KGCOE Senior Design Technical Review Pre - Read

Water Falls Media (P11301) - M. DeMayo, J. Mathews, B. Sheron, D. Watkins

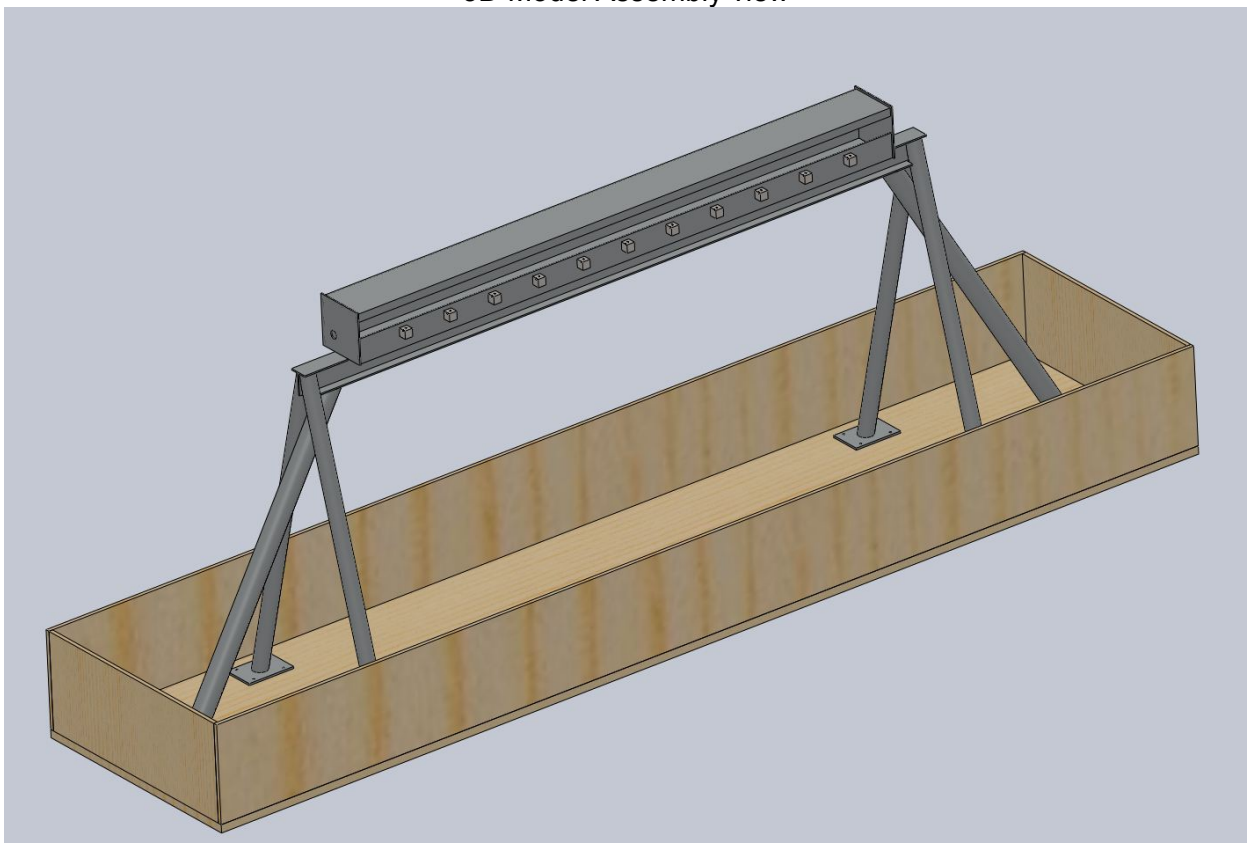
Meeting Purpose : The purpose of this meeting is discuss the design of the water falls media design project. CAD drawings and detailed schematics will be presented and critiqued. Various risks will be assessed and construction plans will also be discussed. Finally, preparation for Senior Design II will be discussed.

