Courseware and Curriculum Development For a Wireless Electronics Class

by

Ariel Rodriguez

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Masters of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

January 2007
[February 2007]

Author ....
Department of Electrical Engineering and Computer Science
January 16, 2007

Certified by ....
Steven B. Leeb
Professor, MIT
Thesis Supervisor

Certified by ....
Robert W. Cox
Assistant Professor, UNC Charlotte
Thesis Supervisor

Accepted by ....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Courseware and Curriculum Development For a Wireless Electronics Class

by

Ariel Rodriguez

Submitted to the Department of Electrical Engineering and Computer Science
on January 16, 2007, in partial fulfillment of the
requirements for the degree of
Masters of Engineering in Electrical Engineering and Computer Science

Abstract

The design of basic wireless building blocks (such as oscillators, amplifiers, and modulation circuits) and modern encoding techniques such as CDMA (Code Division Multiple Access) are in high demand by employers of recent graduates. This thesis sets forth a lesson plan and laboratory kit design to be used in the development of a new class that teaches these wireless system design techniques. Such a class will help students gain both the theoretical and practical experience required of them in today's industry.

Thesis Supervisor: Steven B. Leeb
Title: Professor, MIT

Thesis Supervisor: Robert W. Cox
Title: Assistant Professor, UNC Charlotte

Acknowledgments

Special thanks go out to my advisors: Professor Steven Leeb who taught me what it means to be a professional engineer and Professor Rob Cox whose knowledge and guidance made this thesis possible. I'd also like to thank Jim Paris for his sage-like advice on microcontroller programming. My sincerest gratitude goes out to the Grainger Foundation and Cambridge-MIT Institute for their financial support. Finally, to all those who believed in me: my family and friends; I could not have completed this without your unending love and support thank you so very much.

Contents

List of Figures

List of Tables

Listings

Chapter 1

Introduction

Wireless systems are among the most widely used technologies by todays engineers and circuit designers. As a result, employers are starting to demand more experience in these areas from new recruits. This demand is not narrow in scope. Commercial wireless technologies alone encompass a broad range of circuit theory and manufacturing techniques.

Given this wide-ranging employer demand, students need to be introduced to wireless technologies early on; at the undergraduate level. As part of their education they need to be introduced to several basic wireless electronic building blocks (i.e. amplifiers, oscillators, resonant circuits) as well as modern data transmission techniques such as those used in CDMA communication systems.

Not only must they be introduced to the fundamental concepts underlying these technologies but they also need experience applying them in real-world situations. An undergraduate laboratory class that surveys several wireless electronics techniques has the potential to successfully provide students with both the theoretical and practical experience they need. This thesis aims to develop the equipment (laboratory kit, test equipment, etc.) and preliminary lesson plan for such a class.

Chapter 2 presents a new laboratory kit designed to be the centerpiece of the class. Labs in subsequent chapters are all based around the design and contents of the laboratory kit. The systems built into the kit feature a "permanent topology" approach to engineering and construction. This approach contrasts sharply to most solderless breadboard-based kits, but gives students the opportunity to explore more systems within one semester.

Chapters 3 through 7 described each of the five laboratory assignments recommended to be completed in one semester. Chapters 3 and 4 have students create the fundamental transmission and reception systems for the rest of the semester. These include tuned amplifiers, AM and FM modulators, and a superheterodyne receiver system. Chapter 5 details several exploratory exercises in phase-locked loops, culminating in an FM demodulator circuit. Chapter 6 deals with direct sequence spread spectrum techniques such as code division multiple access methods of transmission using microcontrollers. Finally, chapter 7 delves into topics related to electromagnetics including transmission line problems, antenna patterns, and microwave generation.

Chapter 8 summarizes the intended goals of the exercises listed in the previous chapters. Additionally, the lab kit is designed with a modicum of modularity that allows for expansion of the class material. Possible areas for expansion are also discussed in this chapter.

Chapter 2

The Laboratory Kit

The laboratory kit for the course is the main platform for all of the assignments students will complete throughout the semester. It will contain all of the circuits discussed in subsequent chapters of this document. The kit will follow the design of other course laboratory kits at MIT, taking the shape of a clam shell as shown in Figure 2-1. The kit will open to reveal its two main communication units: a transmission module for broadcasting wireless signals and a reception module for receiving and demodulating signals. The two units will be separable, each having its own power supply which will allow them to work independently from each other. With these two independently operating units, students can create several different wireless communication scenarios with one or more laboratory kits.

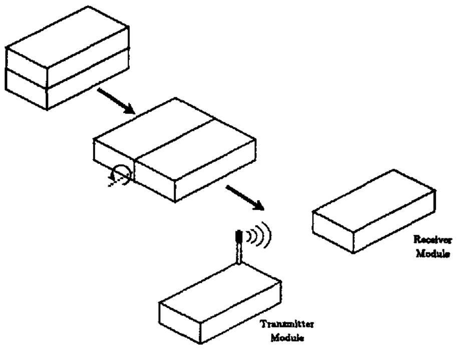


Figure 2-1: Conceptual drawing of the laboratory kit "clam shell" design. The kit will open up and separate into its two main units: a transmission module and a reception module.

Moreover, students will not rely on solderless breadboard for the majority of the circuits constructed for each module. For example, the broadcast framework built into the transmission module will be built using the idea of "permanent topologies". Each major circuit to be studied will already have its topology etched into its respective modules printed circuit board. Students will study the specifics of each circuits design, but will not be required to construct the final version of each circuit to be used for the rest of the semester. Instead, all the components salient to each circuits design (e.g. resistors setting bias points, filter components, etc.) will be replaced by sockets as shown in Figure 2-2. Students will be required to select the appropriate components and place them in these sockets, but will not be required to wire the circuits together. This permanent topology design approach will minimize the time students spend in construction, without sacrificing the important design stages of building each circuit (namely selecting component values according to design criteria).

