DCU Engineering Final Year Project Status Report

Title: A Wireless 3D Embedded RTOS Human Computer Interface

Student Name: Ciaran O'Donnell

Student ID: 15414048

Description:

This project is to design and implement a wireless 3D Human-Computer Interface (HCI). It will be implemented using next-gen wireless devices and using a real-time operating system. The device should allow the user to be able to wirelessly interact with a 3D-environment within the computer. This device may be implemented by integrating accelerometers, gyroscopes and distance sensors interfaced with an embedded device running the real-time operating system (RTOS).

Contents

Table of Figurse

1. Problem and Boundaries

1.1 Motion control

The design of a wireless 3D human machine interface poses many engineering problems to over-come. One of which is the design of the device in order to track and record movement. The device must be able to perceive its motion and be able to track its position in space as well as its orientation within 3-axis. This may be done via implementing accelerometers, gyroscopes, and distance sensors and through research more solutions may become apparent.

1.2 Programming

As well as interfacing these devices the complete device will be programmed using a real-time operating system on an embedded device. This will allow the device to be extremely power efficient, and responsive. However, using a real time operating system means a lot of care will have to be taken to design an efficient operating system. To design such code a lot of research and training will be needed to learn about elements of real-time operating systems such as interrupts, thread, i.o. control, as well as learning to interface with the chosen language and embedded system the device will be based on.

1.3 Wireless Communication

Another engineering problem to overcome is the transport of data wirelessly from the device. It will need to be transported from the device to a computer for application within the software of the computer. There are many different wireless communication methods and it is important to design the communication across to the computer so that it is extremely fast, as to keep the input latency from the movement of the device and movement within the program low.

1.4 Interfacing with the Computer

It will also be another engineering problem to interface with the computer. It is important that the device can be made functional and therefore the information it generates must be interfaced with a computer in order to achieve interaction with the computer.

1.5 Boundaries

This project does not include the need to research and develop for an ergonomic experience with the device. Although steps may be taken to improve the usability of the device throughout the project this is only for the purposes of demonstration and the developer of the device need not spend a vast amount of time designing for ergonomics.

Where application programming interfaces (API) and similar programming repositories are available they may be used. It is not necessary if avoidable for the developer to design code that can be found in publicly available resources. Some examples of this may be API's to interface HCI components with modern desktop operating systems including Windows 10 or Mac OS, or example code resources for the embedded system board the device will use.

2 Background Theory

2.1 3D Human Computer Interaction

Human computer interaction refers to the ability for a person to give input or take output from a computer. The most common examples of this today are with keyboards and mice used with generic desktop computers, but the term includes all types of human interaction with computers such as speech recognition, gesture control, and even simple button presses on a mobile device.

A 3D human computer interface refers to a system which through a user could perform 3Dimensional movements and these movements would be interpreted by the computer and used as an input.

One of the first major consumer devices that may be considered a 3D HCI would be the Nintendo Game Console the “Wii” and its wireless controller. We can see from this devices patent that was granted in the US on 09/09/2008, that the controller uses acceleration data collected before and after a predetermined amount of time to calculate the relative motion of the controller across two axis. [1] This is a relatively simple implementation of detection 3d motion about an axis, however the design of this project will likely include tracking of the device across a distance as well as its rotational position within a space.

The “Wii” remoter also had a method of tracking across distances, this design as determined in another patent assigned to Ninendo Co., shows the method they chose to achieve this was with an image processing device upon the device which recorded information from a light bar emitting a signal from two points to determine the devices position relative to these signals. [2] This design would mean that the embedded device this project will be based upon would need to be capable of collecting image data and perform processing on in order to determine the devices position. This may or may not be feasible depending on the components and development time available to the project.

Another possible design as described by Iason Oikonomidis, Nikolaos Kyriazis, and Antonis A. Argyros in their design for 3D Tracking of Hand Articulations, [3] Is to have an image sensing apparatus connected to the computer that process imagery to determine position in space. Although Oikonomidis et al. use a rather complex setup including a Microsoft Kinect sensor, an array of distance sensors connected to the computer may be a viable solution for this project.

