- support
- developers

- store
- blog
- about
- community
- apps
- meet romo

Blog

What we're thinking

Post navigation

← Previous Next →

Hackers Welcomed: Here’s Our Second Generation Protocol

Posted on April 16, 2012 by phu

When creating the firmware for our second generation Romos, we wanted to stay true to the open, flexible, hackable nature of Romo. Here’s our second generation protocol, so you can move Romo from any smartphone that can emit sounds through an audio jack.

You might have remembered our first protocol allowing us to move the robot left, right, forward, and back (old protocol link). As it turns out, Romo’s movement with only these four commands are extremely limited. Romo couldn’t do slow wide angle turns or have variable speeds. Furthermore, the auxiliary ports we included on the first gen Romos had only one state, ON.

We went back to the drawing board and completely revamped the firmware. The new protocol is a PWM signal sent over the audio channel that gives the robot 255 speeds on each motor, along with the ability to have each AUX channel go forward, back, and stop.

The PWM signal is comprised two channels, the left used for the clock signal and the right for the data signal. The PWM signal is generated as a square wave of 1000hz.

The left channel clock signal comprises of 12 bits. In order to trigger the circuit, the start the square wave has to be low. The peak of each wave denotes a bit. A full 12 bits denotes one command.

The right channel data signal comprises of first 3 bits for the address, the next 8 bits for the actual command, and the last bit as an even parity bit.

- 011 – All 3 Auxiliary Motors

For the next 8 bits, the commands range from 0 to 255.

- 0 (00000000) is full reverse
- 128 (10000000) is stop
- 255 (11111111) is full forward

Any thing in between those values will give you a partial speed.

The last bit is a parity bit that keep the entire frame even. For example, if you have an address of 001 (left motor) and you want to send it in full reverse (00000000), the last bit has to be 1 to make the entire frame even. So the command for that would be 001000000001.

Each motor is triggered separately. It only seems that they move together because of the short duration of the audio signal sent.

Here are some sample audio signals captured on an oscilloscope:

Both motors stop:

Both motors forward full speed:

Both motors back full speed:

Turn right in place = Left forward full, Right back full

Turn left in place = Left back full, Right forward full

Interested in building out a library for Windows, Palm Pre phones for us? We’d love to talk to you, drop me a line at phu (at) romotive.com.

You can find our latest SDK’s for iOS and Android here:

https://github.com/Romotive/Romo-SDK

Happy hacking!

Share

This entry was posted in Code, Robots by phu. Bookmark the permalink.

One thought on “Hackers Welcomed: Here’s Our Second Generation Protocol”

1. Shannon on May 2, 2012 at 6:19 am said:

I'm happy to read your blog

Reply ↓

Leave a Reply

Your email address will not be published. Required fields are marked *

Name *

Email *

Website

Comment

You may use these HTML tags and attributes: `<a href="" title=""> <abbr title=""> <acronym title=""> <b> <blockquote cite=""> <cite> <code> <del datetime=""> <em> <i> <q cite=""> <strike> <strong>`

Post Comment