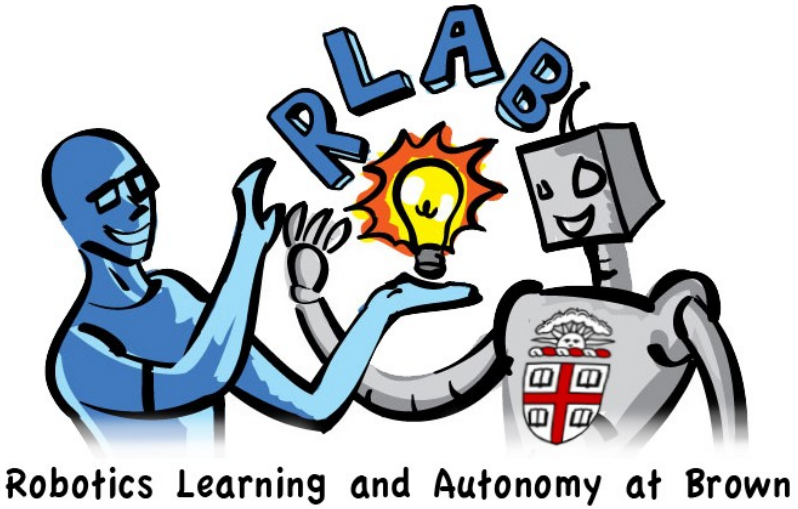
Rosbridge and rosjs

Christopher Crick



Rosbridge

a light weight protocol for using ROS

uses:

- Bash
- JavaScript
- telnet
- serial applications
- uControllers
- etc.

Rosbridge

ROS without

- linking
- implementing TCPROS
- implementing much of anything

Rosbridge

required:

- output
  - stdout
  - serial line

nice to have:

- input
- ASCII strings
- sockets

really nice to have:

- JSON-support
- web-sockets
- a Rosbridge binding
  Like ros.js!

The protocol

\x00{“receiver”:”/topicOrService”,
“msg”:{},
“type”:”package/Type”}\xFF

Web addresses

This presentation:
http://10.102.1.188/rosbridge_and_rosjs.pdf

Rosbridge:
http://10.102.1.188/rosbridge.tar.gz

Javascript editor:
http://10.102.1.188/editor.html

Demo: Rosbridge at the command line

```
$ rosrun rosbridge rosbridge.py
$ mkfifo bridge
$ while true; do cat bridge; done | nc localhost
9090 &
$ echo -n -e 'raw\r\n\r\n' > bridge
$ echo -e '\x00{“receiver”:”/test”,”msg”:
{“data”:”Hello.”}, “type”:”std_msgs/String”}\xff'
> bridge
$ echo -e
'\x00{“receiver”:”/rosbridge/subscribe”, “msg”:
[“/test”,-1,”std_msgs/String”]}\xff' > bridge
```

Pseudoservices

challenge, authenticate, topics, publish, services, get_param, set_param, has_param, delete_param, search_param, get_param_names, unsubscribe, subscribe, log, challenge, authenticate, typeStringFromTopic, typeStringFromService, msgClassFromTypeString, reqClassFromTypeString, rspClassFromTypeString, classFromTopic, classesFromService

Params

/brown/rosbridge/passfile
/brown/rosbridge/host
/brown/rosbridge/port
/brown/rosbridge/hz

