

# Do-It-Yourself Interface Device Prototyping for Virtual Reality



Peter Wins<sup>1</sup>, Aryabrata Basu<sup>2</sup> and Kyle Johnsen<sup>1</sup>

College of Engineering<sup>1</sup>, University of Georgia<sup>1</sup>

Department of Computer Science, University of Georgia<sup>2</sup>

**Abstract**— the off-the-shelf virtual reality community is well aware of the wide variety of inexpensive and robust tracking and display technologies that have emerged from the entertainment industry in recent years. Alongside these developments, there has also been a vast improvement in the accessibility and usability of hobbyist-grade electronics components, which make incorporating lower-level sensing and actuation components into virtual reality systems feasible for even novices in electronics. Combined, these developments have formed the foundation for a renaissance of do-it-yourself interface device prototyping. Development of novel virtual reality interface devices has been simplified to the point that it can be commonplace in virtual reality system design, rather than an exception. This article discusses a variety of ways in which off-the-shelf electronics and devices can be leveraged and combined in virtual reality experiences, and the knowledge required to begin exploring possibilities, through case studies in prototyping virtual reality experiences.

**Index Terms**—do-it-yourself, electronics, off-the-shelf, prototyping, virtual reality

## I. INTRODUCTION

With an increased focus on motion-based and natural interfaces, the gaming industry has created a wide variety of readily accessible, “off-the-shelf”, virtual reality equipment. This off-the-shelf equipment has vastly reduced the barriers to entry into immersive virtual reality (VR) development, reduced costs in the virtual reality industry, and increased the ubiquity of virtual reality devices. While this trend has received a great deal of attention [1, 2], another trend has emerged that may have equally profound consequences: simplified electronics platforms for sensing and actuation. These electronics platforms enable interface development that previously has been inaccessible to many virtual reality researchers because of lack of electronics experience. Together, the vast array of accessible VR devices and electronics form the basis for “do-it-yourself” (DIY) interface device development, *i.e.* novel controllers, displays, and systems. In this article, we discuss tools and technologies associated with DIY development in VR, and case studies of projects that demonstrate DIY interface device development as a part of virtual reality experience design.



Fig.1: Examples of do-it-yourself electronics interfacing hardware. (Counter Clockwise from Top-Left) Phidgets Bipolar Stepper Motor Controller, Phidgets Relay Interface Kit, Oracle SunSpot, Arduino Uno. All devices are programmable through USB.

## II. DO-IT-YOURELF VIRTUAL REALITY

We focus on two categories of off-the-shelf devices that are used for DIY VR. The first is the set of devices that are sold as pre-packaged products (*e.g.* the Nintendo Wii Remote), typically for use with or as entertainment systems, but which are easily interfaced with a PC. These are off-the-shelf virtual reality devices, in that they can be directly integrated without modification. The second category is the set of electronics platforms that simplify the process of interfacing lower-level electronics components such as sensors and actuators, usually with minimal electronics experience. Both require special consideration when using them for novel interface device development.

### 2.1 Off-the-shelf Virtual Reality Devices

As of 2012, there are a multitude of DIY-oriented off-the-shelf virtual reality devices available in all nearly all categories—stereoscopic head mounted displays (*e.g.* Vuzix Wrap 920, EMagin Z800) and projected displays (*e.g.* Viewsonic PJD6531W), 3D wands (*e.g.* Nintendo Wii Remote, Sony Move, Razer Hydra), haptic joysticks (Novint Falcon) and motion-capture systems (Microsoft Kinect). Of note is that, over time, these devices have been converging towards the performance of devices available from professional virtual reality sources, but without the high expense. For example, the Sony Playstation Move controller contains a 9-axis inertial

measurement unit, similar to that within an Intersense Inertiacube, and provides joystick buttons and an active target for camera position tracking. This device is highly specialized as a controller for the Sony Playstation 3 system, but it can be repurposed as an orientation tracker for VR with little effort beyond that required by a typical VR tracking device. In the past, some companies (particularly those in console gaming) have discouraged such repurposing, but recently the DIY community has seen increased support from in the form of free or low-cost software development kits for these devices (*e.g.* Microsoft Kinect for Windows, and Sony Move. Me programs). Companies appear to have recognized value in a large community of educators, researchers, and hobbyists testing and evaluating new uses for their devices. When official support does not exist, or does not exist for a particular operating system (*e.g.* Linux) there are often community projects devoted towards reverse engineering the interfaces and creating software to support the devices, *e.g.* OpenKinect.