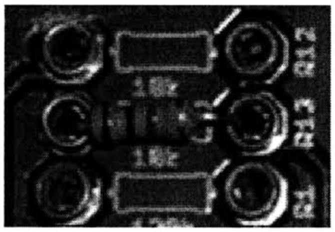


Figure 2-2: Photograph of the component sockets to be used in the laboratory kit "permanent topology" design. Components critical to system design will not be soldered in directly. Instead, they will be selected by students and placed in sockets as shown.

2.1 Transmission Module

The transmission module will be the basis for all the laboratory assignments broadcast and modulation components. The module will consist of 6 separate sections: a power supply, broadcast amplifier, modulation system, CDMA subsystem, solderless breadboard, and extension interconnects. These sections will be arranged inside the kit according to the transmission-module diagram shown in Figure 2-3.

The independent power supply unit will allow the transmission module to work independently from the reception module. The broadcast amplifier will be developed as part of the first laboratory assignment in Section 3.1. The modulation system will also be developed in the first assignment, described in Section 3.2. The CDMA subsystem will be discussed in Chapter 6. A small section of solderless breadboard will also be included for rapid prototyping. Lastly, power and signal interconnects will be included as part of an "extension modules" section so that new functionality can be added to the lab kit in the future. The

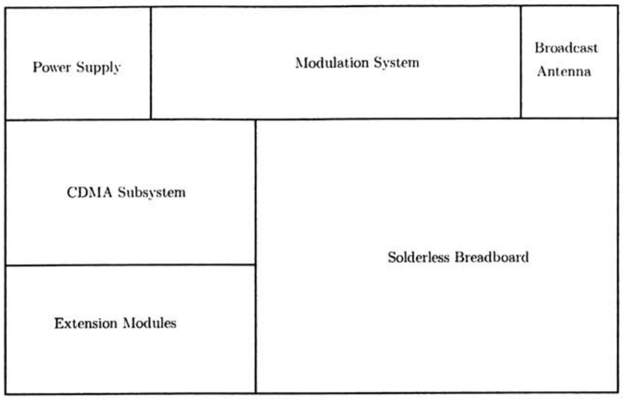


Figure 2-3: Diagram describing the contents of the laboratory kit transmission module.

PCB circuit schematic of the transmission module design is included in Appendix B.1. The modules PCB layout is shown in Figure 2-4 with a larger version included in Appendix C.1. All parts necessary for its construction are included in Appendix D.1.

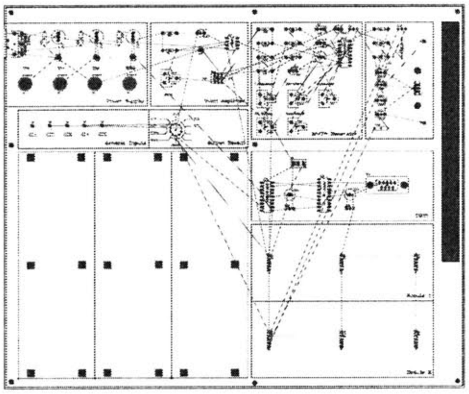


Figure 2-4: Eagle CAD layout of the laboratory kit transmission module.

2.2 Reception Module

The reception module will be the basis for all the laboratory assignments signal reception and demodulation components. This module will be more densely packed than the transmission module. The reception module will be made up of 8 separate sections: a power supply, the superheterodyne receiver, receiving antenna assembly, demodulator section, phase-locked loop section, CDMA subsystem, solderless breadboard, and an extension modules section similar to the one included in the transmission module. These sections will be arranged as shown in Figure 2-5.

Like the transmission module, the reception module will have its own power supply

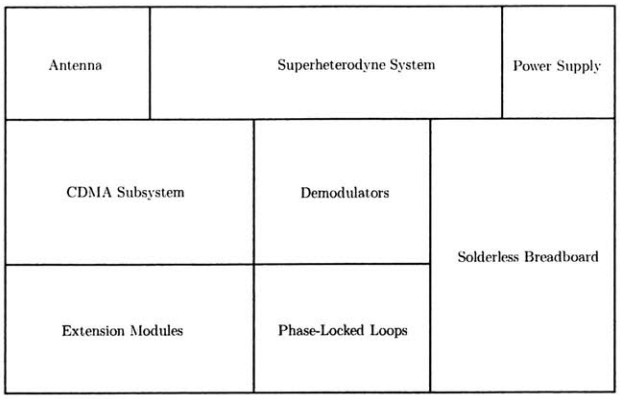


Figure 2-5: Diagram describing the contents of the laboratory kit reception module.

allowing students to use it independently. The superheterodyne receiver will be the main reception component of the kit, its construction and design will be the subject of the second laboratory assignment, described in Chapter 4. The receiving antenna assembly will also be discussed in Chapter 4. The included demodulator circuits are the subject of several different laboratory assignments, each is tailored to a different communication method (e.g. AM, FM, CDMA, etc.). The two included phase-locked loop circuits will be part of design exercises in Chapter 5. The CDMA receiving subsystem is the subject of several exercises in Chapter 6. Solderless breadboard and extension interconnects are provided for prototyping and new functionality just like the transmission module. All of these functional areas are included in the reception module PCB whose schematic is in Appendix B.2. The modules PCB layout is shown in Figure 2-6 with a larger version included in Appendix C.2. All parts necessary for construction are in Appendix D.2.

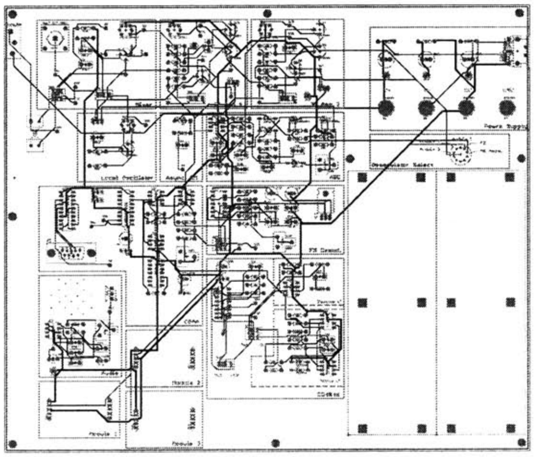


Figure 2-6: Eagle CAD layout of the laboratory kit reception module.

