Making a Wireless Controller

Final Project Design Report

Group Members:
Hector Ramos

Project Director: Dr. David G. Daut

May 14, 2014
# Table of Contents

Abstract  

i. Design Project Overview  
   a. System Design – Final Version  
   b. System Implementation  
   c. System Performance  
   d. System Design Iterations  

ii. Technical Specifications  

iii. Final Project Summary  
   a. System Design – Final Version  
   b. System Implementation  
   c. System Performance  
   d. System Design Iterations  

iv. Task List  

v. Design Project Details  
   a. PlayStation 2 Hardwired Sub-system  
      i. Theoretical Considerations  
      ii. Design Procedure  
      iii. Observed and Measured Results  
   b. 900 MHz Transceiver Sub-system  
      i. Theoretical Considerations  
      ii. Design Procedure  
      iii. Observed and Measured Results  

vi. Sub-system Integration Considerations  

vii. Economic Considerations  
   a. Cost Analysis – Prototype  
   b. Cost analysis – Final Version  

viii. Manufacturability  

ix. Marketability  

x. Individual Team Member Discussions  
   a. Overview Discussion of the Project  
   b. Detailed Discussion of Pertinent Sub-systems  

xi. Conclusion  

xii. References  

xiii. Appendices  
   a. Appendix 1: List of Equipment  
   b. Appendix 2: Simulations and Program Code  
   c. Appendix 3: Datasheets
Abstract:

In this capstone design project, a PlayStation 2 wired controller is attempted to be made wireless via a controller hub for the system transmitting the data on RF channels wirelessly to the PlayStation 2 console. The controller data lines are observed and measured, and a FDD RF system is proposed. Ultimately however, due to purchased transceivers underperforming in terms of advertised data rates and the high data rates of the data lines in the PlayStation 2 system the system could not be successfully created.

1) Design Project Overview:

The project “Making a Wireless Controller” is about taking a PlayStation 2 video gaming controller, originally hardwired, and making it be able to function with the PlayStation 2 gaming console wirelessly. To achieve this, a docking station is created via breadboards for any PlayStation 2 controller; the dock for the controller is at the transmitter end’s breadboard, and the receiver end’s breadboard is stationed at the console. Although a cursory analysis of the project would make it seem like a rather simple affair, practical application in RF design poses a suitable challenge.

The project requires an understanding of the serial duplexing the PlayStation 2 console uses to communicate with the controller. The system uses 9 wires, each wire that communicates data only communicates in one way --- from the console to the controller, or vice versa. To implement a wireless system, each wire must be properly identified and the data rates quantified. There are a total of 5 wires that must be transmitted wirelessly, 2 of which are controller to the console, 3 vice versa. The other 4 wires are various sources of power that will be generated locally for the controller end.

The wireless communications method used for the project is FM modulation in the 900MHz band. The system decided on the in the project was to have two wireless channels, which act in a FDD (Frequency Division Duplexing) fashion to communicate the serial duplexing via encoders.
2) Technical Specifications:

PlayStation2 Console: Sony PS2 Console, SCPH – 90001 NTSC U/C

PlayStation2 Controller: Sony Analog Controller, SCPH – 10010 A

PlayStation2 Extension Cable: ezGear, 6 Foot Long PS2 Controller Extension Cable

2 RF Transceivers: Linx Technologies, TRM-900-NT; Surface Mount 900MHz FSK Transceiver supporting data rates of up to 300kbps

2 RF Antennas: Linx Technologies, ANT-916-WRT-UFL; UFL connector Antenna supporting a frequency range of 900-930 MHz.

SOIC to DIP SMT Adapter: Karlsson Robotics; 1.27mm pitch, 10.16mm body 44pin adapter for a surface mount chip to a ProtoBoard interface

ProtoBoard: EIC-104-1

Battery: 12V Battery, Powers the controller end system via multiple voltage dividers

Lab Materials:

Oscilloscopes: Tektronix, DPO4032 Digital Phosphor Oscilloscope / Teledyne Lecroy, HDO 6054

Power Supply: Tektronix, P503A Dual Power Supply

Multimeter: Agilent, 34401A 6(½)

Soldering Iron: Weller, WES51
3) Final Project Summary:

3.1) System Design – Final Version

In order to implement the wireless controller, two channels must be created for the PlayStation2 controller and console -- one to transmit data from the console to the controller, and another channel to receive data from the controller to the console. In the proposed design, the two channels are frequency divided in a FDD scheme, with a total of 4 transceivers to successfully transmit the data.

