Pier Portal Project II

by
Andy Lam
Brian Markwart

Computer Engineering Department
Electrical Engineering Department
California Polytechnic State University
San Luis Obispo
June 2013

Table of Contents

Acknowledgements

We would like to acknowledge and thank Dr. Bridget Benson for her support in advising us throughout this project. We also want to thank Dr. John Ridgely, Thomas Moylan, Jason Felton, and the Cal Poly Center for Coastal Marine Sciences for their support. We'd like to thank the previous Pier Portal team for all their help answering all our questions. Finally we'd like to thank our sponsors at Yaskawa, Cetacean Research Technology, Seafloor Systems, and Remote Ocean Systems for their donations to this project.

Abstract

The goal of this project is to design and build a remotely controllable camera system that will be deployed at the Cal Poly Pier located in Avila Beach. The system is composed of a camera and all of its lighting and movement controlling hardware enclosed into a waterproof acrylic tube, allowing it to be lowered underwater. The camera will stream its video feed onto a website where any user will be able to watch a live video stream from the pod. In addition, users with the correct privileges will be able to remotely control of the camera system via the website. Remote users will be able to fully control the movement and lighting of the camera system in order to monitor the local sea life at the pier.

Introduction

Figure 1: The Cal Poly Pier at Avila Beach

The waters around the Cal Poly Pier at Avila Beach plays host to a wide variety of sea life. In particular, the pilings supporting the pier often find themselves covered in undersea life like, crustaceans, anemones, and more. The Pier Portal was created in order to give educators, researchers, and the general public an easy method of observing the marine life around the pier. The Pier Portal would allow anyone to view a live streaming video of the sea life around the Cal Poly Pier through the internet browser of their choice. In addition, users with the appropriate permissions would be able to remotely control the camera and freely examine the area around the pier as much as they want.

Background

The 2012 Pier Portal Team

The Pier Portal project was started in 2012 by the first Pier Portal Team. This interdisciplinary team laid out the groundwork of the project. The mechanical engineering side of the team designed the camera pod and the track that the pod would run on and the computer engineering side laid out the basic framework of the website as well as a means of communicating to the website with a Python script.

Existing Solutions

Underwater Camera Systems

One system that functions similarly to the Pier Portal camera system would be the DiveCam vertical camera tracking system invented by Garrett Brown to record Olympic divers [1]. The system is composed of a camera that is guided along a track located alongside the dive platform. The camera is held up by a cable which is released when the divers start their dive which allows the camera to follow alongside the divers as they make their descent. The camera is also enclosed in an aluminum pipe with a clear window along its length to allow the camera to see through it.

Figure 2: DiveCam track located to the right of the dive platform [2]

Requirements

The system should stream a live video feed from the camera to the pier portal website. The video stream should be viewable from any web browser including, but not limited to, Google Chrome, Mozilla Firefox, and Microsoft Internet Explorer. Website users with the appropriate permissions will be able to log in to the website and control the camera as well as the pod functions, with only one user maintaining control over the system at a time. Users with administrator permissions should be able to log in and take control away from standard users.

Specifications

Website

The website will display a video player, rendered with in Flash, which shows a live stream of the video feed from the camera. The website will also display buttons which will allow for control over the tilt, zoom, and focus functions of the camera (the entire pod will rotate so we do not need the camera do pan independently of it) as well as buttons to control all of the pod functions shown in Table 1.

FUNCTION	COMMENT
Rotate Far Right	180° clockwise rotation
Rotate Far Left	180° counterclockwise rotation
Rotate Right	Slight rotation clockwise
Rotate Left	Slight rotation counterclockwise
Rotate to Zero Position	Reset rotation to zero position
Toggle Light Control Mode	Toggle between automatic and manual brightness control for lights
Brightness Decrease	Decrease light brightness (in manual light control)
Brightness Increase	Increase light brightness (in manual light control)
Rotate Lights Up	Rotate lights upwards
Rotate Lights Down	Rotate lights downwards
Reset Light Position	Reset light rotation to center