Chapter 3

Lab 1: Resonant Circuits, Amplifiers, and AM/FM Transmission

This chapter presents the first set of laboratory exercises to be taught in the wireless course. Students will design and construct the main transmission framework to be used throughout the semester, including circuits for amplitude and frequency modulation and a power amplifier for broadcasting signals. The block diagram for this framework is shown in Figure 3-1. Students begin by working with passive energy storage elements, that is, inductors and capacitors. Combining these elements motivates the discussion of resonance and the theme of frequency-based analysis of signals and circuits.

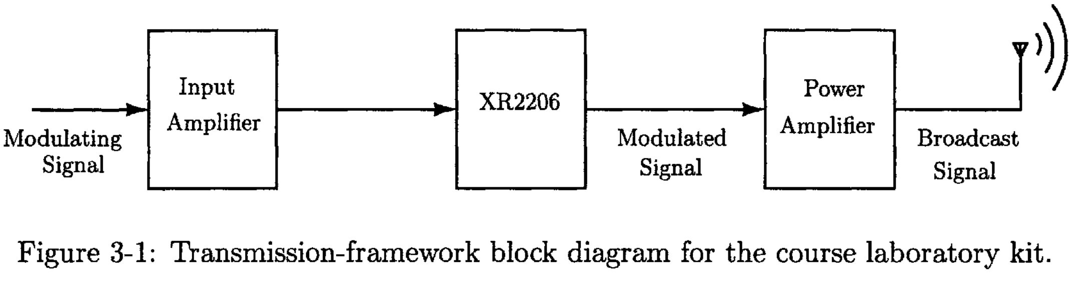


Figure 3-1: Transmission-framework block diagram for the course laboratory kit.

Electromagnetic induction and resonance are key elements to the design of antennas for broadcast. Using resonance, circuits can be built around antennas that will select for and amplify certain frequencies. Frequency selection and amplification are key for creating the broadcast component of the framework. The broadcast antenna to be used will require an amplifier to drive it. Students will use an RF linear amplifier designed to create large voltage swings at its output to increase the broadcast range.

After building the RF amplifier, students will broadcast their own signals from their circuits constructed on the laboratory kit. Two modulation schemes will be used for communication between the laboratory kit and a hand-held radio. First, amplitude modulation will be implemented using the XR2206 monolithic function generator. Then, this same process is repeated using frequency modulation which will also be implemented using the same XR2206 chip in a different configuration.

3.1 The Linear RF Power Amplifier Circuit

A large amount of power must be delivered to the laboratory kit transmission antenna to achieve long broadcast distances. To do this, students will need to construct a power amplifier capable of driving an RF broadcast signal. The power amplifier will be of a linear design, creating an amplified replica of its input at its output. It will have two stages: a buffer and an RF power amplifier. The buffer stage will allow other devices with low output power characteristics to drive the RF power amplifier stage. The RF power stage itself will be responsible for creating a suitable high-power output for driving the broadcast antenna/LC tank. The overall amplifier, including both stages, is shown in Figure 3-2.

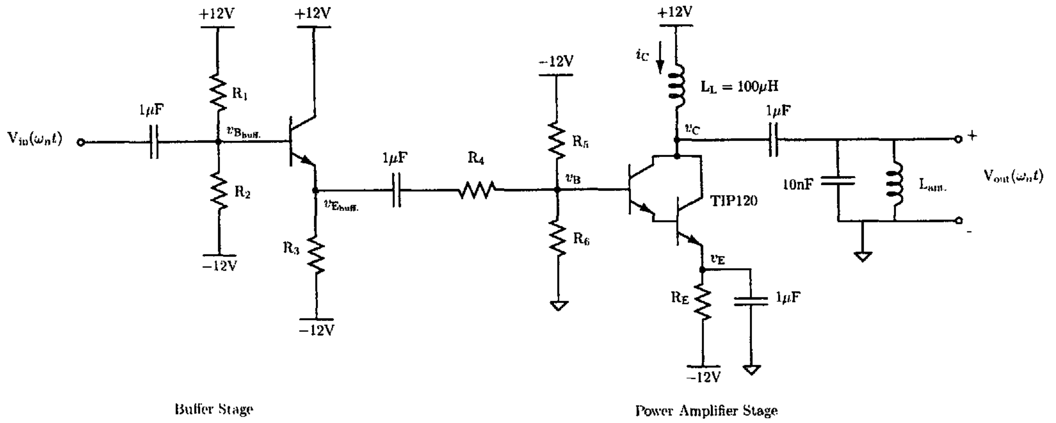


Figure 3-2: RF Linear Power Amplifier used for broadcast signal transmission.

3.1.1 Buffer Input Stage

The first stage in the overall amplifier circuit of Figure 3-2 is a buffer that uses a 2N3904 NPN bipolar transistor. The buffer circuit is an emitter-follower, unity-gain amplifier. It will be used to ensure that circuits generating signals to be broadcast can drive the input of the power amplifier stage.