2.2 RTOS Embedded Systems

One of the most significant engineering challenges with this project is the use of a real time operating system. Real time operating systems can be categories into two types. As described by Andrew S. Tanenbaum in “Modern Operating Systems”, a real time operating system can be a hard real-time system; a system when all deadlines must be hit, and timing must pre-determined. While a soft real-time system may generally meet its timing schedule, but it is okay for the system to sometimes not meet its timing specification. [4] For this project while hard-real time would be preferable for performance it should not be a problem if operations are to be delayed sometimes, so therefore a soft real-time OS will suffice.

One product line of RTOS devices is the TI-CC26xx. It was recommended by the project supervisor that for this project the family of TI SimpleLink devices be used. The reason for this is his previous experience with the devices has shown them to be very capable at both real-time operation with their implementation of TI-RTOS, and strong wireless communication capabilities. These systems are available in “launchpad” variants which include the SOC as well as a breakout board with built in debugging capabilities making it a very comfortable unit to develop upon. The TI-RTOS ecosystem includes many capabilities to make it suitable for this project. It includes a real-time kernel, as well as middleware such as device drivers and

APIs. [5] These resources will be very useful in improving efficiency and providing good performance when developing the device.

2.3 Wireless communication

For our device to be functional it will need to be able to wirelessly transfer data to the computer. The method of transportation will have a major impact on the responsiveness of the device. As this device will be uses with the computer it is communicating with at all times, the communication will therefore only ever be over a short distance. There are many new communication methods being standardised for short range communication wirelessly. These personal area networks (PANs) can be very fast and low powered ways of sending data point to point. Some of these methods of wireless communication are described within an article by Cheolhee Park and Theodore S. Rappaport. [6] One such technology is ZigBee, a communication channel that is extremely low powered. This will allow for the project device to only require a small battery capacity for longer usage. The family of SimpleLink devices described earlier are capable of this method of communication as well as Wi-Fi, Bluetooth, as well as others described on the ti website. [7]

2.5 Interfacing with Computer OS

For this project it will be important for the device to be capable of communicating with a computer, so research must be done to investigate how this can be accomplished. There are several ways this can be complete. The device could be designed to have its own windows or mac driver as a mouse input. There are publicly available resources to help in the development of such drivers. Microsoft offers an API to allow development of Human interface devices (HIDs). [8]

Although the device will be used as a 3D mouse, the mouse input for computers is limited to a single plane of movement. It may therefore be important to also program the mouse to act as a gamepad, as the windows platform allows gamepads to have multiple analogue inputs. We can see all possible input classes included in the API on the windows dev website. [9]

Another option is to use another device already programmable as a HID input for windows as a way to translate the 3D HCI data into a usable format for a windows machine. The Arduino device is capable of this due to libraries wrote for the device to have it act as a mouse or keyboard device. [10]

3 Proposed Design

3.1 Initial Design

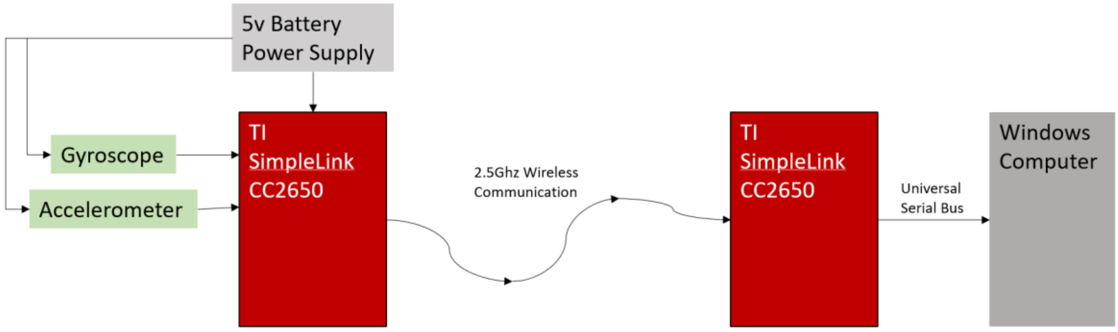

Figure 1 First Design

This is the initial design. It is extremely basic and will only record rotational data as an input. The data is then sent wirelessly to another SimpleLink device which will translate this information as mouse input data for the windows machine.