Still, there are limitations and challenges to repurposing off-the-shelf devices in VR systems. Most are packaged for a specialized purpose, and lack the generality of virtual reality devices intended for immersive applications. For example, a head-mounted display intended for watching movies in privacy often does not take into consideration head-tracking integration or blocking out of the real world from the periphery. Similarly, a motion-sensor may be integrated into a joystick, which, while convenient, forces the designer into a form factor, or to remove and reconfigure internal components (if possible). For high-end VR installations, it is likely still a better choice to use custom VR devices for performance reasons; however, for prototyping new devices, using off-the-shelf hardware often enables more convenient, less expensive, and ultimately more rapid, iterative design. In addition, VR systems developed with off-the-shelf devices can be reproduced more easily and cheaply, potentially leading to larger deployment of VR applications and further innovation.

## 2.2 *Do-it-yourself Electronics*

From the perspective of integration into a VR system, DIY electronics are similar to off-the-shelf virtual reality devices. Most provide input or control through an API, largely abstracting the complexities of the hardware involved. The differences are commonly a lack of an enclosure, minimal assembly of circuits, and singular functionality, *i.e.* to read a specific type of sensor, or to control a specific type of motor. An example could be a DC motor controller: after two to four wires are connected, a programmer can simply set the speed of the motor, free-spin, or brake the motor without the knowledge needed to develop the electronic means by which this is accomplished (*e.g.* pulse-width modulation, an H-Bridge, and a microcontroller).

Several companies have emerged that specialize in the development and sale of DIY electronics. One of the best examples is Phidgets, Inc. ([www.phidgets.com](http://www.phidgets.com)). This company developed from a research project at the University of Calgary to enable simplified prototyping and programming of physical

interfaces [3], and to support growing trends in ubiquitous, pervasive computing. Commonly provided is a component that contains the necessary electronics to interface a sensor (*e.g.* a temperature sensor) or actuator (*e.g.* a motor). The component connects to the host PC through a Universal Serial Bus (USB) interface, or through another similarly ubiquitous communications technology such as Bluetooth or Wi-Fi. Many times the electronics are multi-purpose, able to interface a number of different types of sensors or actuators, but some are also intended for use with a specific component, usually if that component requires substantial circuitry to interface or condition properly. The goal of these components is to produce basic sensing and actuation that is plug-and-play, hiding the details of the electronic circuits needed to accomplish the task from the end-user.

A more complex class of DIY electronics simplifies the programming and interfacing of microcontrollers. Two recent examples can be found in the Oracle SunSpot ([www.sunspotworld.com](http://www.sunspotworld.com)) and the Arduino platforms ([www.arduino.cc](http://www.arduino.cc)). These platforms feature development environments that are familiar to many programmers, abstracting the complexities of transferring programs to microcontrollers, and allowing use of higher level languages than is typical in embedded systems. The SunSpot platform uses Java and the Eclipse IDE, and Arduino uses a variant of the Java-based Processing Environment (although it is programmed in C++). Extensive libraries exist for these platforms that simplify the process of performing certain tasks, such as reading analog to digital converter values into single functions. In addition, the hardware incorporates circuitry for power, communications, and provides connections for external sensors and actuators.

The trade-off with DIY electronics approaches is that they are typically an order of magnitude more expensive than the sum of their underlying parts. This is in contrast to off-the-shelf devices such as the Microsoft Kinect, where the hardware product has little profit margin, and software sales are expected to compensate. Thus, the benefit for virtual reality in using these components is not directly monetary. Instead, it is the reduction in complexity of interfacing the physical world outside of what is possible with commercially sold tracking and display systems. While off-the-shelf electronics may not necessarily substitute for professional products, they can substitute for the knowledge and time required creating professional electronics, which may be of great value.

Many of the components offered by companies require no prior knowledge of electronic circuits (*e.g.* Phidgets), or provide a gentle learning curve (*e.g.* Arduino). From the perspective of a virtual reality engineer, the difficulty of integration is similar to a typical tracking system or immersive display. The only “electronics” knowledge that is generally needed is an understanding of voltage and current constraints in actuating certain components, such as motors. Often, however, the electronics and sensors/actuators are sold in units, with intentionally matched specifications.



Fig.2:A user controls a parachute simulator constructed with off-the-shelf electronics and gaming devices.