ros.js

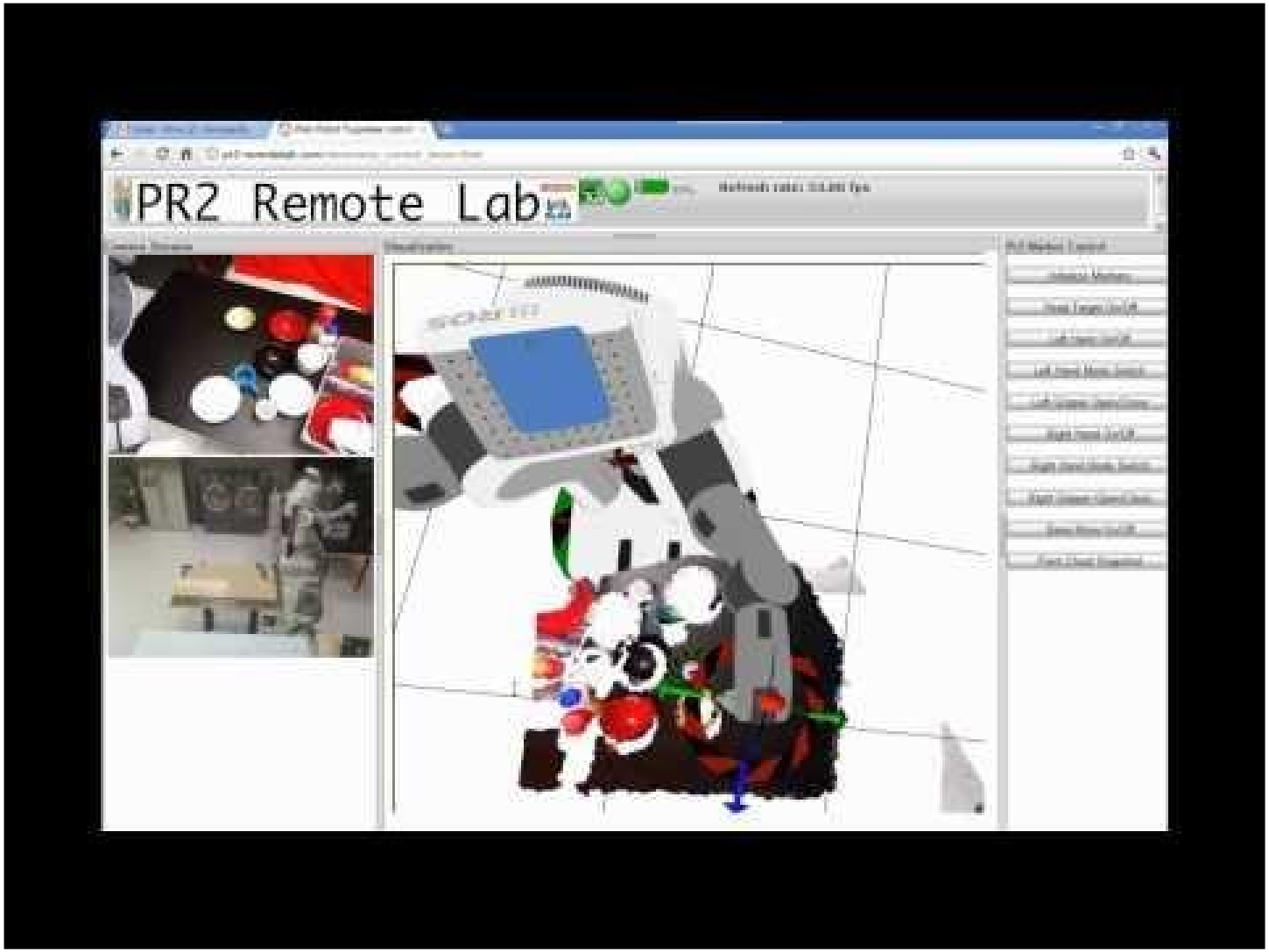
a JavaScript binding to rosbridge

ros.js hands on

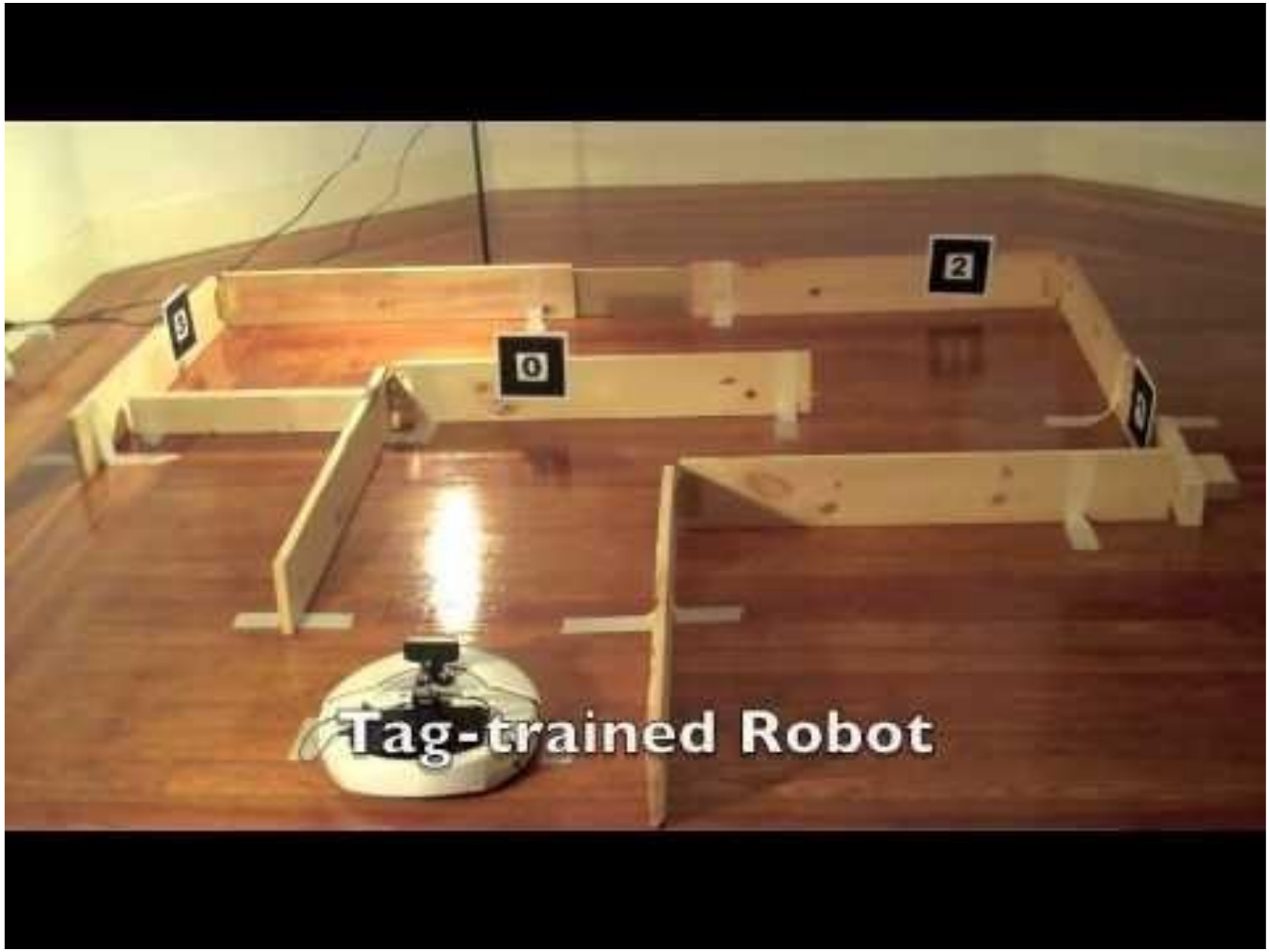
ros.js

- is a lightweight Javascript binding for ROS
- depends on rosbridge
- allows you to write pure HTML5 robotic apps
- uses websockets and JSON under the covers

What can you do with ros.js?



What can you do with ros.js?



What can *you* do with ros.js?

- zero install teleop
- iPhone and Android motion and touch UI's
- rapid prototyping
- utilize web-services from ROS
- control
- write code once
  - use in browser
  - use on robot (server side Javascript)

Basics

- connecting

Basics

- connecting
  - set correct URL
  - register callbacks for websocket events

Basics

- using the debugger

```
var tick = true;
setInterval(function() {
  if (tick) {
    console.log('tick'); //or document.title = 'tick';
  } else {
    console.log('tock'); //or document.title = 'tock';
  }
  tick = !tick;
},1000);
```

Basics

- calling a service

```
con.callService('/rosbridge/topics',[],function(topics) {
  document.write(topics);
});

//this is one of the rosbridge pseudo-services
```

Basics

- publishing

HTML: `<br/><a href="#" id="pub">pub</a>`

```
document.getElementById('pub').addEventListener('click',function() {
  con.publish('/msg','std_msgs/String',{data:"Hello, World."});
});
```

Types