![Diagram of PS2 Controller and 12V Supply](image)

Figure 3.1: Controller-Side System

Figure 3.1 shows the controller-end side of the proposed system: With a wireless controller, a power source must be provided for the controller and receivers. Using a 12V power supply and voltage divider networks make it possible for a 12V battery to power both the controller and the transceivers used for transmitting the information wirelessly. The transmitter and receiver in Figure 3.1 are frequency divided as well to serve two different channels.
Figure 3.2: Console-Side System

Similar to Figure 3.1’s controller-side system, Figure 3.2 shows the console-side system: since the console is powered locally, no external power sources need be supplied to the console end of the system. The front-end of the console end side of the receiver also has two transceivers frequency divided, for transmitting and receiving the data across two channels for the system.

The extension cable block featured in Figures 3.1 and 3.2 notes that an extension cable cut in half with the individual wires within tinned to a Protoboard was implemented. This is necessary to transmit the and receive the data from the transceivers, as the original PlayStation 2 controller cables are not readily compatible with chipsets and protoboard interfaces.

It is important to note that most transceivers out on the market currently are not capable of supporting the data rates required by the PlayStation2 system — the data of each line must be encoded and decoded at their respective lines, as to provide enough of a compression of the data to allow multiple lines to be transmitted on one transceiver system of the two FDD channels. This is also an economically conscious design choice, as it is expensive to implement numerous transceivers within an electronic.

Unfortunately however, while the theory and proposed system is capable of providing a wireless system for a wired PlayStation2 controller, due to hardaware limitations and transceiver underperformance only one wire was properly implemented wirelessly. This however shows proof of concept, and given more transceivers and time to create microcontrollers capable of encoding/coding the information before transmission the system would be able to properly be implemented.
3.2) System Implementation

Before transmitting the data wirelessly, it is essential the data lines used to transmit the information between the PlayStation2 controller and console be thoroughly understood. An invaluable asset to this project is the “Interfacing a PS2 (PlayStation 2) Controller” guide from Curious Inventor [1]; it provides a good wealth of information behind the serial duplexing system the console uses to communicate with the controller. As shown below by Figure 3.3, the controller and console are connected by a series of 9 lines, 5 of which are pertinent for data, the other 4 being either being a power source or ground. The data lines all communicate digitally, with square wave signals being sent over 5 bytes.

Figure 3.3: PlayStation 2 Serial Wire Interface
As stated earlier, only 5 of the 9 data lines show in Figure 3.3 transfer data between the controller and the console. Each of these wires can only communicate in one direction: either from the console to the controller, or from the controller to the console. Each of the 5 lines communicate data simultaneously together in a serial duplexing fashion, so that the controller and console can both transmit and receive the necessary data simultaneously using multiple one-way data lines.

With the serial duplexing of the PlayStation 2 system, the ‘direction’ of each data line’s communication must be known before a wireless system can be properly implemented. Once the serial duplexing directions are known, one frequency channel for the RF transceiver can be dedicated for all communications from the console to the controller, and another frequency channel can be dedicated for all communications from the controller to the console. It is important to note that in order to keep multiple data lines on one frequency channel, encoding and compression is necessary, with decompression and decoding techniques used at the receiver end. This will introduce certain levels of lag to the system, but are necessary to practically apply the wireless system to the controller.

![Figure 3.4: Serial Duplexing Data Line Direction](image)

The directionality of the data lines given in Figure 3.3 are shown in Figure 3.4. 3 Lines, ‘Command’, ‘Attention’, and ‘Clock’ are all sent from the console to the controller. 2 lines are sent from the controller to the console: ‘Data’ and ‘Acknowledge’. With two frequency divided RF channels, one frequency can transmit encoded Command, Attention and Clock lines from the console to the controller, while another frequency transmits Acknowledge and Data lines from the controller to the console. Figures 3.1 and 3.2 show both a transmitter and receiver at both ends of the system; this is to implement the frequency duplexing of the serially duplexed information within the PlayStation 2.
To provide the frequency divided RF channels, the TRM-900-NT transceiver from Linx Technologies was employed. This transceiver is capable of both transmitting and receiving data at 900MHz, and the schematic of the pin layout used is shown below in Figure 3.5 below.