3.2 Second Design

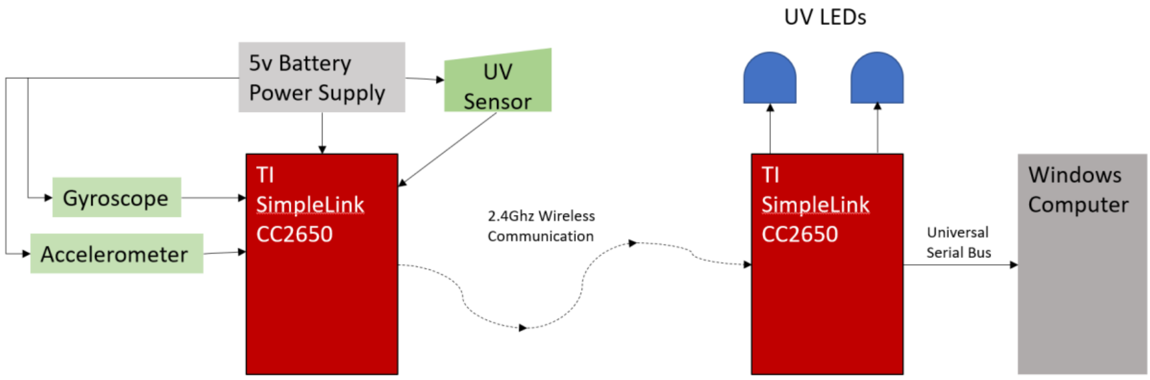

Figure 2 Second Design

This design builds on the previous design to add functionality. This design uses Ultraviolet LEDS a known distance apart as signals that can be used to determine tracking of the wireless device. This will allow the device to send both rotational and positional data. However, this means that the board will now have to either process or send the UV data. This may take a lot of bandwidth which may saturate some of the lower powered wireless communication methods and be difficult for the board to process in real-time along with the rotational data.