### III. CASE STUDIES IN PROTOTYPING

To illustrate the potential of DIY interface device prototyping in virtual reality, we consider virtual reality systems and interface device prototypes that have been created in our laboratory, many as student projects, but that utilize DIY off-the-shelf VR devices, electronics platforms, and often a combination thereof.

#### 3.1 DIY Parachuting Simulator

The goal of this project was to create an interactive, immersive simulator for the experience of parachuting from an airplane, and controlling your decent. One of the primary goals of this project was to simulate air resistance. The end-result can be seen in Figure 2. A Phidgets Relay Interface Kit 0/0/4 controller was used to switch on or off three large box fans that produced a wind effect proportional to the speed and direction of travel through the virtual environment (by switching the appropriate fans on and off at a proportional duty cycle). This project incorporated two Sony PlayStation Move controllers affixed to elastic cords to control the parachute. These were tracked with Sony Playstation Eye cameras and controlled the direction of travel. Also seen is the EMagin Z800 HMD to create an immersive visual experience.

This project is a useful example of the potential of using DIY electronics to control more of the real world than is typically controlled in virtual environments. Despite the best efforts of virtual reality engineers to isolate users in virtual environments, the real environment still plays a substantial role in a VE. The temperature, humidity, or light level could affect the VE experience substantially. For example, a training simulator for fear-of-the-dark (nyctophobia) exposure therapy would certainly want to control the ambient lighting. Using off-the-shelf electronics, the ambient light in the real room could adjust according to the lighting in the virtual room. All that would be needed would be a light sensor (to detect luminosity), a light-switch dimmer (to set the ambient light level), and a servomotor (to move the dimmer handle). Other environmental controls could be used similarly, *i.e.* operating a

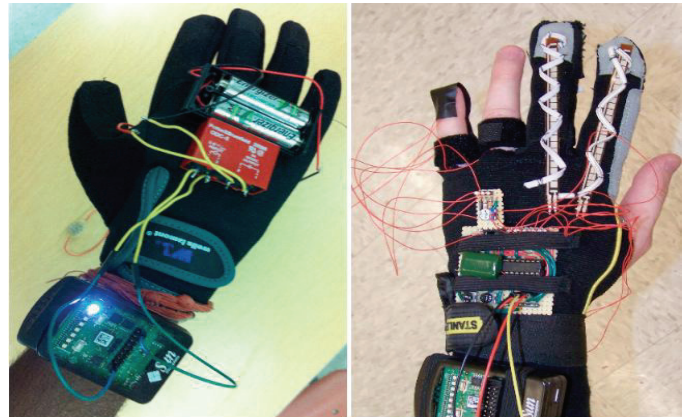


Fig.3:Prototype wireless data gloves (left), and thermal haptic glove (right): multi-sensor gloves incorporating bend, force, and physiological components.

humidifier/dehumidifier, or controlling the temperature through heating and cooling. Such effects are rarely explored, but are potentially tremendously powerful to create a sense of presence.

#### 3.2 DIY Data Gloves

There are hundreds of different types of sensors, each having many varieties and capabilities. The same is true for actuators; each combination could result in a novel VE interface that may have benefit for a particular class of applications. However, the community of researchers and engineers creating novel hardware interfaces is currently small because of the electronics background necessary to innovate. Prototyping with off-the-shelf electronics reduces this barrier to entry, enabling users of devices such as the Wii Remote to enhance and potentially design a better Wii Remote. Classic joystick style controls, buttons, accelerometers, gyroscopes, digital compasses, and vibration motors are all accessible through off-the-shelf electronics. Even the components required to build active haptic devices (*e.g.* high-torque servos) are now readily accessible. Furthermore, one could incorporate sensors and actuators that would not be common in gaming, but could be useful in virtual reality.

Two prototypes can be seen in Figure 3. The first is a data glove, which in addition to sensing finger angles using bend-sensors, also sensed the amount of force applied at the fingertips, tilt of the wrist, the heart rate of the user using a fingertip pulse sensor, and palm sweating through skin conductance sensing. The intended application was medical palpation-based examination training. The second prototype was an extension to the original glove that incorporated a thermoelectric cooler/heater within the glove, creating a thermal-haptic device. This was used as part of a virtual fire-fighting environment, where a heating sensation was stimulated in the palms when the user was within a certain proximity to virtual flames.