Students will begin their design of the buffer circuit by choosing an emitter resistor value that will create a desired emitter current of 9mA. The voltage across the emitter resistor, $\mathrm{R_3}$, is

$$\mathrm{V_{R_3} = I_E R_3} = \left(9 \times 10^{-3}\right) \mathrm{R_3}$$

Rearranging this equation and substituting for the potential difference across the resistor,

$$\mathrm{I_E} = 9 \times 10^{-3} = \frac{\mathrm{V_{R_3}}}{\mathrm{R_3}} = \frac{v_{\mathrm{E_{buff.}}} -^{-} 12\mathrm{V}}{\mathrm{R_3}} = \frac{v_{\mathrm{E_{buff.}}} + 12\mathrm{V}}{\mathrm{R_3}}$$

The voltage at the transistors emitter will be 0.7 V below the voltage at its base according to the 2N3904s specified base-emitter voltage drop [1]. Therefore, the emitter-current formula becomes

$$\mathrm{I_E} = 9 \times 10^{-3} = \frac{v_{\mathrm{B_{buff.}}} - 0.7\mathrm{V} + 12\mathrm{V}}{\mathrm{R_3}}$$

Assuming a negligible base current, the voltage at the base will be entirely determined by the biasing network shown in Figure 3-2. The base voltage is therefore

$$v_{\mathrm{B_{buff.}}} = \frac{\mathrm{R_2}}{\mathrm{R_1 + R_2}}(12\mathrm{V} -^{-} 12\mathrm{V}) = \frac{\mathrm{R_2}}{\mathrm{R_1 + R_2}}(24\mathrm{V})$$

Using two 12kΩ resistors for $\mathrm{R_1}$ and $\mathrm{R_2}$, the current through the bias network is

$$\mathrm{I_{bias}} = \frac{24}{\mathrm{R_1 + R_2}} = \frac{24}{24\mathrm{k}\Omega} = 1\mathrm{mA}$$

which is much greater than the base current

$$i_{\mathrm{B}} = \frac{9\mathrm{mA}}{\beta} \approx 90\mu\mathrm{A}$$

for a conservative estimate of 100 for the current gain β. Thus, this base-current calculation verifies the previous assumption of negligible base current affecting the bias voltage.

Moreover, the base voltage should be set mid-way between the collector-emitter voltage. With this bias point, the buffer output signal can swing (at maximum) symmetrically about its supply rails. Thus, to set $v_{\mathrm{B}_{\mathrm{buff.}}}$ to 0 V (half-way between +12V and -12V), the bias network resistors must be equal. For a base voltage of 0 V, the emitter-current formula becomes

$$\mathrm{I_E} = 9 \times 10^{-3} = \frac{-0.7 + 12\mathrm{V}}{\mathrm{R_3}} = \frac{11.3}{\mathrm{R_3}}$$

Solving the formula for the necessary emitter resistance, $\mathrm{R_3}$,

$$\mathrm{R_3} = \frac{11.3}{9 \times 10^{-3}} \approx 1.2\mathrm{k}\Omega$$

With the emitter resistor value now selected the buffer circuit design will be complete.

3.1.2 Power Amplifier Stage

The power amplifier stage will use two NPN bipolar junction transistors in a Darlington pair configuration. These two transistors are packaged together as a single component called the TIP120 [2]. In a Darlington pair, the two individual transistors create an overall circuit with current gain equal to the product of the two single transistors gains [3]. Thus, the TIP120 acts as a single NPN bipolar transistor with a much higher gain than a single transistor. This high-gain property is valuable for maximizing the power delivered to an attached load, which in this case is a broadcast antenna. Higher amounts of power provided to an antenna load will translate directly into longer broadcast ranges.

Students will analyze the TIP120 Darlington pair as if it were a monolithic NPN bipolar transistor. The resistor network at the base of the TIP120 (consisting of resistors $\mathrm{R_5}$ and $\mathrm{R_6}$) will set the forward-active region bias point. In the case of the TIP120, it must be biased at -4 V to remain in its forward-active region (thus the ratio of bias resistors must be $\frac{\mathrm{R_6}}{\mathrm{R_5+R_6}} = \frac{2}{3}$). The input signal to the TIP120 power amplifier will be coupled to the bias point using a resistor, $\mathrm{R_4}$ to limit the input current from the previous stage and the 1μF capacitor from Section 3.1.1 for AC coupling.

The emitter of the Darlington pair will be tied to the -12V supply rail using a resistor,

$\mathrm{R_E}$. A 1μF capacitor to ground will also be placed in parallel with $\mathrm{R_E}$. The resistor in the parallel combination provides emitter degeneration for stability in the face of thermal runaway. Connecting $\mathrm{R_E}$ to the -12V rail also increases the voltage swing of the output to 24 V peak-to-peak. However, this resistor also limits the gain of the amplifier. In order to counteract this gain limitation, the parallel capacitor will be used. The parallel capacitor will appear as a short to high frequencies, increasing the RF gain of the circuit while allowing the $\mathrm{R_E}$ to remain for its emitter degeneration properties.

At the collector of the TIP120, a 100μH inductor will be placed. This inductor is large enough to provide an approximately constant current through the collector of the transistor. Including this constant current from the inductor $\mathrm{L_L}$, $\mathrm{I_{DC}}$, the total collector current can be modeled as:

$$i_\mathrm{C} = \mathrm{I_{DC}} + i_\mathrm{rf}\sin(\omega_n t)$$

The signal current $i_\mathrm{rf}\sin(\omega_n t)$ will be provided by the input signal (intended for broadcast) at the base. Although this small-signal model of the collector current would be considered inappropriate given the large amplitudes of the signals in question, the LC tank circuit attached to the collector output provides a measure of linearization for any distortions (thus the small-signal model will continue to be used) [4]. The constant current provided by the large inductor $\mathrm{L_L}$ will be blocked by the 0.33μF capacitor at the collector. Thus, the output voltage will be related only to the drain signal current $i_\mathrm{rf}\sin(\omega_n t)$. The LC tank will be tuned to this signal frequency ω_n, operating at resonance to create a large-amplitude sinusoid.