![TRM-900-NT Transceiver Pin Layout](image)

Figure 3.5: TRM-900-NT Transceiver Pin Layout

The T/R Select line is responsible for triggering the transceiver to either transmit or receive, and the DATA_IN and DATA_OUT lines handle the respective information the RF transceiver will be responsible for transmitting or receiving.
3.3) System Performance

Unfortunately, due to the RF transceivers performance not being 300kbps as advertised, the wireless system’s performance greatly suffered. The performance tested in the lab of the transceivers was ~210kbps, as illustrated by Figure 3.6 and 3.7 below:

Figure 3.6: RF Received Signal of a 200kHz Square Wave

As shown above, when a 200kHz square wave with a high of 7V and a low of 100mV is sent wirelessly using the receiver architecture followed in Figure 3.5’s TRM-900-NT transceiver, the square wave is properly received, albeit scaled --- the high value is scaled down to 1.7V. This shows that the transceiver and performs well and accurately given this criteria, though there is a time delay introduced by the RF transmission of the signal.
As seen in Figure 3.7, the threshold for the RF transmitter and receiver is 220kHz. At this frequency, errors in the received square wave begin to emerge; in the photo above, the second to last square wave holds high, despite the transmitted signal having been sent low. Unfortunately, this error was beyond the project’s control --- the RF transceivers did not perform as advertised, despite having been configured correctly.

Depending on the data line transmitted, the received signal varies. Lower data rate lines within the PlayStation 2 data lines are properly received with a time delay, while higher data rate lines experience heavy distortion, as shown by Figures 3.8 and 3.9. Figure 3.8 shows that with a lower data line within the serial duplexing system, the Data line in the case, the received signal is accurate, although time delayed. In Figure 3.9, the Clock line was wirelessly transmitted and as can be seen, the high data rate of the signal affects the received signal – there is data lost in the RF transmission.

Out of all the lines individually tested and sent wirelessly while the other lines were hard wired, only one line successfully transmitted and had the system still perform normally: the acknowledge line.
Figure 3.8: Data Line Transmitted/Received Signal

Figure 3.9: Clock Line Transmitted/Received Signal
3.4) System Design Iterations

The first prototype to the wireless system was a Frequency Duplexed system where every line shown in Figure 3.4 would have its own RF Frequency channel. This proved idea turned out to be extremely expensive, and a micro-controller used to encode/decode the signals and other compression techniques were immediately considered upon acknowledgement of the fees implied. The prototype system’s design is featured in Figure 3.10.

Figure 3.10: Prototype FDD Wireless System

The second iteration of the project included a locally generated clock on the controller side of the system. The locally generated clock was meant to replicate the console’s clock signal accurately, so that the locally generated clock could be synchronized with the received clock – an unnecessary procedure, but the thought process behind it was that the clock signal was the most important signal within the system, so extra care should be taken in making sure it could be received accurately. As Figure 3.11 shows, the locally generated clock would be added to the transceiver of the controller side of the system to be synchronized with the RF received signal in this prototype system.
In the second prototype, the locally generated clock was intended to be identical to the clock signal being sent by the console side of the system to the controller; however the main issue is that the clock would then need to be synchronized with the received signal, otherwise all of the received and transmitted signals would be out of sync and unusable. This is an oversight, since if the received signal is going to be used for the clock regardless, there is little reason to create it locally and then synchronize the signal, assuming there isn’t any issues with the transmission. However if there was a problem with the transmission of the signal, it is likely that the other channels suffer as well, and the system overall would be unusable.

To locally generate the clock, a 1MHz square wave oscillator, frequency divided by two D flip-flops was to be employed. Each D flip-flop reduces the frequency of the original square wave by half, so the end signal would be 250kHz, which is the frequency of the clock wave generated by the console. The theory behind the D-type Flip-Flop is shown in Figure 3.12:
Finally, the locally generated clock prototype suffered from one more major flaw: the clock generated from the console is more complex than what was anticipated. The clock is separated into 5 bytes forming a packet, with each packet being separated by a heavy time delay. This characteristic proved difficult to replicate accurately, and the locally generated clock was ultimately abandoned. The clock’s characteristics 5 byte characteristic can be seen in Figure 3.13:

![Figure 3.13: Clock, Data and Command Lines [Top to Bottom]](image)

