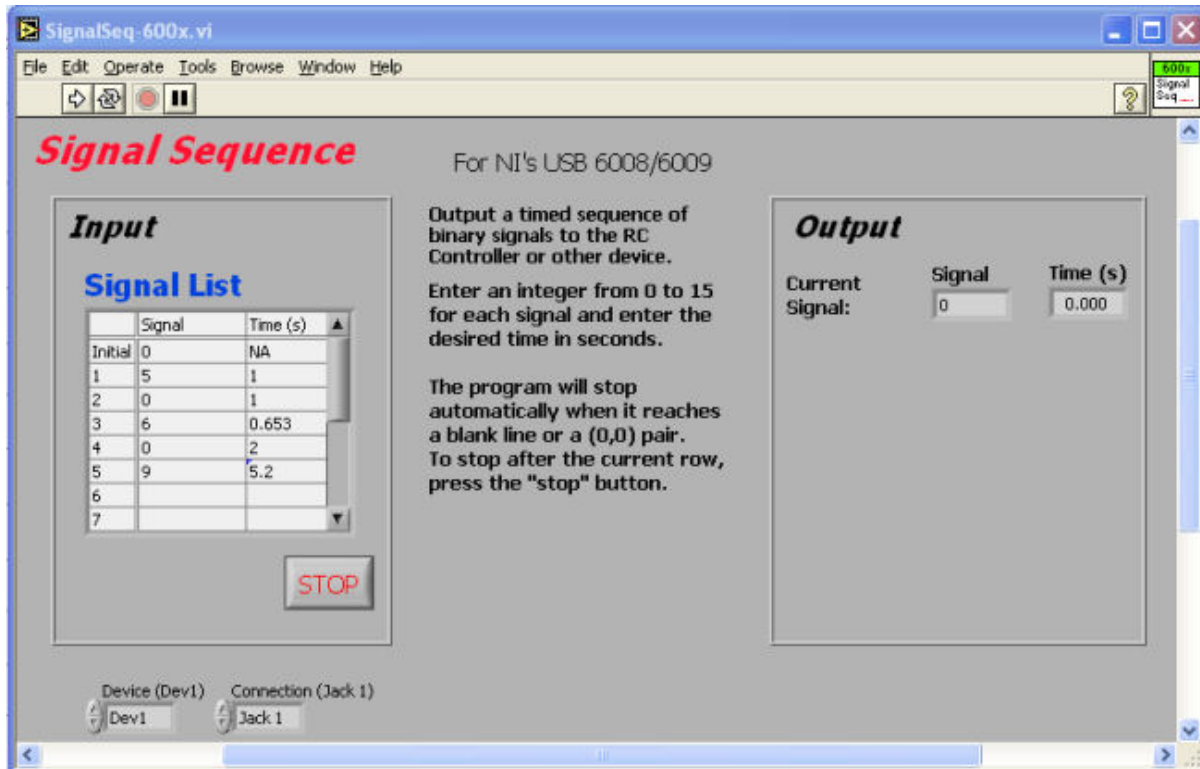
SIGNAL SEQUENCE VI

The SignalSeq-600x VI (virtual instrument) is intended to serve as an example for writing instructions to the Electronic RC Controller. The Front Panel shown on the below is the screen that lets

| Wire | Function | USB-600x |
|--------|----------|----------|
| Black | Ground | 32 GND |
| Blue | Data 0 | 17 P0.0 |
| Yellow | Data 1 | 18 P0.1 |
| Green | Data 2 | 19 P0.2 |
| Red | Not used | NC |
| White | Data 3 | 12 P0.3 |

* Some USB-6008 or 8009 interfaces shipped with the NI-DAQmx base drivers, rather than the full version. If you have the base version, download the full version from National Instruments at <http://sine.ni.com/nips/cds/view/p/lang/en/nid/10181>

you enter a sequence of codes with the time for each code to be executed. Each code must be an integer from 0 to 15 and times will be executed with a precision of 1 ms.



You should verify that the VI is set to use the correct device (this should be "Device 1" unless you have more than NI interface attached to your computer) and that the jack setting matches the adapter jack to which the RC Controller is attached. Enter the sequence you want, make sure the vehicle is powered up, and press LabVIEW's run arrow to start. The sequence above, for example, makes JAKKS Road Champ move forward for 1 second, pause for 1 second, pivot left for 0.653 seconds, pause for 2 seconds, and then pivot right for 5.2 seconds

Simply running the Signal Sequence VI lets you program complex sequences of movement for your vehicles. If you are ready for more advanced tasks, note that the entire program is available for you to examine and modify by selecting "Window" and displaying the "Block diagram."