Moreover, the amplitude of the signal voltage at the collector of the TIP120 will swing symmetrically about the supply voltage. This property of the power amplifier appears due to the need for volt-second balance across the collector inductor $\mathrm{L_L}$. In small-signal terms, the collector inductor appears as a DC short (it does not allow a steady-state voltage to appear across it) allowing the sinusoidal output voltage to swing evenly about the collector-emitter voltage [4]. This provides sufficient output power for driving the antenna inductor to provide a broadcast signal at distances reasonable for communications across a large room. An oscilloscope capture of the power amplifier in operation is shown in Figure 3-3.

Thus, the complete RF linear amplifier circuit provides a power amplifier stage that can

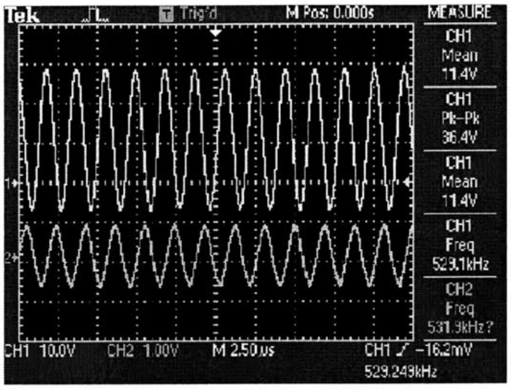


Figure 3-3: Power amplifier stage operating near the maximum output voltage. The input signal (bottom trace) is a simple sinusoid of magnitude approximately 1V peak-to-peak. The output voltage is over 36V peak-to-peak.

drive the LC antenna circuit for ranged broadcasts. The amplifier requires input currents that are perhaps too great for the output stages of other circuits generating signals to be broadcast. As a result, the RF linear amplifier also includes a buffer stage prior to the power amplifier stage to allow low output current circuits to drive the TIP120-based power amplifier. The combination of these two stages creates a suitable transmission frontend circuit for all of the courses broadcasting needs.

3.2 The XR2206 As A Modulation Platform

The data signals intended to be broadcast in later laboratory assignments vary in frequency between 100 Hz and 20 kHz. In order to accommodate this broad range of potential frequencies for broadcast the XR2206 monolithic function generator will be used. The XR2206 will modulate a single carrier frequency using the data signals created in each laboratory assignment. Using a single carrier cuts down on complexity, and the XR2206 allows for a carrier frequency that is low enough that it can easily be manipulated using breadboard-based circuits if necessary. Moreover, the XR2206 can perform various different modulation schemes, making it a convenient teaching platform on modulation techniques.

3.2.1 Amplitude Modulation

The first modulation scheme students will explore is amplitude modulation. A data signal will be used to change the amplitude of a carrier signal that is in turn broadcast using the power amplifier/resonant tank circuit from Figure 3-2. The AM system will follow the canonical DSB/WC (Dual Sideband With Carrier) configuration shown in Figure 3-4 [5]. An input modulating signal $x(t)$ is multiplied by a carrier signal $c(t)$. The resulting signal has a scaled version of $c(t)$ added to it to create the complete broadcast signal $y(t)$.

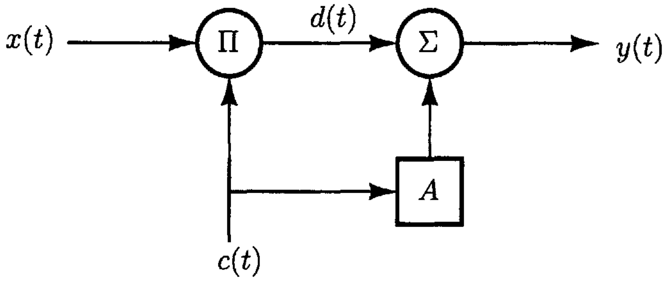


Figure 3-4: Block diagram of the canonical DSB/WC modulation system.

Furthermore, the modulating signal $x(t)$ is a sinusoid of the form $V\cos(\omega_m t)$, where V is the amplitude of the signal, operating at some frequency ω_m, known as the modulating frequency. The carrier, $c(t)$, is also a sinusoid but operates at a higher frequency ω_c (the carrier frequency) and takes the form $\cos(\omega_c t)$. The modulating signal is multiplied by the carrier to create the intermediary signal $d(t)$ resulting in

$$d(t) = x(t)c(t) = V\cos(\omega_m t)\cos(\omega_c t)$$

As a last step, a scaled replica of the carrier signal is added to $d(t)$ to create the final modulated signal $y(t)$. The carrier is added to $d(t)$ to create an offset in the amplitude of $y(t)$. This property is useful for asynchronous demodulation of the signal at the receiver, as discussed in Chapter 4 and in [5]. The final modulated signal to be broadcast is

$$y(t) = d(t) + Ac(t) = V\cos(\omega_m t)\cos(\omega_c t) + A\cos(\omega_c t) = (V\cos(\omega_m t) + A)\cos(\omega_c t)$$

Factoring out the component A, gives the following formulation of $y(t)$:

$$y(t) = A(1 + m\cos(\omega_m t))\cos(\omega_c t) \quad m = \frac{V}{A}$$

The factor m in this formulation is known as the modulation index or modulation depth. It is the maximum fraction of the carrier signal amplitude that is modulated [5]. That is, for a modulation index $m = 0.5$, the carrier signal varies 50% above or below its original amplitude. Students will be able to adjust this modulation index with the AM system they will build using the XR2206.

Therefore, to construct the AM system described above, the XR2206 will be used in its amplitude modulator configuration shown in Figure 3-5. The modulating signal $x(t)$ will be AC coupled into pin 1 of the XR2206 using a $0.33\,\mu$F capacitor. The input pin also has a DC bias placed on it by the resistor network consisting of R_5, R_6, and R_7. The ratio of the magnitude of the modulating signal to the DC bias at this pin will set the modulation index m. That is:

$$m = \frac{|V|}{A} = \frac{|V|}{\frac{\mathrm{R}_6+\mathrm{R}_7}{\mathrm{R}_5+\mathrm{R}_6+\mathrm{R}_7}12\ \mathrm{V}} = \frac{|V|(\mathrm{R}_5+\mathrm{R}_6+\mathrm{R}_7)}{12(\mathrm{R}_6+\mathrm{R}_7)}$$

Moreover, the DC bias and thus the modulation index, can be adjusted using the potentiometer R_6. However, the user must be careful to not set the bias point to be half the supply rail (in the case of Figure 3-5, 6 V). Doing so will set the XR2006 for a suppressed-carrier mode of operation which will not be covered in this lab exercise.