In Figure 3.13, the clock signal is the top waveform, followed by the data and command lines. As can be seen, the clock signal is necessary for the synchronization of the data and command lines, all syncing up to the 5 bytes of the clock signal.
4) Task List:

**Week of 2/10:** Researched PlayStation 2 transmission methods

**Week of 2/17 + 2/24:** Developed a way to successfully observe the PlayStation 2 wire Characteristics

**Week of 3/10 + 2/24:** Observe and test each line’s data, Reconfirm Curious Inventor’s report data

**Week of 3/17:** Clock Further Researched, Locally Generated Clock considered

**Week of 3/24:** Locally Generated Clock abandoned, RF Transceiver system researched

**Week of 3/31:** Directionality of the Serial Duplexing System measured, Data Rates questioned

**Week of 4/7:** Data Rates found for the system, RF Transceivers purchased

**Week of 4/14:** Research into possible RF encoding/decoding techniques for a FDD system

**Week of 4/21:** Encoding/Decoding abandoned due to too much workload, RF Transceivers attained --- found to be surface mounts

**Week of 4/28:** RF Transceivers successfully mounted to an adapter for the ProtoBoard, Lines tested wirelessly failed

**Week of 5/5:** Data rate found to be lower than advertised, compromise made and one data line was found to successfully wirelessly transmit the signal with the rest hard-wired
5) Design Project Details:

5.1) PlayStation 2 Hardwired Sub-system:

5.1.1) Theoretical Considerations

The first subsystem, the PlayStation 2 Communication System, is the most crucial part of the project and must be properly observed, identified and measured before the wireless system is constructed. As identified in Curious Inventor's guide [1], the PlayStation 2 communicates in a serial duplexing fashion, shown in Figure 3.4.

Serial communication is the act of sending data, one bit at a time sequentially in a format over a communication channel. In the PlayStation 2 system, all communication along every data line is serial, with square waves being used to digitally transmit the data and bytes. With the directionality expressed in Figure 3.4, the system enacts a Full Serial Duplexing system: the system is able to send and receive data at the same time. An example of this can be seen below, in Figure 5.1.1:

![Figure 5.1.1: Serial Duplexing of Clock, Data, and Command Lines](image)

In Figure 5.1.1, the Clock, Data and Command lines can be seen, from top to bottom. The full serial duplexing system communicates using the clock line to sync up all the lines, where the Data line can communicate data from the controller to the console, and the console can communicate data from the console to the controller on the Command line.
The methodology for the full duplex serial communication is shown above in table 5.1.1, and can be observed in Figure 5.1.1. It is important to note that most lines used in the system idle on the high value, and are active low. This idle-high active-low characteristic can be seen along all lines used in Figure 4.1.1. In the figure, a packet can be seen: each byte is separated by ~30uS, and each packet of 5 bytes is separated in the serial duplexing system by ~250uS.

The lines used for the serial duplexing system are described below in detail. It is important to note that while Curious Inventor’s guide [1] was used to help reverse engineer the lines, good Engineering practice was performed and each data line was tested locally to re-affirm the data given. Each line unless otherwise specified idles high and performs active lows.

1. **Brown - Data**: Controller -> PlayStation. This line transmits all relevant controller information for the console, such as which buttons are pressed at each time. Held buttons are detected by the high refresh rate of each packet. ~250uS.

2. **Orange - Command**: PlayStation -> Controller. This line transmits data from the console to the controller; this line typically idles on the waveform shown on 4.1.1, except to send data for the vibration motors and other relevant console to controller data.

3. **Grey - Vibration Motors Power**: A power source of 6-9V is provided from the console.

4. **Black – Ground**: Provides the ground for the controller, provided from the console.

5. **Red - Power**: A power source of 3-5V is provided from the console; in this system, 3V is supplied.

6. **Yellow - Attention**: This line gets the attention of the controller for the console. This line idles high before each packet by pulling itself low before each packet. At the end of each packet, it idles high again.

7. **Blue - Clock**: Sends a ~255kHz square wave signal across every packet. It is vital that this waveform be extremely accurate.

8. **White – Unknown**: This line is unused by the console and controller used in this project.

9. **Green - Acknowledge**: This line shows that the controller acknowledged the attention line sent from the console. The line idles high until a byte is passed, on which is active lows and then immediately idles high again. Acknowledges each byte passed.