3.3 Third Design

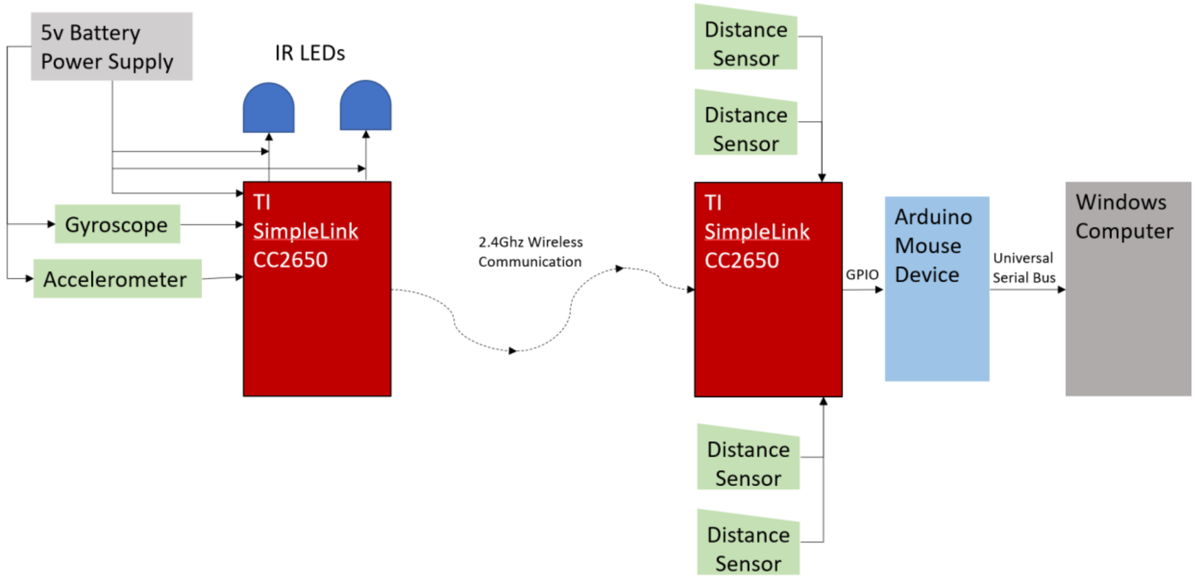

Figure 3 Third Design

This design again builds on the previous, this time however the wireless device now has infrared LEDs incorporated into the device and the other board implements an array of distance sensors, this will off load some of the processing from the wireless board to lower the power and processing that needs to be done by the wireless board. This design also incorporates an Arduino device as a translator for mouse commands. This allows the use of the Arduino mouse and gamepad libraries in order to simplify the programming of the system (this change can also be made to the previous two designs if need be).

3.4 Fourth Design

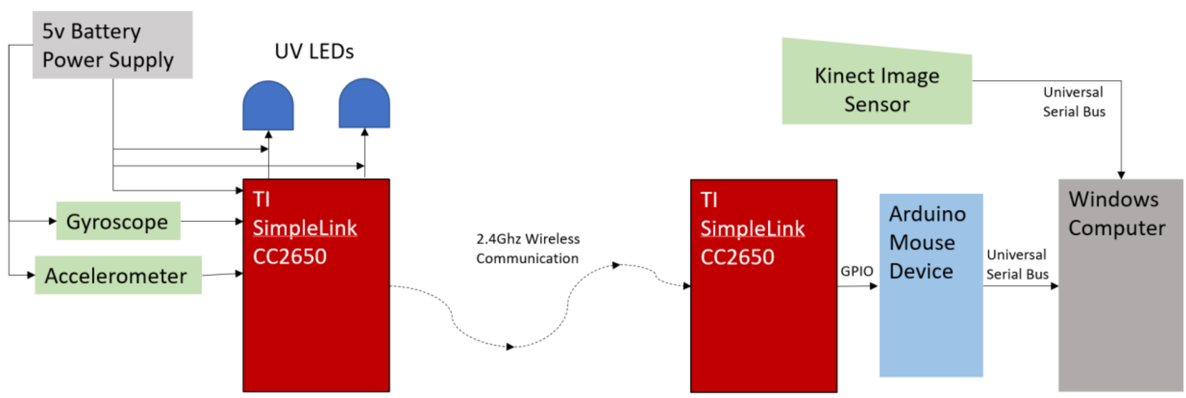

Figure 4 Fourth Design

This design again builds on the previous one, replacing the UV sensor with a Kinect image sensor, this design may drastically improve tracking capabilities however the system will now need to interface with the windows machine through two separate interfaces. This will increase the complexity of both the programming and usability of the device.

Out of these four possible designs the one the project will begin based on is design number three. Although elements from the other designs may be incorporated further into the projects life cycle. This design allows for a relatively simple tracking system that should not hurt the responsiveness of the system and not add too strongly to its complexity.

4. Work Schedule / Plan

As this project has many different elements that must all be implemented to have a functional device the organisation of work so that one element of design is not overlooked is important. To accomplish this the work should be split into sections for each element and timetabled accordingly.