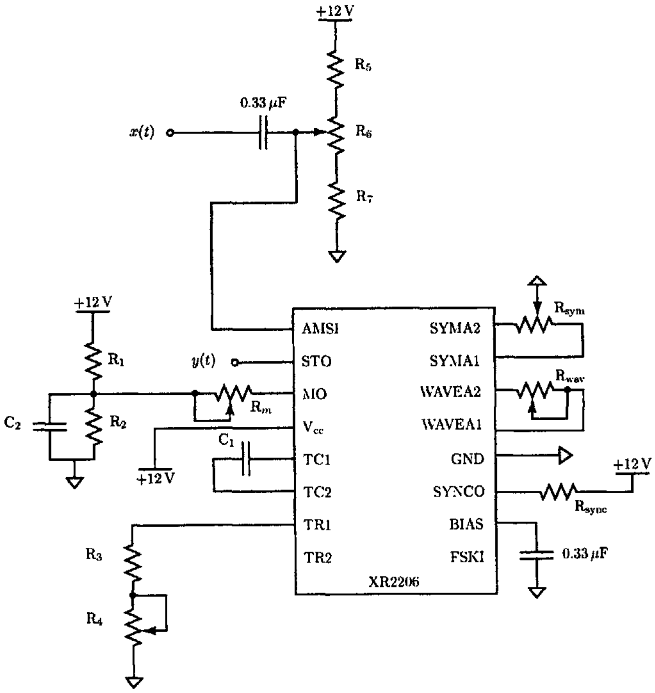


Figure 3-5: XR2206 circuit configured as an amplitude modulator.

In the circuit of Figure 3-5, the carrier frequency can be set by adjusting the potentiometer R_4, and is observable at both the output pin (pin 2, STO) and the synchronization pin (pin 11, SYNCO, which provides a square wave at the carrier frequency for reference purposes). Three other potentiometers: $\mathrm{R_{sym}}$, $\mathrm{R_{wav}}$, and R_m will be used to adjust the shape of the output signal $y(t)$. $\mathrm{R_{sym}}$ adjusts the symmetry of the output waveform. $\mathrm{R_{wav}}$ adjusts the shape of the waveform; how much it looks like a sinusoid. Finally, R_m adjusts the magnitude of the output waveform.

An example oscilloscope capture of the XR2206 working as an amplitude modulator is shown in Figure 3-6. The modulating signal (top trace) is given a DC bias (middle trace) to set the modulation index m. This signal in turn modulates the amplitude of the broadcast signal (the bottom trace).

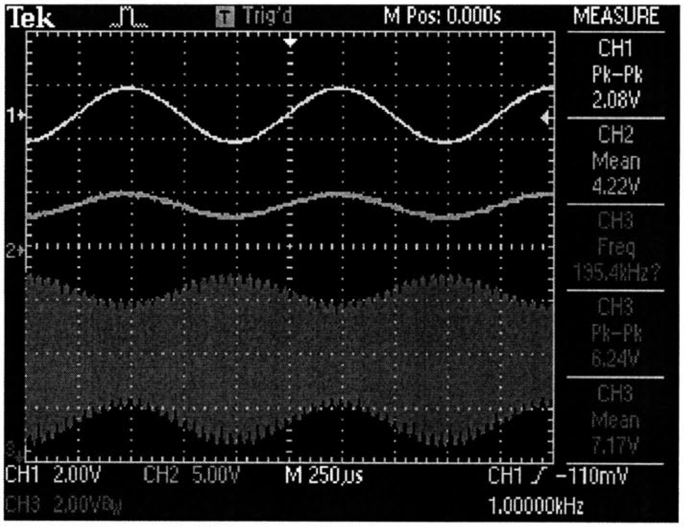


Figure 3-6: Oscilloscope trace of the XR2206 acting as an amplitude modulator with $m \approx 0.25$. The top trace is the modulating signal. The middle trace is the modulating signal with DC offset to create a modulation index of 0.25. The bottom trace is the final modulated carrier signal.

Using the XR2206 as an amplitude modulator will therefore create a circuit replica of the canonical AM system from Figure 3-4. With this circuit, students will be able to explore the properties of amplitude modulation systems including: adjusting the carrier frequency, the modulating signal, and changing the modulation index.

3.2.2 Frequency Modulation

The second and last modulation scheme students will explore is frequency modulation. In contrast to the AM system from Section 3.2.1, the modulating signal in a canonical FM system changes the frequency of the carrier. This frequency-modulated carrier signal is then broadcast using the power amplifier/resonant tank circuit of Figure 3-2. Unlike AM systems, FM systems are highly non-linear as the modulating signal $x(t)$ now changes the frequency of a broadcast signal $y(t)$ [5]. [5] describes FM system operation by expressing a carrier signal $c(t)$ in the form:

$$c(t) = A\cos(\omega_c t + \theta_c(t)) = A\cos(\theta(t))$$

In this equation, ω_c is the carrier frequency and $\theta_c(t)$ is the phase of the carrier. The overall angle of $c(t)$, $\theta(t)$, can be modulated by a function $x(t)$. For example, the phase of $c(t)$,

$$\theta_c(t) = \theta_0 + k_p x(t)$$

consists of θ_0, a constant, and $x(t)$, a scaled version of the modulating signal. This in turn results in a modulation of the overall angle $\theta(t)$ of the carrier signal. Moreover, if $x(t)$ is used to modulate the derivative of $\theta(t)$ then:

$$\frac{d\theta(t)}{dt} = \omega_c + k_f x(t)$$

which would allow the modulating signal to change the frequency of $c(t)$. Therefore, if the broadcast signal $y(t)$ were related to the carrier signal $c(t)$ by:

$$y(t) = c(t) = A\cos(\theta(t)) = A\cos\left(\int \frac{d\theta(t)}{dt} dt\right)$$

then $x(t)$ modulates $y(t)$ as follows

$$y(t) = A\cos\left(\int \omega_c dt + \int k_f x(t) dt\right) = A\cos\left(\omega_c t + \int k_f x(t) dt\right)$$