The core DIY electronics platform for these projects was the Oracle SunSpot platform, which is primarily designed to teach wireless sensor networks, but has become popular in the physical interfacing community because of its simplified





Fig.4: A DIY Mortar Simulator. The microcontroller, Razer Hydra, and mortar props can be seen at the bottom.

Java-based programming model [4]. It also offers wireless deployment of the sensors, which is uncommon in other off-the-shelf electronics products. Using the SunSpot hardware, it was possible for application-specific immersive interfaces to be rapidly constructed with little electronics knowledge.

### 3.3 DIY Mortar Simulator

The goal of this project was to create a simulator for the experience of firing a mortar, a cannon that features high arcing projectiles fired at an enemy. Similar to the parachuting simulator, a desired feature was a haptic effect, in this case, a passive haptic effect. The user would be able to pick up one of a number of types of mortar rounds, aim the cannon, and drop in the mortar, which would then fire the correct virtual projectile in the aimed direction. To enable this, a tube was connected to a tripod camera mount, which was tracked in position and orientation with an attached Razer Hydra motion controller, which has an integrated electromagnetic tracking device. Passive props were used for mortar rounds, and each was outfitted with a unique radio frequency identification tag. An RFID sensor component was placed on the top of the tube, which when inserted, detected which type of mortar was being inserted. The firing of the mortar (when it strikes the bottom of the tube) was precisely timed by a force sensitive resistor placed at the bottom, whose value was polled continuously by an Arduino Uno microcontroller. Finally, a large, flat-screen television capable of active stereoscopic viewing with shutter glasses was used as the display. These televisions are inexpensive, widely available, and are a reasonable alternative to more commonly employed projected displays used in VR simulators. Figure 4



Fig.5: Two rotations of a prototype rotating display which senses and is controlled by an audience of users through a Microsoft Kinect (top). A Phidgets Bipolar Stepper Motor Controller and Stepper Motor are used to move the turntable (bottom).

shows the completed simulator.

This project was an excellent example of a sophisticated simulator being constructed rapidly. The project took only three weeks to complete, and most of the time was spent on the virtual simulation. However, having a realistic, tangible interface greatly increased the utility of the simulator, as it immediately enabled a number of useful features, such as being able to understand physical constraints and loading timing that would have been difficult to simulate virtually.

### 3.4 DIY VR Platforms

A DIY approach is not limited to building new applications or controllers for VR, but can also be used to construct platforms upon which VR experiences can be built. Three examples of this that we have built are a user-following rotating display, an extremely low-cost augmented reality display, and a portable immersive virtual environment.

*Rotating Display:* A common task in a virtual environment is to manipulate and visual a virtual object. To be able to physically walk around and examine the virtual object requires a combination of user tracking and an advanced display, such as an HMD, CAVE, or immersive workbench. A planar fish tank display can approximate this approach cheaply, but it was developed after recognizing that a planar fish tank VR display only occupies the maximum visual angle in the user's view when its normal aligns with the viewing direction. As seen in Figure 5, our rotating display is an ordinary display mounted on

top of a computer-controlled turntable, and combines several low-cost, easy-to-assemble, off-the-shelf components to create a unique viewing and control solution. A stepper motor was mounted below a turntable, and a Phidgets Bipolar Stepper Motor Controller board was used to send simple rotations to the device. A Microsoft Kinect mounted above the display tracks users (24 joints each) relative to the display within the vicinity. Users can then direct the display to follow them with a hand gesture (hand above the head). From this information, a control algorithm attempts to maintain a zero angle between the display surface normal and the controlling user's head position, sending commands to the Phidgets controller to adjust when necessary. The current angle, desired angle, and user poses are relayed over a virtual reality peripheral network (VRPN) interface [4] to enable VR experiences to fully leverage the display, by generating perspective correct views (*i.e.* fish-tank VR) when the desired angle is different from the current angle.

The rotating display project is currently being used to study the various ways in which physically moving around a virtual object can be used to better understand visualizations. It can also be used to enable an optimal viewpoint for mixed reality experiences, where real objects, *e.g.* a patient or object being assembled can be manipulated and moved around, with the display following.