<table>
<thead>
<tr>
<th>Communication Line</th>
<th>Data Rate</th>
<th>Data High</th>
<th>Data Low</th>
<th>Byte Spacing</th>
<th>Packet Spacing</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>250kbps</td>
<td>3V to .6V</td>
<td>.1V to 0V</td>
<td>30.3 uS</td>
<td>254.1uS</td>
</tr>
</tbody>
</table>

Table 5.1.1: Serial Communication Values
5.1.2) Design Procedure

In order to properly observe and quantify the data lines used in the PlayStation 2 system first the lines must be made available for the lab equipment. By taking the controller extension cable and cutting it in half, each line shown in Figure 3.3 can be seen. Taking each line and properly tinning each it does not provide adequate access to the ProtoBoard since the wires are too thin. However tinning and soldering each line to a multi-pin connector for the ProtoBoard is adequate and can be seen below:

![Figure 5.1.2: Controller Cable Wired to ProtoBoard](image)

Each wire in Figure 3.3 can be seen to in Figure 5.1.2 above. The orange wiring between the cables in Figure 5.1.2 is so that both ends of the cable can be faced towards each other on the ProtoBoard easily, leaving the system hard-wired and allowing the characteristics of each line to be observed within the system by measuring each line individually via oscilloscope.
5.1.3) Observed and Measured Results

With the wiring allowed by the tinning shown in Figure 5.1.2, the characteristics of each line are easily observed and replicated, using Curious Inventor’s guide [1] as a reference. As shown by Figure 5.1.3, Curious Inventor shows that button presses of Up and R2 on the controller are transmitted on the Data Line on bit 5 of byte 4, and bit 2 of byte 5:

![Figure 5.1.3: Curious Inventor’s Clock/Data Line Example](image)

This data from Curious Inventor proved to be accurate, as it was successfully replicated in the lab, shown in Figure 5.1.4.

![Figure 5.1.4: Clock, Data and Command Lines, with Up and R2 Pressed](image)
The attention and acknowledge lines were also successfully observed and their characteristics measured, as shown by Figure 5.1.5:

![Figure 5.1.5: Clock, Data, Attention and Acknowledge Lines](image)

The characteristics observed in both Figures 5.1.4 and 5.1.5 both are with the entire system hard-wired; the observed characteristics are understood and follow Curious Inventor’s guide, so the PlayStation 2 system can now be wirelessly sent.
5.2) 900MHz Transceiver Sub-System:

5.2.1) Theoretical Considerations:

In order to fully implement the wireless controller, two transceivers systems with encoders and decoders in a FDD system will have to be implemented. This is due to the high data rates of every line in the PlayStation 2 system; compression techniques must be implemented to encode and decode the lines before they are transmitted onto the RF channels. A Figure for this subsystem is shown below:

![Figure 5.2.1: Frequency-Division Duplexing with Encoders and Decoders](image)

As seen in Figure 5.2.1, by using two Transceiver systems simultaneously frequency divided, it is possible to fully implement a wireless system for the PlayStation 2 Controller system. It is important to note that at the data rate of ~255 kbps for each line, the original concept of using a transceiver system for each line is not economically sound or viable. However with micro-controllers, an encoding/decoding scheme shown above can be successfully followed, allowing for the system to be successfully implemented in a wireless format. Although the process of encoding and decoding multiple lines will introduce a time delay to the final system, with the data rates and hard-wired signals the PlayStation 2 uses to communicate Figure 5.2.1 introduces a practical application to transmit the system wirelessly in a typical RF environment.

Unfortunately however, the RF Transceiver and its wireless transmission of the data lines for the PlayStation 2 system suffer from aliasing issues. This is due to the Nyquist frequency advertised not being observed within the system. The advertised frequency was 300kHz, however in the lab only 220 kbps were properly observed.

\[ f_s \geq 2 \cdot f_c \]

Figure 5.2.2: Nyquist Sampling Formula

The equation in Figure 5.2.1 states that the sampling rate must be at least twice the desired frequency observed: In this case, the advertised Nyquist frequency was 300kHz, however the frequency was found to be 220kHz in the lab.
5.2.2) Design Procedure

The TRM-900-NT transceiver from Linx Technologies was implemented for the wireless transmission of lines in the PlayStation 2 system. However there was originally a bit of an oversight: the TRM-900-NT is a surface mount chip, and the system was being developed on a ProtoBoard. This made an adapter necessary --- however the TRM-900-NT chip is a non-standard chip; its body size is particularly large. Due to this large body size, no standard adapter could be bought for the surface mounts, and some innovation had to be introduced. By cutting a surface mount adapter with the proper length and pitch in half, the body width issue was successfully avoided, and the 44 pin TRM-900-NT was successfully installed onto the ProtoBoard.