If $x(t) = V\cos(\omega_m t)$, then

$$y(t) = A\cos(\omega_c t + \frac{k_f V}{\omega_m}\sin(\omega_m t))$$

The instantaneous frequency of $y(t)$ will then be

$$\frac{d\theta(t)}{dt} = \omega_c + k_f x(t) = \omega_c + k_f V \cos(\omega_m t)$$

which will vary sinusoidally between the values of $\omega_c - k_f V$ and $\omega_c + k_f V$, making the total change in frequency,

$$\Delta\omega = \omega_c + k_f V - (\omega_c - k_f V) = k_f V$$

Rewriting $y(t)$ to include the total change in frequency leaves

$$y(t) = A\cos(\omega_c t + \frac{\Delta\omega}{\omega_m}\sin(\omega_m t))$$

The factor $\frac{\Delta\omega}{\omega_m}$ is known as the modulation index m for FM systems. Similar to the modulation index in AM systems, it is related to the portion of the total frequency that changes with $x(t)$. The case where m is small is known as *narrow-band* FM. For a narrow-band signal,

$$\begin{aligned} y(t) &= \cos(\omega_c t + m\sin(\omega_m t)) = \cos(\omega_c t)\cos(m\sin(\omega_m t)) - \sin(\omega_c t)\sin(m\sin(\omega_m t)) \\ &\approx \cos(\omega_c t) - m\sin(\omega_m t)\sin(\omega_c t) \quad \text{for } m \ll \frac{\pi}{2} \end{aligned}$$

With this approximation, the spectrum of $y(t)$ is reduced to a small band near the carrier as shown in Figure 3-7. The band is related to the modulation index m. Students will be able to, as with the previous AM system, adjust this index in-circuit with the hardware FM system.

Therefore, to construct the FM system described above, the XR2206 will be reconfigured to act as a frequency modulator. Figure 3-8 shows the circuit to be constructed by students. The modulating voltage signal $x(t)$ is placed at the timing resistor pin TR1 (pin 7) of the XR2206. This is the same pin responsible for setting the carrier signal frequency. This allows the magnitude of $x(t)$ to directly modulate the carrier frequency. The carrier frequency can be grossly adjusted (setting the centered frequency ω_c from Figure 3-7) using the potentiometer R_5.

Moreover, the amplitude of the modulating signal is directly related to the carrier fre-

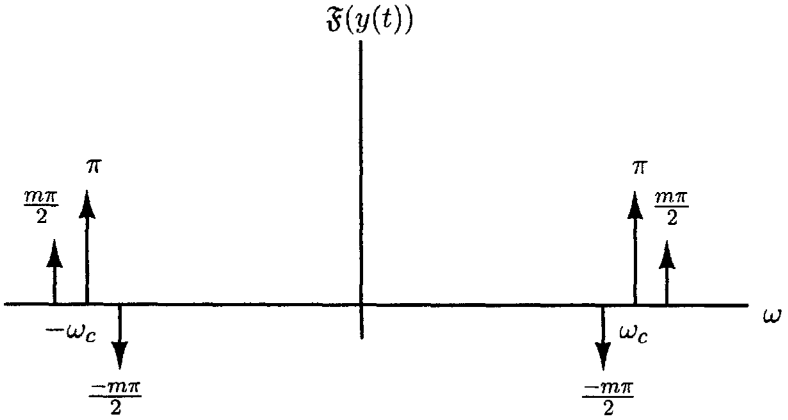


Figure 3-7: Approximate spectrum plot for a narrow-band FM signal from [5]

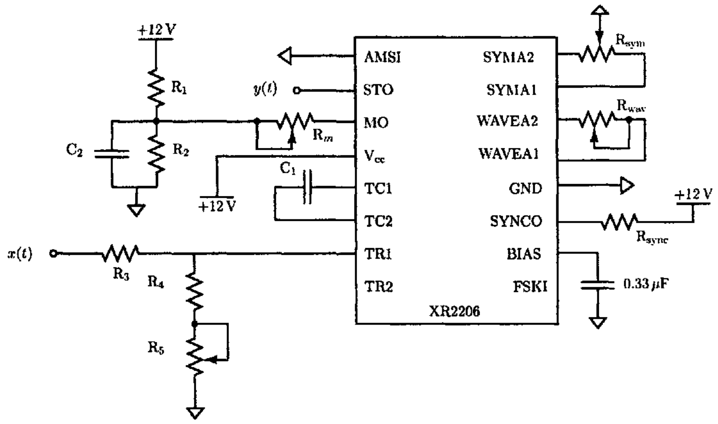


Figure 3-8: XR2206 circuit configured as a frequency modulator.

quency. The relationship between the modulating signal and the carrier determines the modulation index m. The XR2206 datasheet describes this relationship by the equation:

$$f = \frac{1}{(\mathrm{R}_4 + \mathrm{R}_5)\mathrm{C}_1}\left(1 + \frac{\mathrm{R}_4 + \mathrm{R}_5}{\mathrm{R}_3}\left(1 - \frac{\mathrm{V}_c}{3}\right)\right) \text{ Hz}$$