It is also possible for you to write your own VI using signal sequence as a sub-VI. For example, you might write a VI that monitors an analog light sensor on the USB-6008 and orders your vehicle to execute a particular sequence of moves depending on how the light level changes.

Understanding the Signals

The table below shows the response typical of several 4-function, 27-MHz vehicles, including a vehicle with front-wheel steering (illustrated by New Bright's model 2424) and one with tank-type steering (illustrated by JAKKS's Road Champ RC

vehicle). Some 27-MHz vehicles will respond to all 4 signals, but in different ways—for example, the signal that makes one vehicle go forward may make another vehicle move backwards. Depending on the specific coding used by the manufacturer, some 27-MHz vehicle may respond to only some of the 4 signals and some vehicle may not respond at all. *Have fun experimenting with different vehicles! We have found miniature radio-controlled mice, boats, submarines, flying saucers and many other low-cost “toys” that **do** respond to the RC Controller.*

Note that many movements of the vehicle require 2 simultaneous signals. To make the New Bright truck turn left, for example, you need to make it move forward while simultaneously turning the steering wheels to the left. You achieve this simply by sending the sum of two signals. If “2” makes it move forward and “4” makes the steering wheels turn right, then signal “6” will make both happen simultaneously. To move the Road Champ forward, you need to send “5” (1+4) to make both the left and right wheels move forward.

Some signals are allowed under the rules of arithmetic but don’t really make sense physically. Sending $S=3$, for example, tells New Bright’s model 2424 truck to go forward ($S=2$) and backward ($S=1$) simultaneously. In reality, one of the two signals will probably take priority so the vehicle does respond. It is much easier to understand and control the vehicle’s response, however, if you think in terms of the true binary elements—0, 1, 2, 4 and 8—which are shown in bold. If two different signals produce the same result, always use the lower value.

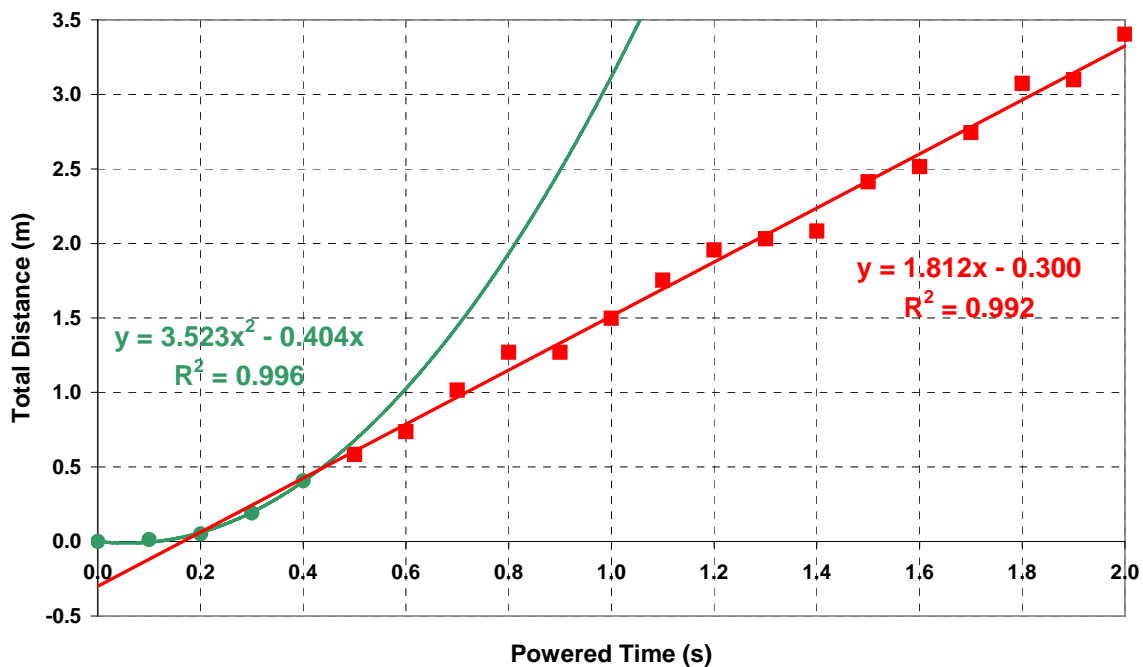
| Integer (base 10) | Binary pattern | $2^3 = 8$ | $2^2 = 4$ | $2^1 = 2$ | $2^0 = 1$ | New Bright Model 4242 | JAKKS Road Champ |
|----------------------|-------------------|-----------|-----------|-----------|-----------|----------------------------------|--------------------------------|
| 0 | 0000 | 0 | 0 | 0 | 0 | STOP | STOP |
| 1 | 0001 | 0 | 0 | 0 | 1 | VEHICLE BACK | LEFT WHEEL FORWARD |
| 2 | 0010 | 0 | 0 | 1 | 0 | VEHICLE FORWARD | LEFT WHEEL BACK |
| 3 | 0011 | 0 | 0 | 1 | 1 | | |
| 4 | 0000 | 0 | 1 | 0 | 0 | STEER RIGHT | RIGHT WHEEL FORWARD |
| 5 | 0001 | 0 | 1 | 0 | 1 | BACK RIGHT | VEHICLE FORWARD |
| 6 | 0010 | 0 | 1 | 1 | 0 | FORWARD RIGHT | PIVOT LEFT |
| 7 | 0011 | 0 | 1 | 1 | 1 | | |
| 8 | 0100 | 1 | 0 | 0 | 0 | STEER LEFT | RIGHT WHEEL BACK |
| 9 | 0000 | 1 | 0 | 0 | 1 | BACK LEFT | PIVOT RIGHT |
| 10 | 0001 | 1 | 0 | 1 | 0 | FORWARD LEFT | VEHICLE BACK |
| 11 | 0010 | 1 | 0 | 1 | 1 | | |
| 12 | 0011 | 1 | 1 | 0 | 0 | | |
| 13 | 0000 | 1 | 1 | 0 | 1 | | |
| 14 | 0001 | 1 | 1 | 1 | 0 | | |
| 15 | 0010 | 1 | 1 | 1 | 1 | | |