Figure 5.2.3: TRM-900-NT with adapters installed onto the ProtoBoard

As seen in Figure 5.2.3, the surface mount adapters can be seen to be cut in half to adapt to the large body size of the TRM-900-NT. Two ANT-916-WRT-UFL antennas from Linx Technologies can also be seen to be installed onto each transceiver’s UFL port, completing the wireless system. The schematic used for each TRM-900-NT transceiver can be seen in Figure 3.5, with the left transceiver being the transmitter, and the right transceiver being the receiver.
As seen below in Figure 5.2.4, this wireless transceiver setup was then tested with a mostly hard-wired system for the PlayStation 2 controller cable. By testing each line wirelessly while the rest of the cables were left hard-wired, the performance of the wireless system can be tested and estimated despite not having micro-controllers or multiple transceivers to implement the system fully.

Figure 5.2.4: TRM-900-NT integrated with the PlayStation 2 Hard-Wired System
5.2.3) Observed and Measured Results

As shown by Figures 3.8 and 3.9, any waveform over 220kHz introduce data corruption and aliasing by the system. This is important to note, since any high data line will by immediately corrupted by this system. By using the circuit layout introduced in Figure 5.2.4, each of the 5 data lines was tested wirelessly sequentially, with the rest of the 4 data lines being left hard-wired. For each of the following Figures, the top waveform is the transmitted waveform, while the bottom waveform is the received waveform.

Figure 5.2.5: Clock Line Wireless Transmission

As can be seen by Figure 5.2.5, the heavy aliasing of the clock signal made the controller unusable, so the clock line could not be transmitted wirelessly successfully.
Figure 5.2.6: Data Line Wireless Transmission

Figure 5.2.7: Command Line Wireless Transmission
As seen by Figures 5.2.6, 5.2.7, and 5.2.8, the wireless transmission of lower data lines get received rather accurately. Unfortunately however, none of these lines transmitted wirelessly on their own made the PlayStation 2 controller functional --- this is because of the time delay the wireless stage introduces into the system. Looking closer at Figures 5.2.6/7/8, it can be seen that each signal is properly received, however the signals are indeed slightly out of phase due to a time delay, and that time delay makes the system unable to function. It is estimated that if the same time delay were introduced throughout the entire PlayStation 2 controller system the wireless transmission would indeed be successful.

It is important to note that out of the 5 data lines one line was capable of being transmitted wirelessly while the other lines were hard-wired and have the PlayStation 2 controller be functional: the Acknowledge line shown below in Figure 5.2.10.
Figure 5.2.9: Acknowledge Line Wired Transmission

Figure 5.2.10: Acknowledge Line Wireless Transmission
The Acknowledge line was the only one out of 5 in the current design that, when wirelessly transmitted, still upheld a functional PlayStation 2 controller. The reason behind this is that the acknowledge signal is the only data line that can suffer a time delay and still not affect the overall system. As spoken of in 5.1.1, the Acknowledge line idles high and pulls low at the end of each received byte, as can be seen in Figure 5.2.9. Since the acknowledge line must wait until after the bytes are received, it seems that a small time delay with the input is adequate enough to result in a functional system.

However, looking at Figure 5.2.10, it is evident that introducing the Acknowledge line to the RF transceiver greatly modifies its behavior. In Figure 5.2.10 the top waveform is the clock, the second waveform is the transmitted Acknowledge signal, and the bottom waveform is the received Acknowledge signal. By taking a further look at the Figure, it can be seen that the voltage of the signal received and transmitted change wildly --- the received signal is highly attenuated and is susceptible to noise, and the transmitted signal shows a similar attenuated characteristic. Regardless, the line does pull low and high in between each byte sent, and this seems to prove efficient for the PlayStation 2 controller system.
6) Sub-system Integration Considerations:

As spoken of earlier, there was a series of difficulties installing the PlayStation 2 Controller and the RF Transceivers to the ProtoBoard. In the case of the PlayStation 2 Controller, a row of ProtoBoard pins tinned and soldered to the PlayStation 2 Controller’s lines enabled the system to be installed onto the ProtoBoard, as seen in Figure 5.1.2.