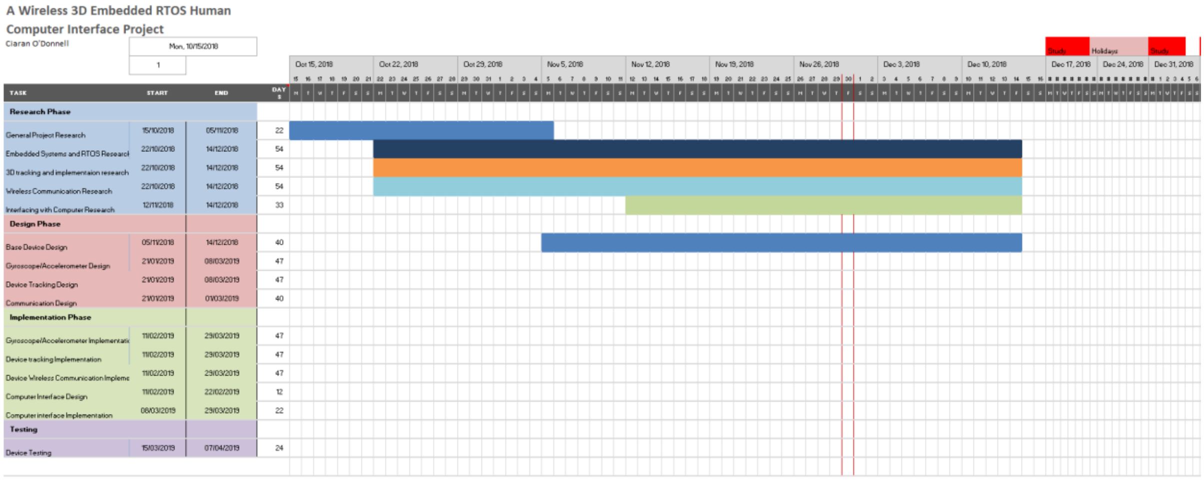

Figure 5 Schedule Oct to Dec

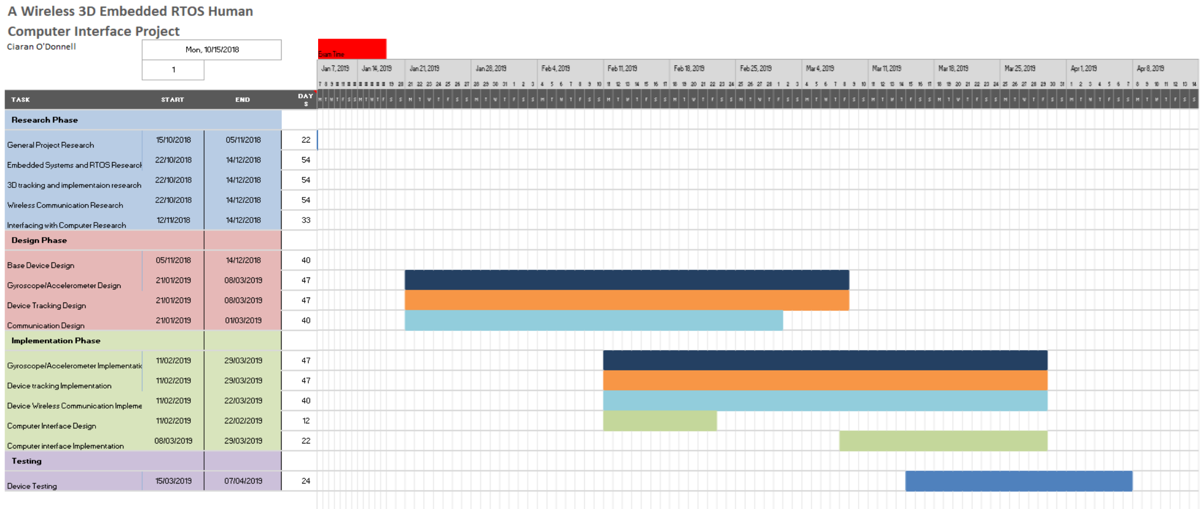

Figure 6 Schedule Dec to April

In the design of the schedule it was estimated that some phases of the project will take longer than others, and consideration was taken that some phases cannot commence until work has been completed on others, one such example is the interfacing of the device with the computer. If the device is not sending relevant data to the computer interfacing this data with the computer will be counterproductive and therefore work on interfacing with the computer will begin at a later date than the implementation of the rest of the device.