Table 1: List of pod commands

Pier Server

The Pier Server will be a computer located at the pier alongside the actual pod itself. The server will be relaying commands sent from the website to the pod. The server will also be running the crtmpserver (C++ RTMP Server) which is a high performance streaming server that converts the camera's ActiveX video object into a RTMP (Real Time Messaging Protocol) stream. The RTMP stream is delivered to an Adobe Flash application called JW Player that is embedded on the website. RTMP only works in Flash and not in HTML5. Unfortunately Flash is no longer supported on mobile devices (Android, iOS, or Windows Phone) but Flash is a better solution than ActiveX which is only playable on Internet Explorer (Windows PC only).

Camera

The camera is a 2.0 megapixel IP dome camera with 10x optical zoom and is capable of 30fps at 1080P (1920x1080) resolution and providing up 8Mbps of data. It is powered over Ethernet (POE) and requires 12V and about 1.3 A (the amps range from 0.7 A to about 1.3 A depending on what the camera is doing). The network interface for the camera is Ethernet 10/100 Base-T (RJ45) and can also accept commands and be controlled through RS485 port (using Pelco-D/P protocol). There is also composite video output for potential use with a DVR. The camera is embedded with ActiveX controls (for both the video stream object and the pan, tilt, zoom, and focus functions) for IP control. ActiveX is a software framework created Microsoft and in the context of a webpage it requires Microsoft Windows and Internet Explorer. As this extremely restricts the usefulness of a website, the ActiveX controls for the camera video stream and controls must be converted to or completely bypassed for something that allows for many users on different platforms, operating systems, and web browsers to access the Pier Portal website.