**Augmented Reality Head Mounted Display:** High-end augmented reality requires the use of an HMD, either optical or visual see-through, with cameras for tracking the environment for immersive overlay. Typically, these displays are expensive, the least available display that we could find being \$1500 (Vuzix Wrap 920AR). In reality any HMD can be used as a camera-see through HMD, provided that cameras are mounted to the display. The difficulty is that the cameras need to be fairly high resolution (640x480), lightweight, and with a fast update rate (*e.g.* greater than 60HZ) to maintain the augmented reality illusion. Fortunately, the Sony Playstation Eye camera is such a camera, available at low cost. By removing the enclosure and mounting two of the cameras side by side, we created a stereoscopic camera bar that can be combined with any stereoscopic capable HMD to form an augmented reality HMD for less than \$100 USD (The Eye camera retails for \$40 USD). We also wrote software that allows the Eye cameras to track the Sony Move controller spheres to enable hand tracking within the visual field, and combined this with ARToolkit [5] to enable passive marker-based tracking. Our implementation uses a Vuzix Wrap 920 HMD (\$250 USD), and can be seen in Figure 6. The only issue currently is that the field-of-view of the cameras (57 degrees) is not matched to the HMD's (35 degrees). Changing the lens on the cameras is possible, and inexpensive, or a larger field-of-view HMD could be used. However, the lack of an inexpensive wide field-of-view HMD is the currently the biggest barrier to DIY VR.

**Portable Immersive Virtual Reality:** The last project illustrates a generic, portable, immersive virtual reality platform built entirely from DIY VR components. We describe this

more fully in [6]. Figure 7 illustrates the key components. We provide body orientation sensing by using an inertial sensor within a smartphone (iPhone 4s) mounted on the hip. Also mounted on the hip is the base of a Razer Hydra controller (the electromagnetic source). One of the Razer Hydra controllers is held by the user, providing control over a virtual hand and pointing-based navigation around the virtual environment. The other controller has its body removed, retaining only the electromagnetic sensor. This component is mounted to a Vuzix Wrap 920 HMD, providing position and orientation of the HMD relative to the hip-mounted source. The rendering of the virtual environment comes from the smartphone, the most recent of which are becoming highly capable 3D renderers. The final component is a battery-powered Beagleboard single-board computer, which provides the USB interfacing of the Razer



Fig.6:DIY portable immersive virtual reality. A user is outside in immersive virtual reality, with rendering and processing components mounted on a belt beneath his shirt.

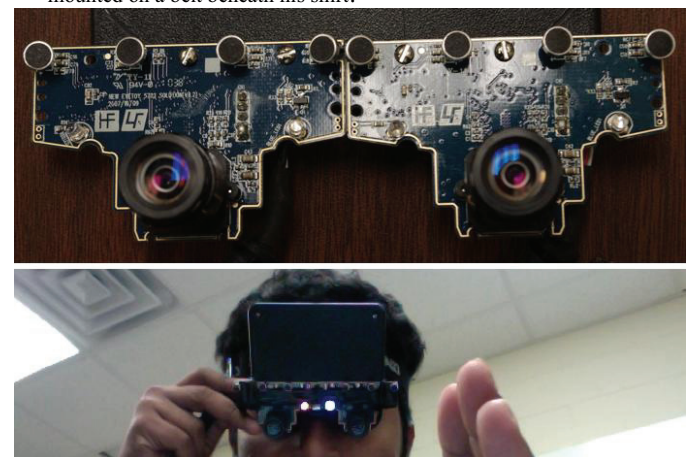


Fig.7: (top) DIY camera-see through augmented reality bar created from the internal components of two Sony Playstation Eyes. (bottom) The augmented reality bar mounted to a Vuzix Wrap 920 HMD.

Hydra, and communication with the smartphone over an ad-hoc Wi-Fi network. For around \$1000 USD, a relatively complete DIY immersive virtual reality platform can be constructed by



anyone, and be used for simulation, visualization, or gaming.

#### IV. CONCLUSIONS

The availability of inexpensive, high quality off-the-shelf VR components, and DIY components is providing VR researchers and engineers the opportunity to improve existing applications and invent new products much faster (and sometimes much cheaper) than by using conventional electronics design techniques. We presented several DIY projects, which leverage these components to improve or innovate in virtual reality applications. We hope that these projects serve as inspiration to begin learning about VR and incorporating sensors and actuators into novel VR experiences.