ROS std_msgs/String:

string data

ros.js equivalent:

{data: "Hello"}

Types

ROS turtlesim/Velocity:

float32 linear
float32 angular

ros.js:

{linear: 2.0, angular: 0.0}

Simple teleop (1/2 slides)

```
var myTurtleTopic = '/turtle13/command_velocity';


var codes = ['87','83','65','68'];
var actionMap = {'87':{'linear':2.0,'angular':0.0},//w is up
                 '83':{'linear':-2.0,'angular':0.0},//s is down
                 '65':{'linear':0.0,'angular':2.0},//a is left
                 '68':{'linear':0.0,'angular':-2.0},//d is right
                };
var activeMap = {'87':false,'83':false,'65':false,'68':false};


document.onkeydown = function(evt) {
  activeMap[evt.keyCode] = true;
};
document.onkeyup = function(evt) {
  activeMap[evt.keyCode] = false;
};
```

Simple teleop (2/2 slides)

```
setInterval(function() {
  var linear = 0;
  var angular = 0;
  for (var i = 0; i < 4; i++) {
  var code = codes[i];
    if (activeMap[code]) {
      linear += actionMap[code]['linear'];
      angular += actionMap[code]['angular'];
    }
  }
  con.publish(myTurtleTopic,'turtlesim/Velocity',
              {linear:linear,angular:angular});
              },100);
```

Turtlebot teleop

ros.js geometry_msgs/Twist:

{

linear:{x:0.0,y:0.0,z:0.0},

angular:{x:0.0,y:0.0,z:0.0},

}

topic is: /cmd_vel

Can you adapt the previous code?

Questions?