

Lucas Vickers
Technical Proposal for Blue Flower / Flor Azul
Prepared for Federico Muelas

1. Abstract

The Blue Flower / Flor Azul project combines a low tech medium (ink and water) with high tech mediums involving digital rendering, image (shadow) tracking, and a high powered optical projection system. The seamless combination of mediums is part of what gives the Blue Flower its appeal. However, in order to make this project successful, the various technologies need to function together with minimal down time and human intervention. An additional obstacle is the fact that this system will be installed outside in New Mexico. First, a system that can withstand high temperatures from within an enclosed environment (without active ventilation) is required. Second, as the artist and technologist are located in New York City, the system needs to be remotely debuggable and programmable. The system also needs to be built in a modular fashion so components can be easily replaceable in the event of hardware failure.

2. Objective

The functional requirements for this project have already been described by the artist. The objective at hand is to provide a reliable system to accomplish the following technical goals.

- Provide a means to program, debug, and maintain the system from a remote location using only an internet connection.
- Successfully analyze an image of the water formation and derive the pixilation information. This requires first determining the circular area of interest, analyzing the darkness of each pixel, and then converting the internalized image into a pixel resolution that equally matches the display screen. The coloration information of these pixels can then be used to determine whether each panel is dark or light.
- Successfully analyze an image of the display screen and determine where the building shadow is being cast. Apply this information to the circular area of interest, and then determine an accurate segmented line formula representing the shadow. This line will be used to determine which pixels are in render mode or reflect mode. A more complex shape can be used to track a variety of shadows if feasible.
- Utilize existing TCP/IP and or FTP protocols to upload images from both cameras and algorithmic state information to a university web server. The images can be used by the University for a webcam, and the algorithmic state information can be used for tweaking of the tracking algorithm (to be discussed in a later section). While the system will internally operate at a rate of 1 frame per second images will be uploaded at a slower rate, such as 1 frame every 5 seconds.

- Provide a reliable means of communication between the main processing board (which will be housed in the projector) and the circuitry which controls the display (housed in each wall panel). Due to the distance simple serial communication will not be reliable, *especially* if the communication wires are run along the same path as any power wires.
- Provide an interface to the university so that custom messages can be displayed on the screen. This interface must be fool proof and easy to use. *This is a secondary requirement in my opinion. I will first focus on the above goals.*

3. Technology

As many commercially available technologies will be used for this project as possible. When dealing with low volume projects it is wiser to buy products which have been commercially designed and tested, as they will come with technical support and warranties. Generally purchasing all products from the same company is wise as well, as they can provide integration support.

3.1 RabbitCore

An embedded system platform will be used for the main system. When programmed correctly these systems are far more reliable than personal computers due to the focused application and programming of an embedded system. These systems generally contain no moving parts, give off little heat, require little electricity, and can withstand relatively extreme temperatures. I propose using the RabbitCore¹ line from Rabbit Microprocessors. The RabbitCore line can operate in temperatures up to 185 degrees Fahrenheit and only require 3.3volts of power.

3.2 RabbitLink

Rabbit Microprocessors provide a system called the RabbitLink², which allows for programming and debugging of their RabbitCore processors over an internet connection.

3.3 Rabbit RIO and Serial-Ethernet Bridge

The Rabbit RIO³ provides for additional digital outputs. While additional outputs above what the RabbitCore provides are not required, the Rabbit RIO connects to the RabbitCore via a serial connection. This serial connection can be converted to an Ethernet connection using the Serial-Ethernet Bridge⁴. The use of Ethernet will extend the allowed communication distance from roughly 10 feet (serial) to over 300 feet, all the while providing TCP/IP error correction to guarantee reliable communication.

¹ <http://www.rabbit.com/products/CoreModules/>

² <http://www.rabbit.com/products/RabbitLink/index.shtml>

³ <http://www.rabbit.com/products/rio/>

⁴ <http://www.rabbit.com/products/em1500/index.shtml>

3.4 Camera

A camera sold and supported by Rabbit will most likely be used at first. It is yet to be determined how well this camera will perform, and also if the software provided with this camera can be used to interface to other cameras (I suspect it can).

4. Technology: Design Approach

4.1 Network Considerations

All development will be done over a local intranet connection using the RabbitLink from day one. This will allow me to test the functionality of the product and ensure that remote debugging and programming is in fact reliable. It will also provide me with the opportunity to learn any quirks and tricks before I am put in a strenuous debugging situation.

The university will provide access to the net or subnet which hosts the RabbitLink card via a VPN, which will allow for controlled access to the interface. The university will also provide a storage area on one of their web servers for the camera images and debugging information. Utilizing the large storage space available on web servers, a file system will not have to be supported on the RabbitCore. The RabbitCore software development system has integrated support for TCP/IP and FTP, which will allow for the uploading of these files.

4.2 Development Methodology

The down side of embedded development is that debugging is not always easy as visual on-screen feedback is not possible. The system needs to be designed in a way that important information is logged to debug files, which can then be analyzed to either fix bugs or enhance algorithms. This will slow the development time initially but in the long run will provide for enhanced maintenance abilities and will increase system reliability.