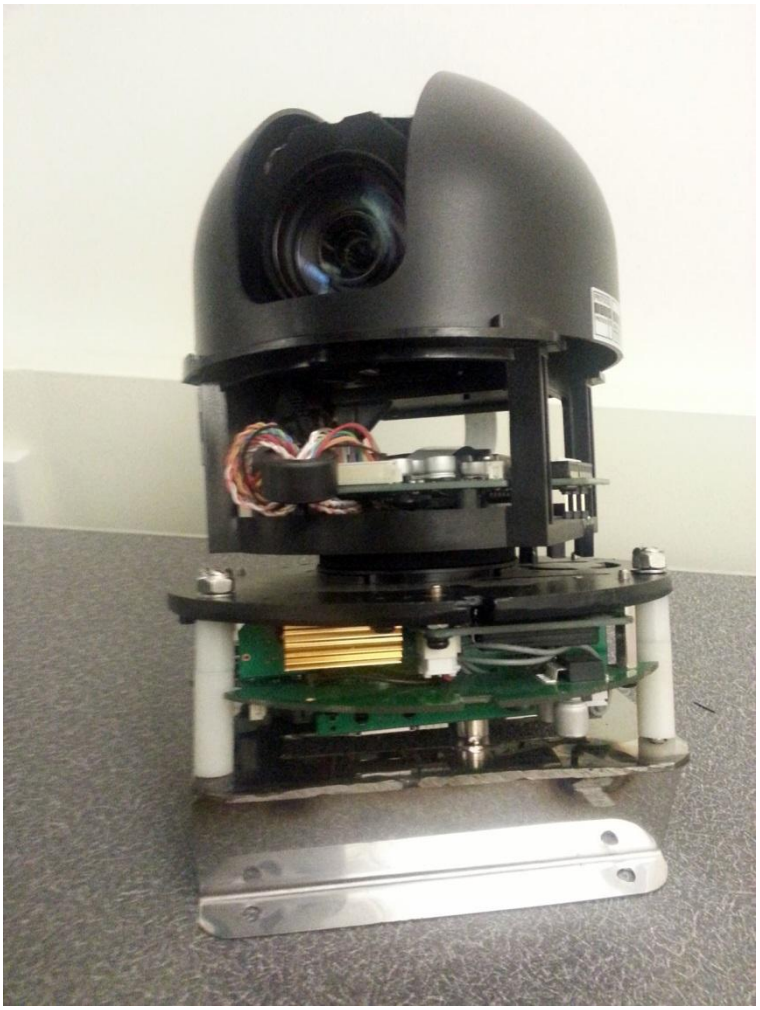

Figure 3: Pier Portal Camera

Design

Software

Pier Portal Website

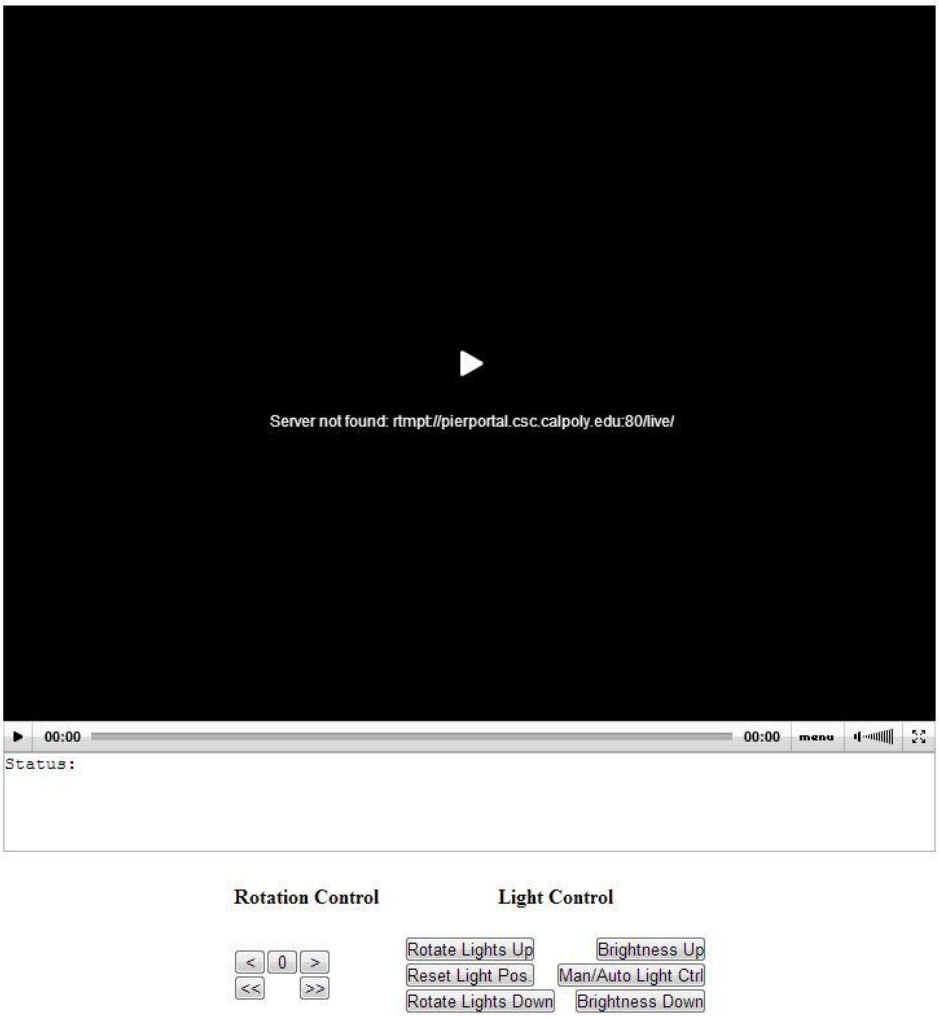

Figure 4: Pier Portal Website Layout

The pier portal website is the frontend of our system which users will use to view and control the camera and pod. The website uses JW Player to display a video stream from the CRTMP server in order to get around the ActiveX requirement for the camera and allow users using any browser to view the video, rather than just Internet Explorer. Below the player are buttons to send commands to the pod. When a button is clicked it calls a JavaScript function which pulls up a PHP script that opens a socket to the pier server and sends a command.

Pier Server Python Script

The pier server runs a Python script In order to relay commands from the website to the pod. When the script is started it will allow the user to choose between commanding the pod directly and allowing the script to receive commands from the website.