Materials to be Reviewed : CAD Drawings (Mechanical), 3D Model (Mechanical), Risk Assessment (Mechanical and electrical respectively), Prototyping (Mechanical and electrical respectively), Construction process (Mechanical and electrical respectively), Simulations and schematics (electrical)

Meeting Date: 11/5/2010
Meeting Location: 09 - 3119
Meeting time : 2:30 - 3:30PM (Mechanical) 3:30 - 4:30PM (Electrical)
Timeline:

	Meeting Timeline	
Start time	**Topic of Review**	**Required Attendees**
2:30	Brief introduction / update from concept review	Slack, DeMayo, Mathews, Schweppe, Landschoot
2:40	Overall design (3D model / general part function)	Slack, DeMayo, Mathews, Schweppe, Landschoot
2:50	CAD drawing review/ critique ideas Construction process/ arrangement	Slack, DeMayo, Mathews, Schweppe, Landschoot
3:10	Updated specifications	Slack, DeMayo, Mathews, Schweppe, Landschoot
3:20	Closing thoughts/ questions	Slack, DeMayo, Mathews, Schweppe, Landschoot
3:30	Introduction / Controller Overview	Slack, Watkins, Sheron, Schweppe, Patru
3:40	Pseudo Code	Slack, Watkins, Sheron, Schweppe, Patru
4:00	Schematics Layout	Slack, Watkins, Sheron, Schweppe, Patru
4:20	Closing Thoughts / Questions	Slack, Watkins, Sheron, Schweppe, Patru

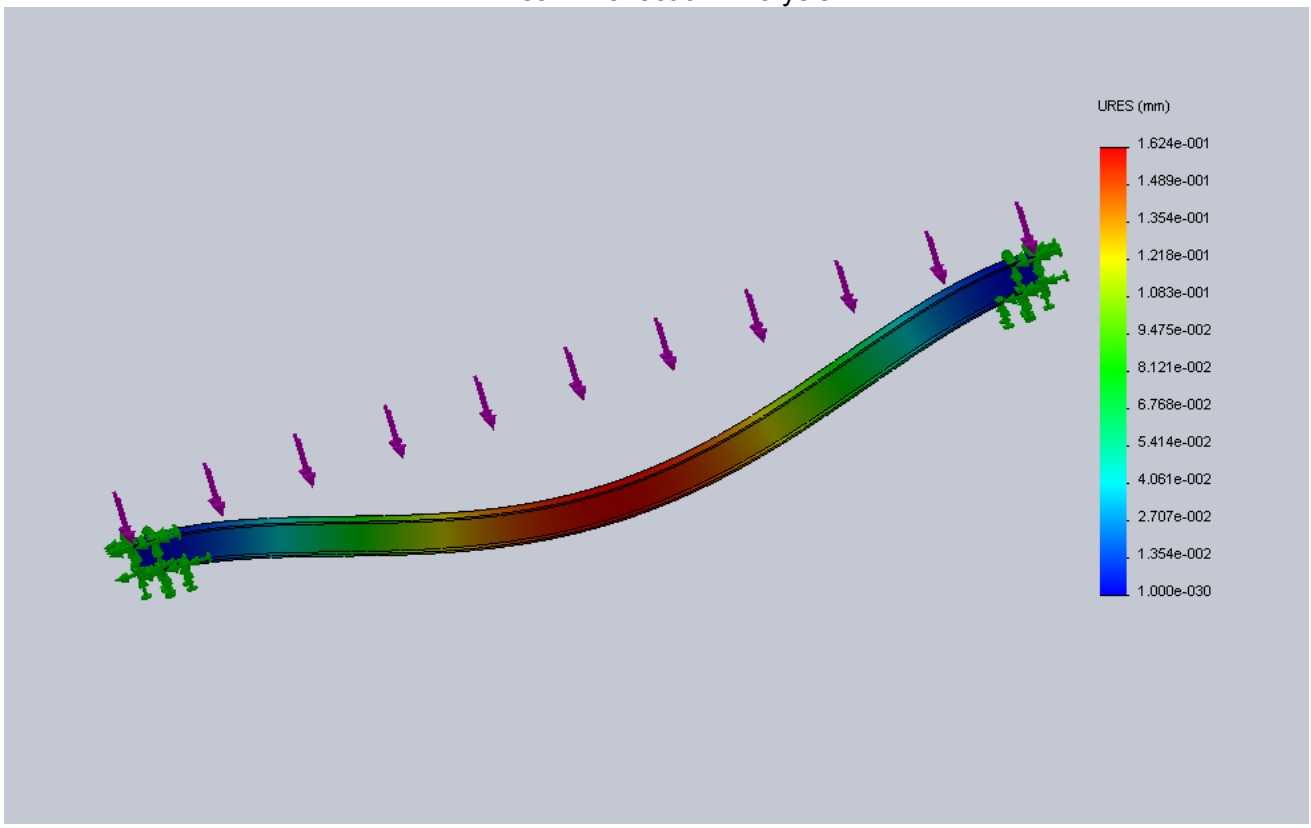
3D Model Assembly view



The materials to be used in this waterfall are mostly wood and aluminum. The pouring lip, electrical equipment, wheels, pump, hose, and fastening equipment are not shown in this drawing for simplicity.

