**DEVELOPMENT JOURNAL**

This file will eventually act as a daily journal of sorts detailing what was done during a work session. It will include any relevant information that cannot be directly assumed through common sense. Details of implementation, design, and problems encountered will be logged as well. Currently the file is rather sparse due to disorganization up this point. Future updates will be logged into this file as necessary and will hopefully allow for copy-paste when it comes time to write the GRP report.

February 25 – February 27

**Problems with EasyVeep and a Custom EasyVeep Implementation**

The purpose of this project is to allow physical real time targets to interact with virtual process simulations. EasyVeep was selected primarily based on easy of acquisition, but it was found that interacting with EasyVeep was more challenging than initially expected. One of the main issues with EasyVeep was its use of ASCII based communication instead of binary communication. This increased serial utilization, and given the control words and baud rates used by EasyVeep it is impossible to achieve latency times of less than 10ms turn around, and even then, that refresh rate could only be maintained for a very short burst.

It was discovered that each of the EasyVeep process simulations were completely encapsulated in their own, independent swf files. A quick Google searched revealed that the Adobe Flash ActiveX control could be easily added to a Visual Studio project. Seeing that minimal effort was needed to run the swf files in a custom Windows application it was decided that to easy debugging and protocol implementation a custom version of EasyVeep should be created. This allowed for full control over serial parameters and protocol.

The swf files were still a black box with no fields publicly exposed in plain text. To allow interfacing with the process simulations, a trial version of the Sothink SWF Decompiler was downloaded and used to analyze the Actionscript code. This allowed for important variable names to be extracted so that they could be read and modified by the custom EasyVeep implementation. Some important variable names are as follows:

|  |  |
| --- | --- |
| **ActionScript Variable Name** | **Purpose** |
| EprgName | English Program Name |
| EprgLeirasX | English Program Description ( X = 0..10) |
| EDigSensX | English Sensor Description ( X = 1..16) |
| EDigActX | English Actuator Description (X = 1..16) |
| DAX | Digital Actuator Value ( X =1..16) |
| DSX | Digital Sensor Value (X=1..16) |
| ASX | Analog Sensor Value ( X=1..16) |

These variables can be accessed in the using the GetVariable and SetVariable functions of the Flash ActiveX Control. For example, finding all the sensor s associated with a process model can be performed as followed after the model is loaded.

do{

SensorDescription = axFlash.GetVariable(String.Format("EDigSens{0}",

SensorDescriptionIndex));

if ( SensorDescription != "" )

movieInfo.Sensors[SensorDescriptionIndex-1] = new DigitalSensor(

SensorDescription, SensorDescriptionIndex);

SensorDescriptionIndex++;

} while (SensorDescription != "" && SensorDescriptionIndex <= 16);

There does not appear to be any event the fires as the sensor values change, so as a result, the sensor values have to be polled constantly to determine when a change has occurred. More development to follow… Primary concern is falling over to a digital input/output when the simulation supports analog values. This will require a bit more testing to figure out the best way.

March 4, 2013

Primary goals:

* Move to github
* Clean up source codes/refactor
* Continue implementation on MyEasyVeep

A repository was created on github in order to replace the confusing, overly verbose SVN repository that was being used. Github was selected due to its standing in the open source community and its excellent user interface that makes documenting and browsing code much easier than the Assembla interface. The code is currently hosted in a publicly visible repository that is available for read only access at <https://github.com/benjaminob6/GRP>.

The migration process is rather slow going as cleanup is being performed and README.md are being added where necessary. One of the benefits of the Github interface is the Markdown Render that automatically displays any README.md files that may be in the viewing directory. This allows for the inclusion of nicely formatted descriptions, hyperlinks, and even pictures; things which were not possible with the SVN repository. As of the writing of this note (11:22AM) only the MyEasyVeep application source has been uploaded, but the READMEs have improved navigation and documentation considerably with minimal effort.