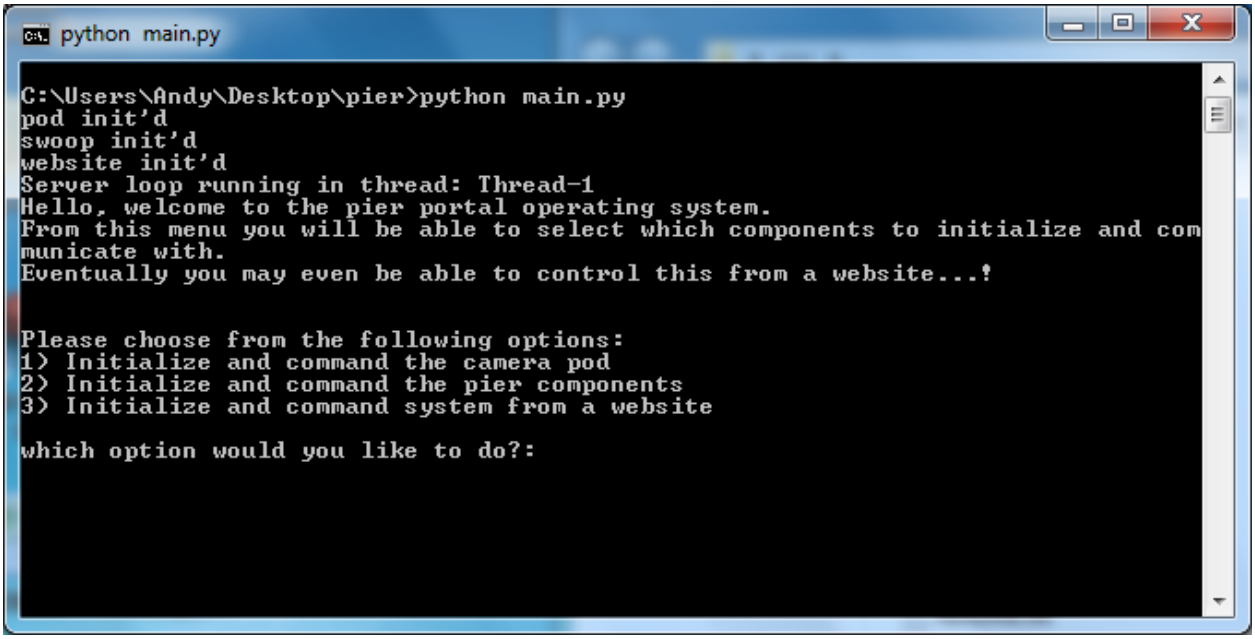

Figure 5: Choosing what commends the pod

If the user chooses to command the pod directly, they will be shown a list of available commands they can type in to control pod.