The total assembly will stand 48” tall and will produce a fluid curtain that is 72” wide and 45” tall. The upper trough consists mainly of an Aluminum box beam that is 72” wide with a 6” x 6” cross section. The pouring spout will be made from bent aluminum sheet to get as flat of a lip as possible. The catch trough is 132” long, 13” tall and 30” wide. The total assembly will approximately stand 50” tall.

I-Beam Deflection Analysis



We ran an analysis of the I-Beam to be used for the main trough support, to see what the maximum deflection of it would be. This is important, because if the deflection is too great, the water will not pour out of the trough smoothly. Under an 80 pound load, 50 pounds from the water in the trough, and 30 pounds from the trough itself, the deflection in the I-Beam is only 0.1624 mm, which is well below the maximum value of 1 mm.

Mechanical Information to be Discussed

- Estimated cost, and the Bill of Materials
- Overall mechanical structure of catch trough, wheels, support pipes and I-Beam
- Detailed drawings of mechanical structure
- Main trough, pour lip, and adjustment blocks
- Feasibility analysis on the water curtain
- Updated engineering specifications
- Risk assessment by component
- Construction/ processing plan
- Ability to meet customer needs and engineering specifications

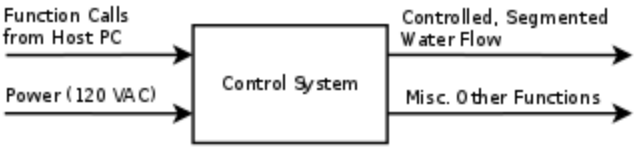
Control System and Power Supply

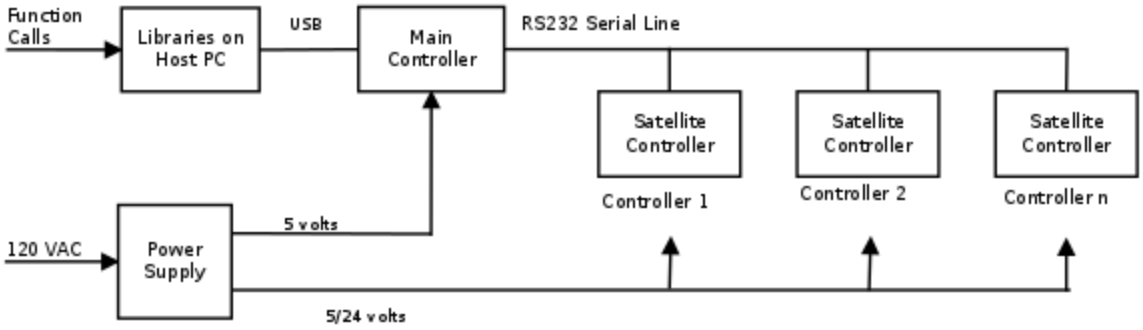
Segments of the waterfall are to be able to be opened and shut off electrically via solenoids, each one controlling a small damn to be placed over a section of the trough. Each solenoid will be controlled by a small AVR microcontroller via a power MOSFET. Each microcontroller will listen on an RS232 data bus, which will be fed by a single main controller. (Thus, the main controller will be the only one transmitting, while the rest receive.) Each slave microcontroller will be assigned an address, and will listen for commands to that address. The main controller will communicate as a USB device with a host PC, and relay instructions to the appropriate slave controller to turn its designated portion of the waterfall on or off.

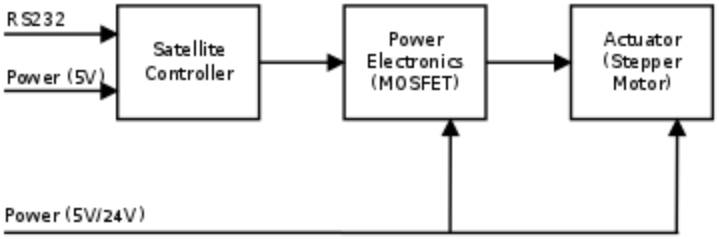
Interfacing with the host PC will be performed by an open-source C library (Object Development's V-USB), with components running on the main AVR controller as well as on the host PC. On the host PC the C library will be able to be used in a variety of applications, namely in a Python library, allowing the client to control the device in a familiar programming environment.

Two power supplies will be provided for low voltage DC: one 24 volt supply for the solenoids themselves, and one 12 volt supply for the control electronics. (The 12 volt supply may be lowered to other voltages as needed via linear regulators.) The pump motor will be fed by the 120 VAC mains. The power supplies and main controller will be housed in a waterproof box on the side of the unit. Wiring will be run in waterproof conduit, and a waterproof shield will protect the wiring and electronics along the top of the trough.

