

Hello,

Over the past decade, I've presented a bunch of talks with slides about various aspects of ESP software and firmware.

I've made these available in the past on the MBARI FTP server, but it purges files after 30-days.

So...

I just posted them all, permanently, on the ESPshore server.

<http://espshore.mbari.org/doc/talks/>

Some of these slides are outdated, but the bulk of the material is still valid.

Here's an index for reference:

<a href="#">ESPaxes.pdf</a>	51403
<a href="#">ESPcomms.pdf</a>	176235
<a href="#">ESPcoreLib.pdf</a>	87771
<a href="#">ESPenvironment.pdf</a>	99274
<a href="#">ESPerrors.pdf</a>	123146
<a href="#">ESPi2c.pdf</a>	79951
<a href="#">ESPohsul.pdf</a>	152573
<a href="#">ESPparams.pdf</a>	109696
<a href="#">ESPscripting.pdf</a>	81286
<a href="#">ESPensors.pdf</a>	108602
<a href="#">ESPservos.pdf</a>	1175053
<a href="#">ESPshore.pdf</a>	174828
<a href="#">ESPsim.pdf</a>	109832
<a href="#">ESPslide.pdf</a>	107038
<a href="#">ESPtelecom.pdf</a>	472490
<a href="#">ESPtriggers.pdf</a>	111504
<a href="#">usersAndGroups.pdf</a>	40582

These are ordered as I'd recommend you read them:

ESPscripting:

A gentle introduction to ESP scripting with references to learning the Ruby programming language.

ESPparams:

ESP protocol scripts have sensible defaults for take a lot of parameters you can tweak.

ESPenvironment:

Environment variables and file directory layout.

Important for figuring out where to look for ESP's data products and where the ESP looks for its Ruby scripts.

ESPerrors:

Understanding ESP errors and error messages

ESPsim:

How to run simulations of ESP scripts to verify them before deployment  
(catch errors in scripts before you waste time and reagents)

EStelecom:

ESP communications via telecom networks  
(more up to date than ESPcomms and ESPshore)

ESPslide:

Controlling ESP actuators (see ESPservos also)

ESPaxes:

A quick tour of the ESP's internal software objects.  
Error messages make a lot more sense after you've read this.

ESPservos:

Understanding the details of controlling the ESP's many DC servo motors.  
(trajectories, current limits, etc.)

ESPsensors:

Use of the ESP's CTD, can and ISUS contextual sensors

ESPtrigger:

How to set up trigger conditions that the ESP can monitor to start processing when the environment is likely to contain organisms of interest.

ESPcomms:

An overview of the communications methods the ESP has used.  
This does not yet cover the new PPTP Virtual Private Networks and cell comms,  
But it does discuss serial and ethernet.

ESPcoreLib:

Basic logging, time parsing and error recovery techniques.

ESPi2c:

Very technical information about the ESP's use of the I2C electrical data bus.

ESPohsu1:

An overview of ESP mission scripts and use of environmental conditions to trigger activities.

ESPshore:

Communications while deployed (largely outdated)

ESPuser:

Very generic UNIX / Linux information. (users, accounts, etc.)

---

--

Brent Roman MBARI  
Software Engineer Tel: (831) 775-1808  
7700 Sandholdt Road Moss Landing, CA 95039  
mailto:brent@mbari.org http://www.mbari.org/~brent