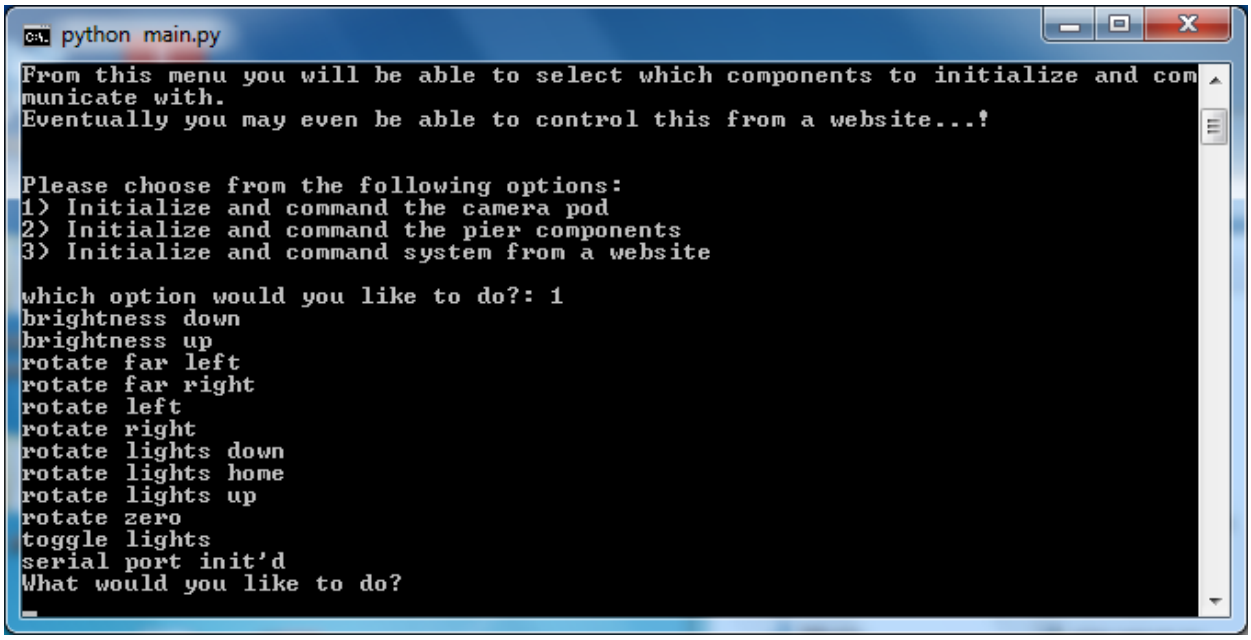

Figure 6: List of commands

If the user chooses to allow the script to receive commands from the website the script will act as a socket server. It will wait for a command from the website and when it receives one it will check the command against a list of commands that the script has stored. If the received command is valid the script will send a command character to the microcontroller in the pod through an RS485 cable.