The RF Transceiver’s fix was more complex, however --- since the transceiver purchased was a surface mount and a ProtoBoard was being used for the circuit design and not a PCB, an adapter was necessary. Soldering wires to each of the 44 pins manually was considered, but due to it being very prone to error it was ultimately abandoned in favor of the 44pin adapter. The adapters on the market however did not accommodate for the chip’s abnormal size, though the pitch and pin number was easily found. To get around this problem, a 44pin adapter was purchased with normal width, and then cut in half and soldered together with the chip on top, as shown in Figure 5.2.4.

Further Sub-system integration concerns for future or unfinished work on the project would be properly dividing a 12V Battery to power supply both the controller’s buttons and analog supplies which are about 5V and 9V respectively, as well as the RF transceiver which is 5V. Integrating a switch system onto the transceiver as well to help it consume less power is a further integration concern for the power supply.

The final Sub-system integration consideration would be making the system readily portable, which is highly desired of any wireless controller, especially one for gaming. Having the wireless controller system be robust is an important requirement, as well as it being small and having room for the antennas is desirable.
7) Economic Considerations:

7.1) Cost Analysis – Prototype:

A) PlayStation 2 Controller: $20
B) Extension Cable: $5
C) Battery: $6
D) 4 RF 900MHz NT Series Transceiver: $140
E) 4 916MHz WRT Series UFL Antenna: $20
F) 4 SMT to DIP Adapters: $60
G) 4 Micro-Controllers: $60

**TOTAL: $356**

It is important to note that this cost analysis adds the cost for a battery, 2 additional transceivers and antennae, and 4 micro-controllers. Shipping costs have been included.

7.2) Cost Analysis – Final Version:

A) PlayStation 2 Controller: $10
B) Extension Cables: $1
C) Battery: $3
D) 2 RF 900MHz NT Series Transceiver: $50
E) 2 916MHz WRT Series UFL Antenna: $5
F) 2 Micro-Controllers: $30
G) Packaging: $10
H) Detailing: $5

**TOTAL: $114**

Due to economies of scale, the original component costs have been scaled down. A large mention is also that before the mass production version, it is necessary to implement a one channel system for all relevant communications to scale the costs greatly and make it possible for this product to be affordable.
8) Manufacturability:

This product is relatively easy to manufacture due to the surface mounted transceivers; this makes it very easy to create this product with a PCB. With PCBs being used to manufacture this product, wires can be added and extension cable ports added at both ends of the system without much cost or difficulty. It is important to note, however, that due to the 12V power source needed for the components in this project the battery might cause manufacturing issues in terms of the size of the overall product.

9) Marketability:

Unfortunately since this product is not only for an outdated system but $100 even in the large scale production estimations of the system this product is ultimately unmarketable. PlayStation 3 controllers, a now outdated generation in gaming technology compared to the PlayStation 4, costs $60 dollars with more functionality and features, such as USB connections.

However if this product were to be made during the 2nd generation of PlayStations, the PlayStation 2, this product would still be rather costly; PlayStation 2’s at the time cost roughly $200-$300, and controllers were generally sold for $30 each. Accounting for inflation and the decreasing cost for similar performance over the years, this product would have cost even more to manufacture at the dawn of the PlayStation 2, making it far too expensive for the consumer market.

Comparing to a product called “PS2 Wireless Controller” by GameStop, the price offered currently for a new wireless controller is $20, about 1/5 the cost of the projected industry-scale market cost for the wireless controller of this project.
10) Individual Discussions:

10.1) Overview Discussion of the Project

The project to make a wired PlayStation 2 controller wireless was originally thought to be a very simple one at the proposal of the system; this proved to be false. Reverse engineering a full serial duplexing communication system employed by the PlayStation 2 added the first level of difficulty to the wireless controller system. The high data rates introduced by each line introduced the is a complication that unfortunately goes mostly unsolved by the end of the capstone --- this partly due to the transceivers purchased not performing as advertised, but also due to the fact that no micro-controllers employed in a encoding/decoding fashion were employed, making the system overall difficult to transmit practically over RF channels due to the high data rates of each line. This is highly regrettable, as a fully functional wireless controller was desired.