Collecting Data for the Vehicle

A very good introductory algebra activity with the RC Controller is to give student teams the task of sending the vehicle a distance, x . Tell them in advance where their vehicle will start, but make it clear they will not know the location of the finish line until a few minutes before the start. To prepare, have the students collect data for the distance traveled by the vehicle when it is powered for a variety of time intervals. If you have several different vehicles that respond to the RC Controller (perhaps brought in by the students themselves), each team can work with its own vehicle. Let the groups take turns to test their results.

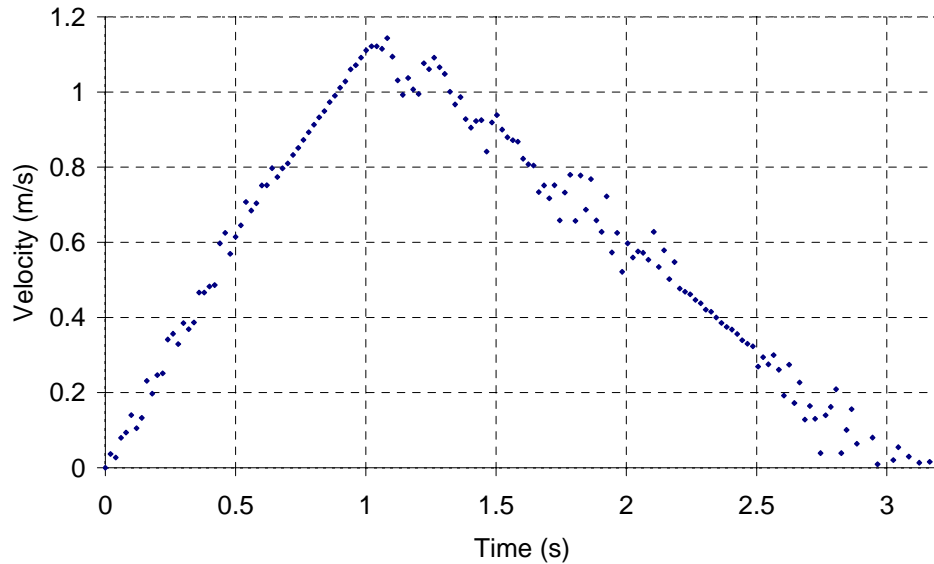
Typical data is shown below. Note that it is generally linear for times greater than about 0.5 seconds, but the y-intercept is not zero. It takes the vehicle a short time to get started. If you want your students to work with linear equations, limit them to times greater than 0.5 seconds. For more advanced students, you may want to include very short times and very short distances.

Total Distance vs. Powered Time



The two graphs below illustrate data collected using Vernier's motion detector to show instantaneous velocity and displacement of a vehicle similar (but not identical) to the one shown above. Note that the previous graph shows total POWERED time—the time during which power is delivered to the vehicle's motor—and total distance traveled. All vehicles have an additional deceleration time during which they continue to move after the power stops.

Velocity vs Time 1.000 s of Power



Displacement vs Time 1.000 s of Power