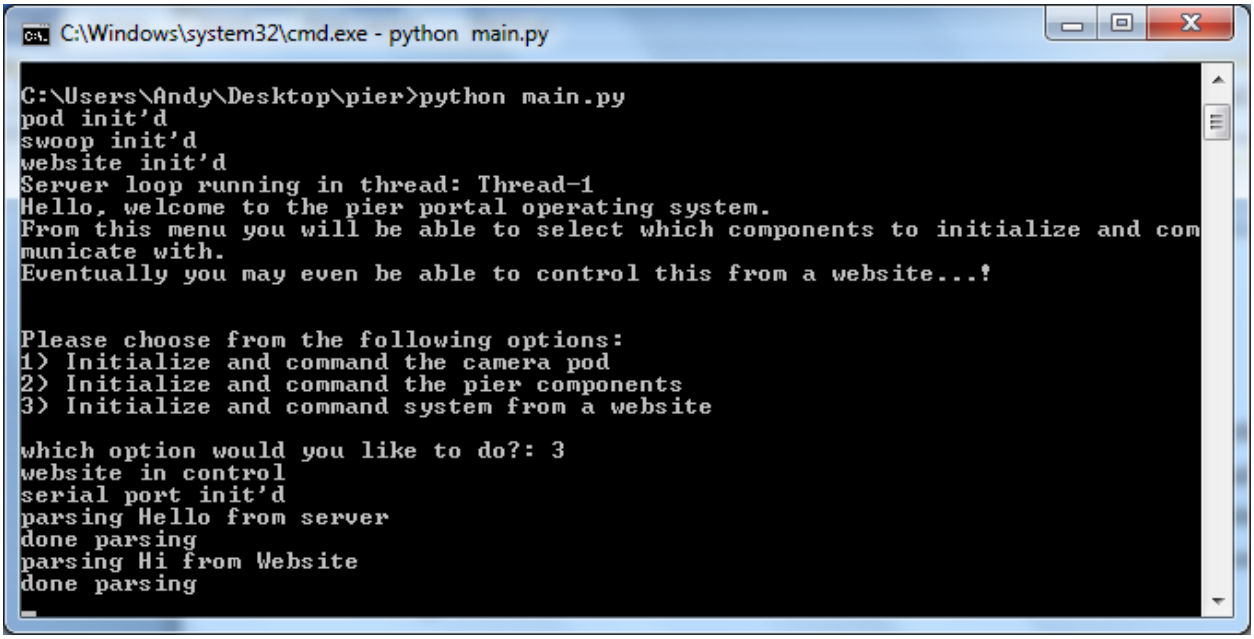

Figure 7: Waiting for command from website

Communications

For website control the Pier Portal system uses many different communication protocols and languages. The back end the website communicates with the pier via a standard TCP socket. This socket provides a link between the website and the various computers at the pier via a Python script that is running on the server deployed at the pier. The website uses PHP to send data over a socket. On the other end of the port there is a python script that is running a threaded server that receives the requests from the website and communicates with the pod microcontroller and will in the future communicate with the recently donated winch controller to appropriately work the system. The Python script uses the 'PySerial' library to communicate with the pod microcontroller board and uses the standard Python struct library for byte packing and management. Due to the length of the pod cable, the previous Pier Portal group chose to use RS-485 for this communications line as it provides a high baud rate at wire lengths of over 100 feet.

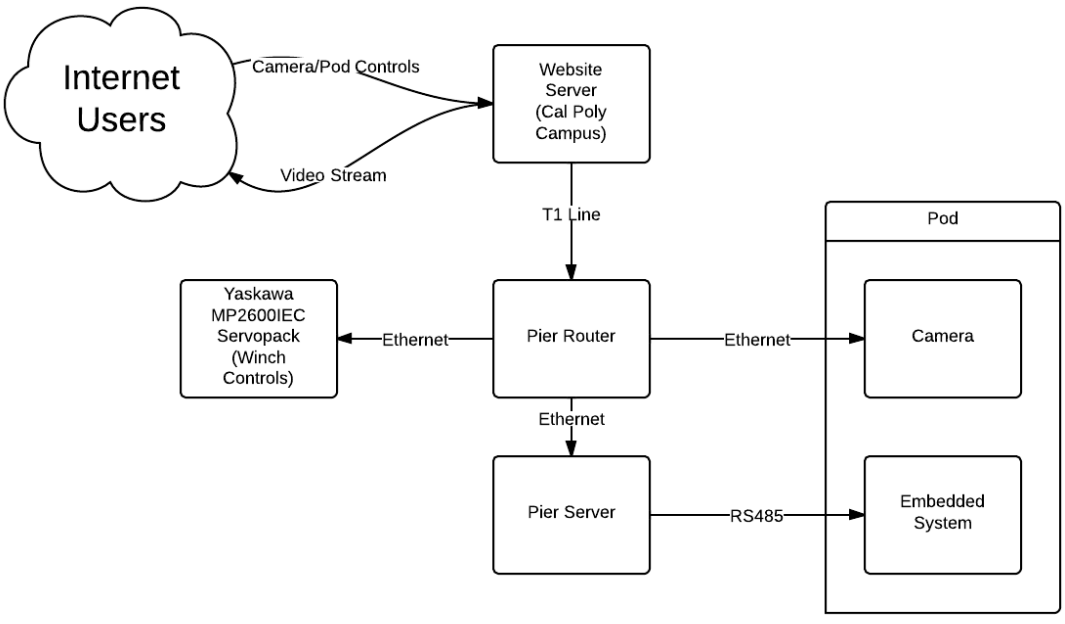

Figure 8: Connection diagram

Last year's Pier Portal team provisioned the server that hosts the website and with the help of Cal Poly's Network Administration set up static IP addresses for the Ethernet devices, and set up pin-hole requests to access the devices remotely. Sometime between the end of Pier Portal I (last year's Pier Portal team) and the start of Pier Portal II (this year's team) Cal Poly's Network Administration changed the network to not allow static IP addresses. Initially, at the start of Pier Portal II, we were unable to access the "Pier Camera" (at the pier and in 192-118) from on-campus computers that were connected via Ethernet to the Cal Poly network or from outside the Cal Poly network. We were only able to connect to the camera while on campus and connected to SecureMustangWireless. With help from Network Administration, we were able to determine that the "Pier Camera" needed to be configured as Manual DHCP (Dynamic Host Configuration Protocol) instead of static IP. Manual DHCP is similar in that the camera will always get assigned the same IPs. When the network sees the camera (identified by its MAC address) on the network and it's on the 192-118 subnet it receives the same specific IP address every time and when it's on the pier's subnet it receives another specific IP address every time. The assigning of the IP from the network is automatic and does not require us to manually change the IP settings when we change locations. In an effort to use as few conductors as possible through the winch cable, the previous Pier Portal team decided to use only 4 data wires for the camera. This is not an issue because the camera is rated for 10/100 Mbps communications, but certain 1 Gbps switches may not recognize the connection due to the open circuit on the four other lines.