That’s not to say that the project was a complete failure, however – more communications techniques were learned and perfected during the process such as serial duplexing and FDD techniques. Reverse engineering the controller’s data lines also was interesting and rewarding when the data was successfully replicated. Unfortunately due to working alone on the project there were periods of time that the capstone had to go untouched due to large workloads in other courses. This taught an important lesson that an assistant or group member can be extremely helpful, even if they help in small quantities to be able to detect possible errors in the system sooner rather than later that can be tacked and handled quickly.

Finally, it is appalling to see that after cost evaluations, considering numerous RF and data compression techniques and searching for the best parts at reasonable prices to see a wireless controller sell on the market for a fraction of the price proposed in industry scale production. It is appalling, yet intriguing to see that throughout the numerous difficulties in the systems of the project a cost effective solution to the same problem was employed and marketed at such a reasonable price.
10.2) Detailed Discussion of Pertinent Sub-systems

One of the originally considered systems for the wireless controller system employed having a separate RF transmitter and receiver for each data line. This architecture proved to be extremely naïve, as it underestimated the data rates of each line, and greatly underestimated the cost of transceivers on the market. It was quickly abandoned in favor of two RF transceiver systems frequency divided, as shown in Figures 5.2.1. This architecture proves to be more cost efficient and practical, and employs a encoder/decoder at the each end of each system – this is to minimize and compress multiple lines over one RF channel.

As shown in Figure 5.1.2 and discussed earlier, in order to get reverse engineer the data lines of the PlayStation 2 controller system each line had to be tinned on to a row of pins and then plugged into a ProtoBoard. After this process was completed, reverse engineering the data lines and reconfirming Curious Inventor’s [1] results was the primary concern at the stage of the project. Each data line was found to have a characteristic of 5 bytes per packet, and each packet separated by the same amount of time. It is highly important to note that each line is synchronized and each data line has the same number of bytes per packet.

A locally generated clock was considered for the system, as shown in Figure 3.11; this idea was ultimately abandoned when the process of replicating the time delays between the bytes and especially the packets proved to be difficult as that complicated the square wave signal considerably; the received clock signal was determined to be sufficient in light of these facts.

Once the RF Transceivers were purchased, it was quickly found out that they were surface mounts, and this set back the project about a week as the special answer to the adapter size was implemented, as shown in Figure 5.2.3. Upon having the full transceiver available, the aliasing issues began arising, as shown in Figure 5.2.5. It was at this point of the capstone design that the focus of the project became just transmitting one signal across the RF channel successfully for proof of concept.

Unfortunately transmitting one line wirelessly proved to be more difficult than anticipated: lower data rate lines would be received with a time delay and the PlayStation 2 controller would not function, and the higher data rate lines suffered from heavy aliasing. The only data line that would function wirelessly while the rest of the data lines were wired was the Acknowledge line, as shown in Figures 5.2.9 and 5.2.10. However it is important to note that while this proves the wireless transmission of one data line can function, the characteristic of the Acknowledge line wildly changes when transmitted wirelessly. The line begins to idle low and active high, the opposite of the line hard-wired. It is also highly susceptible to noise, as shown in Figure 5.2.10.
11) Conclusion

In conclusion, making a wireless controller was far more difficult than was originally anticipated for a gaming platform. There are multiple lines with high data rates that must communicate simultaneously in a serial duplexing fashion, and this requires a well-designed system to implement cost effectively and have it be robust – which is possible, as shown by a wireless controller being sold for $20 on the market.

The capstone design project was also very useful for learning engineering practice and compromises. Multiple times throughout the project a system had to be implemented in a unique way not readily available on the market and some innovative techniques with what was given at hand were employed. Skills like this are invaluable, as many prototyping reports report similar work-around during the beginning phases of projects.

Unfortunately the system could not be fully implemented, though a wireless channel was successfully formed across the Acknowledge line. This warrants further research, as the characteristic of the line changes greatly when applied to an RF channel; this is likely due to an input or output impedance mismatch, however.

Further work in the project to make a functioning wireless controller would include researching the reason for distortion across the acknowledge line, using micro-controllers for encoding and decoding various lines onto one channel with the data compressed and finally implementing a second RF transceiver system on a separate frequency channel.
References