I have experience with all of the technologies mentioned for this project except I do not have extensive computer vision experience. I have published multiple papers and I have worked closely with a professor whose background is in computer vision / robotic vision named Rustam Stolkin⁵. He has written multiple tracking algorithms and is willing to provide me with consultation support.

Part of the reason Rabbit Microprocessors was chosen is because they provide a Dynamic C package for programming their microprocessors which comes with software libraries for interfacing with digital cameras, communicating with their IO boards, TCP/IP support, etc. This package also provides for a multithreaded like environment, which will prevent any kind of network delays or network issues from interfering with the screen analysis and rendering.

⁵ <http://www.stevens.edu/ses/ceoe/People/Stolkin/Stolkin%20CV.html>

5. Risks

There are three main risks for this project.

5.1 Dynamic C risk

Dynamic C is not ANSI standard. Dynamic C is used to program the Rabbit Microprocessors. While far better than attempting to write the system in Basic or Assembly, the fact that Dynamic C is not ANSI standard means freely available software libraries (such as openCV for image processing) will not be available for use without porting the software (which requires time). Rabbit Microprocessors provides a software library with their processors, but the non ANSI issue still has the potential to cause the schedule of this project to be delayed.

5.2 Vision Algorithm risk

Vision algorithms are generally designed and trained using real world data. For this project real world data would mean a video feed of a shadow being cast over the display screen in various lighting conditions. This will not be available until after the system is deployed, meaning the initial algorithm to be used in this system has the potential to malfunction. The algorithm will be designed in a robust manner, but time should be given between the installation of the piece and the opening of the piece in order to extensively test and retest the systems accuracy. Tests will be done in my studio using a large printout of the screen and a light source. This can provide for the initial algorithm selection and design.

6. What will and will not be provided, and under what budget

6.1 Core System

I will provide the entire computer system mentioned above, algorithm tweaking and optimization, as well as ongoing support. Hardware will be provided on a separate budget from the software development budget. All hardware will be made available at cost. Documentation for both future programmers and trouble shooting support staff will be provided. If there is a serious issue that can not be fixed by remote debugging or part replacement, I will pay for my travel to fix the system (not including initial installation, see below). Any hardware issues should be covered by Rabbit's 5 year warranty⁶. All future software maintenance will be covered under my budget. Software enhancements will be covered under a future budget.

6.2 Chip Housing

I will not provide the housing for the microprocessors as this is not an area I am skilled in. I will work with you in order to come up a modular design and case system.

⁶ http://www.rabbit.com/store/terms_cond.shtml

6.3 Replacement Components

Replacement components will be programmed and provided at hardware cost. I suggest having at least 1 replacement for each component on hand at the university. If a component malfunctions it should be covered by the Rabbit warranty, however this never happens in a timely manner.

6.4 Initial Installation

I will make myself available for at least a week during the initial installation and testing. I would suggest having me on sight. I will not provide for my own transportation expenses for the initial installation.

7. Budget

I have initiated conversation with a sales technician at Rabbit to have a few remaining questions answered. It is not yet known whether or not I can use one RabbitCore to communicate with two cameras or if I need two RabbitCores. Also, the price of the cameras are not yet known. The type of camera used will depend on input from my vision consultant and information about which type of cameras the Dynamic C libraries interface to. I imagine the cameras will run in the low hundreds, if that. Also there will most likely be a discount on hardware prices, as all of the prices I am listing are for individual unit sales over the internet.

7.1 Hardware Budget USD

Quantity	Component	Price Min	Price Max
	Developmental System		
1	RabbitCore Development Kit	300.00	400.00
2	Cameras	TBD	TBD
1	Rabbit RIO Board	300.00	300.00
1	Ethernet-Serial Bridge Development Kit	350.00	350.00
1	Ethernet-Serial Bridge	250.00	250.00
	Total around 1200 + Cameras		
	Production System (Entire Replacement)		
1	RabbitCore	50.00	100.00
2	Cameras	TBD	TBD
1	Rabbit RIO Board*	TBD	300.00
2	Ethernet-Serial Bridge	500.00	500.00
	Total around 900 + Cameras		

* The price of stand alone RIO boards (non development kits) should be well below 300.

7.2 Software development Budget USD

Service	Price Min	Price Max
Development, Testing, Documentation	-	-
Computer Vision Consultation	-	-

The price of development, testing, and documentation varies based on the source of payment. If payment is coming out of the artist's personal budget I will develop this project for a discounted rate. If payment is coming directly from the university I will develop the system at a standard rate. In the event this system turns out to be far easier to build than anticipated, the development price will be discounted accordingly. The price will not rise based on difficulty of the project.

The price of the vision consultant is dependent on the amount of hours he is required.

8. Time Frame

Function	Effort (days)
Initial development configuration	2
Test Rabbit IO over Ethernet bridge	1
Research vision algorithms	10
Set up PC based test environment for vision	2
Test and tweak vision algorithms	10
Port algorithms to Dynamic C	10
Configure camera interface on chip	2
Configure automatic image uploading	3
Implement algorithm logging	5
Design internal display driver	5
Interface display driver	3
Test	10
Provide university text display interface?	15
Total	78
Total without display interface	63