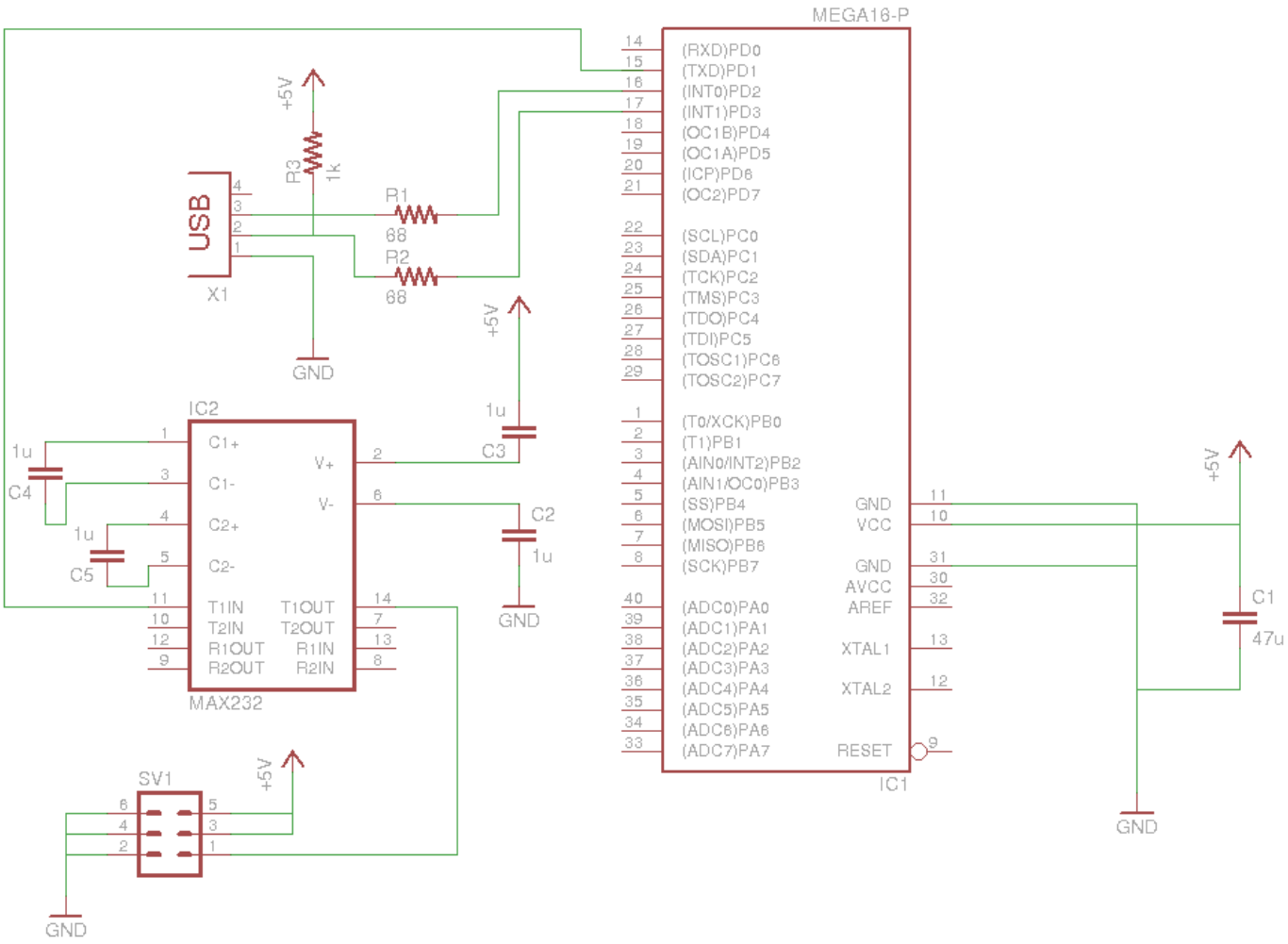
Waterfalls Media Controller Functional Decomposition