Testing

To test the system we opened the website in multiple browsers, checking to make sure that the video stream was viewable and that commands sent from the buttons could make it through to the microcontroller in the pod. Refer to Table 2 for full list of browsers tested and the results for each browser.

Table 2: Browser Test Results

Browser	Operating System	Video Status	Button Status
Android AOSP Browser	Android	Working[1]	Working
Chrome	Android	Not Working	Working
	Linux[2]	Working	Working
	OS X	Working	Working
	Windows 7	Working	Working
Dolphin Browser	Android	Working[1]	Working
Firefox	Android	Working[1]	Working
	Linux[2]	Working	Working
	OS X	Working	Working
	Windows 7	Working	Working
Internet Explorer	Windows 7	Working	Working
Safari	iOS	Not Working	Working
	OS X	Working	Working

[1] Requires Adobe Flash Player for Android

[2] Tested on Ubuntu

Conclusion

The Pier Portal system is not currently finished. For the computer/electrical engineering side of the Pier Portal team this mostly due to the difficulties in getting the camera accessible on the Cal Poly network which were caused by the changes made to the network by the Cal Poly Network Administration. Working with Network Administration to figure out how to get the camera accessible on the entire Cal Poly network and from outside the Cal Poly network, took a lot longer than it should have and prevented us from getting help from our contact, Steven Pearson, from A1 Webcams (the company the Pier Portal I project team purchased the camera from) to develop controls for the camera. After finally getting the camera accessible from outside the Cal Poly network, Steven was able to help us get embeddable website controls for the pan, tilt, zoom, and focus functions of the camera. Unfortunately, since the camera is embedded with ActiveX controls, the embeddable website controls were also ActiveX and as discussed above it is not an ideal solution for the Pier Portal system.

We sought out different ways to convert the ActiveX controls and video. We were able to achieve this for the video stream using the crtmpserver and the JW Player, as described above. For the pan, tilt, zoom, and focus controls we first tried to see if there is way for a Python script to run ActiveX code. Since the “Pier Server” computer is a Microsoft Windows it does have the capability to run ActiveX code. We researched on the internet if this was possible and found different things about Python and ActiveX, but nothing that was applicable to our needs. Since neither of us knows that much about ActiveX, we decided that this was too difficult of a solution to produce in the final weeks of the school year.

Our next idea was to connect the camera to the network and manipulate the controls using the ActiveX buttons/widgets on Internet Explorer, and “sniff” packets that were being transmitted out of the RS485 serial port to be able to determine what the packet structure for the RS485 serial communication for the camera looks like. We asked our contact from A1 Webcams if this was possible and he said that we wouldn’t be able to see any data out of the serial port when controlling via the web because serial communication is analog and via the

web is digital information/communication. Also, the camera end of the serial port is for receiving commands, not transmitting them.

At the very end of the project, Steven mentioned to us that we could run the camera through a DVR and use the DVR to control the camera. The camera would connect to the DVR via its composite/coaxial connector to one of the DVR's camera in/video input jacks on the back of the DVR, as well as RS485 from the DVR to the camera for control. The DVR would connect to the network and would purely be used as a controller for the web. The crtmpserver would still be used for converting the RTSP stream from the camera into an RTMP stream and sending it to the JW Player. JavaScript would be used to control the DVR. Since JavaScript is a simple language, every time a pan, tilt, zoom, or focus command is entered the camera will move a little bit and then stop. It will not be quite as smooth as ActiveX controls but it would allow for more users to use the Pier Portal System.