We feel that the primary limitation of DIY is the same for other off-the-shelf equipment: lack of customization. Creating miniature components is challenging, and without custom printed circuit boards, there may be many wires and cumbersome devices involved. In addition, when considering taking a DIY electronics approach for sensing some phenomenon, one must also consider that standard computer equipment (*e.g.* cameras) may be better suited for certain tasks (*e.g.* sensing light levels), provided that the processing power is available to work with them. Finally, the greatest challenge in creating a custom interface is arguably designing and fabricating mechanical components (such as the turntable of Section 3). There is little off-the-shelf support for this, although 3D printing and laser cutting technology may make this process more accessible. Already there exist companies that specialize in low-volume orders for this type of fabrication, *e.g.* Ponoko and TechShop.

To begin learning about novel VR possibilities, we suggest exploring the wealth of information available at the following websites:

##### USB Sensors and Controllers

- <http://phidgets.com/>
- <http://vernier.com/>

##### DIY Electronics

- <http://sunspotworld.com/>
- <http://arduino.org/>

##### 3D Printing and Construction

- <http://ponoko.com/>
- <http://techshop.ws/>

##### Off-the-shelf Virtual Reality

- <http://www.vrgeeks.org/>
- <http://www.cs.unc.edu/Research/vrpn/>

#### ACKNOWLEDGEMENT

We especially thank the students involved in creating the various projects described in this paper: Angela Zachman, Claudette Lampkin, and Matthew Paulishen (DAWG); Jesse Smith and Paul Schneider (Parachute Simulator); Eun Sun Kwon, Yu Luo, and Chulwoo Lim (Firefighter VE); and Daniel Tanner, Wesley Spires, Grayson Jackson, and Kyle Murphy (Rotating Display); Ryan Thibodeau (Mortar Simulator); Erik

Refour and Christian Saupe (Portable VR).

#### REFERENCES

- [1] J. C. Lee, "Hacking the nintendo wii remote," IEEE Pervasive Computing, pp. 39-45, 2008.
- [2] C. A. Wingrave, B. Williamson, P. D. Varcholik, J. Rose, A. Miller, E. Charbonneau, J. Bott, and J. J. LaViola Jr, "The wiimote and beyond: Spatially convenient devices for 3d user interfaces," Computer Graphics and Applications, IEEE, vol. 30, pp. 71-85, 2010.
- [3] S. Greenberg and C. Fitchett, "Phidgets: easy development of physical interfaces through physical widgets," in ACM User Interface Software and Technology, 2001, pp. 209-218.
- [4] D. Simon, C. Cifuentes, D. Cleal, J. Daniels, and D. White, "Java™ on the bare metal of wireless sensor devices: the squawk Java virtual machine," in ACM Virtual Execution Environments (VEE), 2006, pp. 78-88.
- [5] H. Kato and M. Billinghurst, "Marker tracking and hmd calibration for a video-based augmented reality conferencing system," 1999, pp. 85-94.
- [6] A. Basu, C. Saupe, E. Refour, A. Raji, and K. Johnsen, "Immersive 3DUI On a Dollar a Day," in IEEE Virtual Reality, 2012.



**Peter G. Wins** is pursuing a B.S degree in Computer Systems Engineering from the University of Georgia. He currently works as an Undergraduate researcher in the Virtual Experiences Laboratory, headed by Dr. Kyle Johnsen. His research interests include wireless sensor networks and the off-the-shelf virtual reality.

**Aryabrata Basu** is pursuing a Ph.D in Computer Science from the University of Georgia. He holds a B.S. degree in Information Technology (West Bengal U. of Tech, India). Following graduation, Mr. Basu worked for 2 years on various research projects for Indian Statistical Institute and Bose Institute in Kolkata, India. Since coming to University of Georgia as a doctoral student in 2009, his research focus has been on Engineering Virtual Reality based solutions, innovating designs and studying Virtual Reality architecture. He is also engaged in environmental services, which includes developing mobile applications for Marine Debris collection and recycling items. He is currently a Research Assistant at the Virtual Experiences Laboratory (VEL), headed by Dr. Kyle Johnsen. Mr. Basu is currently affiliated as IEEE student member and ACM student member.



**Kyle J. Johnsen** is an Assistant Professor in the College of Engineering at the University of Georgia. He holds a Ph.D, M.S. and B.S. in Computer Engineering from the University of Florida, specializing in immersive virtual reality. He is currently director of the Virtual Experiences Laboratory, and teaches computer systems engineering courses in virtual reality and sensor networks. His research interests surround the integration of embedded systems and off-the-shelf components to create novel human computer interfaces in a variety of areas from virtual reality to environmental engineering.