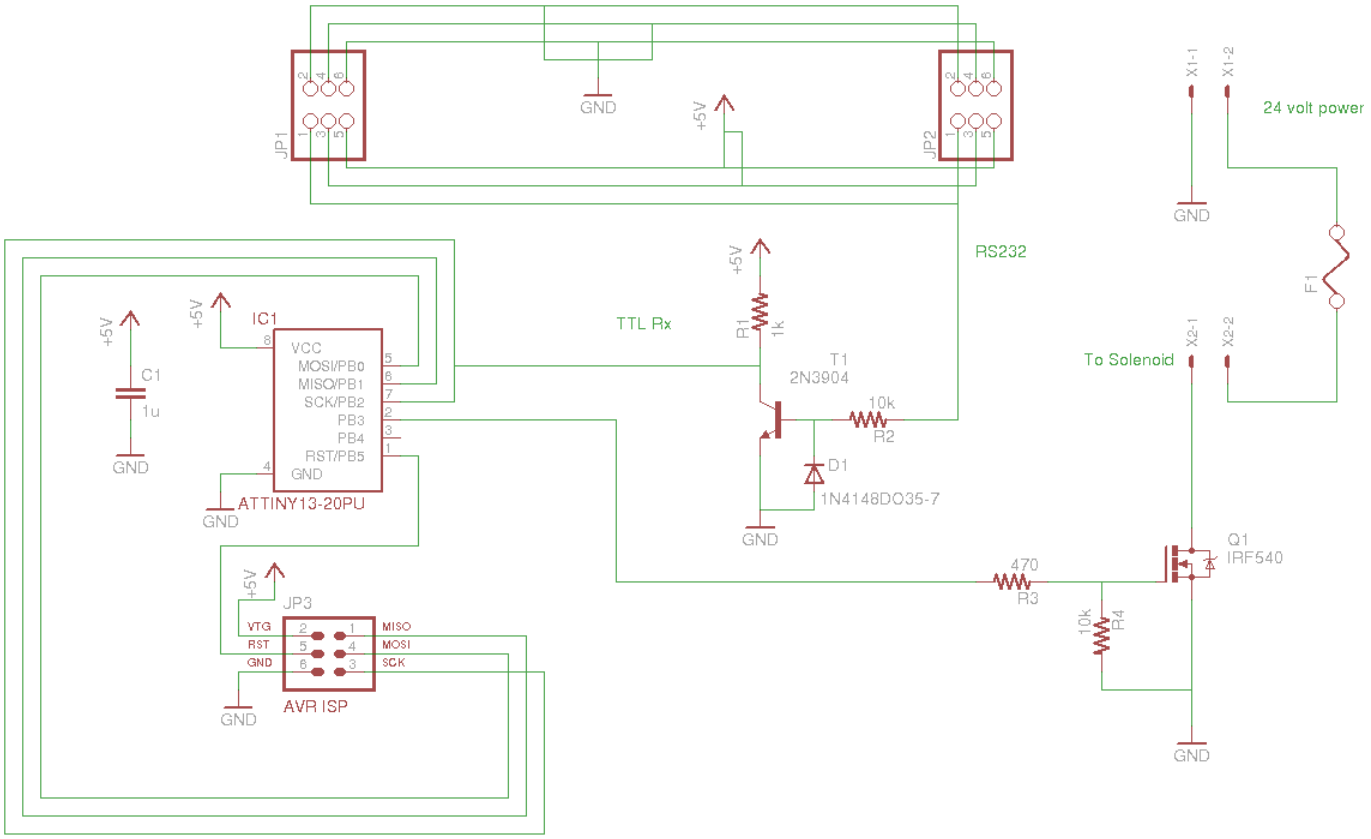




Main Controller Schematic



Slave Controller Schematic



Pseudo Code

Host PC

```
On block function call in Python library
    Execute C function to send command to USB device to block a
certain segment
```

The host PC functionality is basic. The V-USB C library can be abstracted to a function allowing a command (in this case, to block a segment of the waterfall) to be sent to the main controller. This function can in turn be called from a Python function via a wrapper. Using simple program flow control it should be possible for the client to create a variety of different water effects.

Main Controller

```
Initialize serial out
Initialize USB
while(1) {
    if block segment command {
        send to serial "turn segment off"
    }
}
```

After initialization the controller will wait in an infinite loop for a command from the host PC. Upon receiving a command to block a segment, it will then relay this command along the RS232 bus. This controller is in place to provide expandability, allowing other devices to be controlled in the future. (For instance, lighting may be controlled via the controller's PWM features.) Thus,

it is not necessary that this behave simply as a USB to RS232 converter.

Slave Controller

```
Initialize serial in
Initiazlize digital output
Set digital output to 1
while(1) {
    if this controller's address is read {
        if action is to block water segment
            send a logical 0 to output
        if action is to unblock water segment
            send a logical 1 to output
    }
}
```

Each slave controller sits in an infinite loop. Due to the nature of the solenoids a logical 1 will cause it to pull the dam up, allowing water to flow. Since the water should flow by default, the output is set to 1 when power is applied and after the necessary initializations. The controller will listen on the serial bus for its address, and upon finding an instruction intended for it execute the appropriate action.