Since this was suggested to us at the very end of the school year, we did not have time to research different DVR models and then implement the JavaScript code for controlling a DVR to send commands to a camera. We feel that using a DVR would be a great idea for the next Pier Portal team to try. This would also serve as a way to record video from the camera.

Another way to record video from the camera that could be implemented in the future would be saving a local copy of the HD video on the pier. Due to the bandwidth limitations of the T-1 line between the Pier and Cal Poly campus, the camera and website are only able to stream a video with a bit rate of ~1 Mbps. The camera is capable of 1920x1080 video resolution at 30fps and providing up to 8 Mbps of data. A future Pier Portal team could take the raw feed from the camera, send it to the pier server, and save a copy for later viewing while simultaneously streaming a compressed version back to Cal Poly campus using the crtmpserver and JW Player to be streamed onto the website.

One feature that still needs to be implemented is the user login system to prevent unauthorized users from accessing the camera and pod controls. A user queue also needs to be set up to ensure that only one user can control the camera at a time. Administrators also need

to be able to take control away from regular users and assume control over the pod should the need arise.

While there is still a lot left to do on the project, we've made significant progress during our time working on it. Communications have been established all the way through from the website to the pod and the video feed from the camera is no longer locked down to one browser. Significant improvement have also been made in the documentation for the software side of the project which means that the next team can pick up from where we left off with minimal difficulties.

Bibliography

[1] G. Brown. (2013).*DiveCam Camera Inventions by Garrett Brown*. [Online]. Available: http://www.garrettcam.com/camsDiveCam.php

[2] B. Newman. (2013). *Now Diving: Sir Isaac Newton*. [Online]. Available: http://online.wsj.com/article/SB121856740339434067.html

Appendices

Appendix A - Analysis of Senior Project Design

Project Title: Pier Portal II

Student(s): Andy Lam, Brian Markwart

Advisor: Bridget Benson

Summary of Functional Requirements

This project covers the design of an underwater camera system that will stream video to a website. Users will be able to go to the website and see a live video stream from the camera and, once logged in they will be able to remotely control the camera from the website. The website should be browser agnostic, allowing users access to all functions regardless of their choice of internet browser.

Primary Constraints

Because the project was a continuation of a project from the year before, a significant amount of time had to be spent catching up on the work done by the pervious team. Some problems also appeared when trying to get the CRTMP server running due to the lack of good documentation.

Economic

We inherited all of the equipment used in our project because it was a continuation of a project from the year before. That, coupled with the fact that all of our work was mostly software related meant that no extra costs were incurred during the course of this project.

Environmental

The use of our system would allow for easy monitoring of the sea life around the pier. Any environmental changes in the ecosystem could be could be observed by users which would allow researchers to gather information on any environmental threats in the area.

Manufacturability

A significant number of parts for our system were custom built, instead of off-the-shelf parts, which would make reproduction of our system rather difficult.

Sustainability

The system is meant to accessible by anyone at any time over the internet which means that a significant number of devices must remain powered on at all times to ensure proper function of the system. These devices would be constantly drawing power even when not being actively used by someone at the website. One way to mitigate this problem would be to implement a standby mode which would shut off the camera after it has been left idle for a certain amount of time. The current pier server could also be replaced with a computer that draws less power since it doesn't need to perform any particularly computationally intensive tasks.

Ethical

If an unauthorized person were to gain access to the controls of the camera and pod they could use the system to spy on the pier and its surroundings. However, an administrator would be able to log in and take control away from the user.

Health and Safety

The camera plus the other pod electronics can draw currents in excess of 1 amp so if the cable powering the pod should snap for whatever reason, that plus the generally wet environment of the pier could pose a potential safety hazard.

Social and Political

The video stream on the website is intended to be accessible the general public so the system itself could be used to help raise awareness for the environment.

Development

Development of this project required the use of Python, PHP, HTML, C++, JavaScript, as well as experience with embedded systems and some basic networking knowledge.