It will be important to have a complete design for the device by the 1st week in February as to begin the implementation phase. This deadline will ensure enough time to have a complete implementation of the device by the 1st of April.

It is also important that the device is capable of interpreting and recording usable data by the first week of march so that work can begin on interfacing this data with the computer.

The testing of the device will begin the second week in March, this is to ensure that a sufficient amount of testing can be complete and recorded within the project report before its deadline week 11 or 12(8th/12th of April), for this reason testing should be complete by at the latest the 1st of April.

Of course, with all of these deadlines is preferably to allow more time for later elements but if this schedule is adhered to the project should attain satisfying accomplishment.

As can be seen based on the proposed Gannt chart a lot of elements of the project will need to be worked on in tandem with one another. Organizing the work on a weekly basis is therefore very important to ensure all areas of the work is complete. So, to do this a weekly work schedule was also designed for when elements must be complete in tandem.

Project Weekly Schedule

TIME	SUNDAY	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY
9:00 AM			Device Tracking	Embedded System Board	Device Tracking	Computer Interface	
10:00 AM		IN CLASS					
11:00 AM							
12:00 PM							
1:00 PM			Wireless Communication	IN CLASS	Open Period	Open Period	
2:00 PM							
3:00 PM		Embedded System Board					
4:00 PM			Computer Interface		Wireless Communication		
5:00 PM							
6:00 PM							
7:00 PM							

Figure 7 Weekly Timetable

This is the proposed work schedule for when many elements of the project must work in tandem. Care was taken to include open periods in case additional time must be spent on a particular component. This schedule is intended as a guide to help the work remain on schedule, however it important to note that during these stages it will often be the case that these tasks will overlap, and work will often be complete in other areas during each period. However, if this schedule is used as a base no element of the project will be overlooked.

References

[1] K. Sato, “Motion determining apparatus and storage medium having motion determining program stored thereon”. US Patent US 7424388 B2, 9 September 2008.

[2] K. Ohta and K. shi, “Image processing apparatus and storage medium storing image processing program”. US Patent US 2007/0211027 A1, 13 September 2007.

[3] O. Iason, K. Nikolaos and A. A. Antonis, “Efficient Model-based 3D Tracking of,” in *British Machine Vision Confrenece*, Dundee, 2011.

[4] A. S. Tanenbaum, Modern Operating Systems, London: Pearson, 2015.

[5] Texas Instruments, “www.ti.com,” June 2016. [Online]. Available: http://www.ti.com/lit/ug/spruhd4m/spruhd4m.pdf. [Accessed 2 December 2018].

[6] P. Cheolhee and S. R. Theodore, “Short-Range Wireless Communications for Next-Generation Networks: UWB, 60 GHz Millimeter-Wave WPAN, and ZigBee,” *IEEE Wireless Communications,* vol. 14, no. 4, pp. 70-78, 2007.

[7] Texas Instruments, “TI SimpleLink MCU Platform,” Texas Instruments, 2018. [Online]. Available: http://www.ti.com/wireless-connectivity/simplelink-solutions/overview/overview.html. [Accessed 2 December 2018].

[8] Microsoft, “doc.microsoft.com,” Microsoft, 20 April 2017. [Online]. Available: https://docs.microsoft.com/en-us/windows-hardware/drivers/hid/. [Accessed 2 December 2018].

[9] Microsoft, “doc.microsoft.com,” Microsoft, 04 April 2017. [Online]. Available: https://docs.microsoft.com/en-us/windows-hardware/drivers/hid/hid-clients-supported-in-windows. [Accessed 2 December 2018].

[10] Arduino, “Arduino Refrence Mouse,” Arduino, 30 August 2018. [Online]. Available: https://www.arduino.cc/reference/en/language/functions/usb/mouse/. [Accessed 2 December 2018].