The variable $\mathrm{V_c}$ represents the modulating signal $x(t)$ [6]. Substituting the modulating signal into the above equation results in

$$f = \frac{1}{(\mathrm{R}_4 + \mathrm{R}_5)\mathrm{C}_1}\left(1 + \frac{\mathrm{R}_4 + \mathrm{R}_5}{\mathrm{R}_3}\left(1 - \frac{x(t)}{3}\right)\right)$$

With this equation relating the frequency f to $x(t)$, the maximum deviation in frequency

$\Delta\omega$ can be computed as follows:

$$\Delta\omega = \frac{2\pi(f_{\max} - f_{\min})}{2} \quad \text{where} \quad f_{\max} = f\Big|_{\text{maximum V}_c} \quad \text{and} \quad f_{\min} = f\Big|_{\text{minimum V}_c}$$

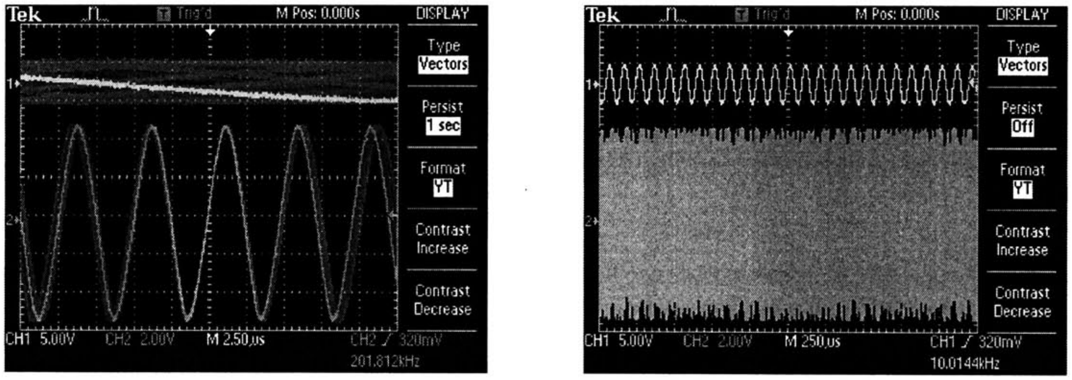
With the maximum deviation, the modulation index can be calculated using the formula:

$$m = \frac{\Delta\omega}{\omega_m}$$

Thus the modulation index is dependent on both the magnitude (through $\Delta\omega$ as described above) and the frequency of the modulating signal. Both of these parameters are adjustable by the students, allowing them to explore several modes of FM operation.

In addition to being able to adjust the carrier frequency and modulation index, the same carrier signal properties from the AM system in Section 3.2.1 can be adjusted using potentiometers R_{wav} (for the shape of the waveform), R_{sym} (for symmetry) and R_m for the magnitude.

Oscilloscope captures of the frequency modulator circuit in operation are shown in Figure 3-9. The modulating signal, displayed as the top trace on both of the captures in Figure 3-9 directly affects the frequency of the carrier signal (the bottom trace) as its magnitude increases and decreases. This effect on the carrier is easier to see in Figure 3-9(a), however the exact frequency of the modulating signal is more clearly visible in Figure 3-9(b).



(a) Carrier signal (bottom trace) exhibiting frequency modulation at approximately 200 kHz The modulating signal (top trace) is at a much slower frequency of 1 kHz

(b) The modulating signal (top trace) and the subsequent FM carrier (bottom trace). Frequency modulation is present, but difficult to see at the time division shown.

Figure 3-9: Oscilloscope captures of the XR2206 operating as a frequency modulator.

Therefore, students will be able to use the XR2206 as a frequency modulator to explore the narrow-band FM concepts described above as well as in [5]. Specifically, with the FM modulator circuit students can explore the properties of FM systems including: adjusting the carrier frequency, and how the modulating signal also affects this frequency over a small band around the carrier frequency.

Chapter 4

Lab 2: The Superheterodyne Receiver, and Asynchronous AM Demodulation

This second laboratory assignment will lead students in the construction of a reception system that will complement the transmission system from Chapter 3. A block diagram of the system is shown in Figure 4-1. The reception system will use a tunable LC tank circuit to act as a receiving antenna. A signal received by this antenna will then be processed by a mixer circuit. The mixer circuit will "mix down" the received signal frequency to an intermediate frequency of 455 kHz in addition to providing some amplification. Lastly, the mixed down signal will be amplified by two IF amplifiers. Once amplified, the signal will be processed by any of the on-board demodulation circuits or any other signal processing systems on the laboratory kit reception module. After constructing the superheterodyne reception system, students will test it by constructing an AM demodulator circuit to work in conjunction with a received AM signal from the kit transmission module. Students will add other demodulators and baseband circuits in later laboratories.

4.1 The Superheterodyne Receiver

In this first set of exercises students will construct the superheterodyne receiver. Students will start by building a receiver antenna circuit to complement the transmitter antenna

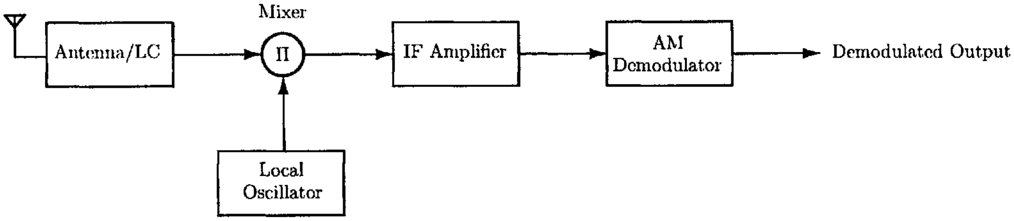


Figure 4-1: Block diagram of the superheterodyne receiver system to be constructed including asynchronous AM demodulator.

built in Chapter 3. Next, students will build the mixer circuit, a vital component in the operation of the superheterodyne that mixes signals down to the intermediate frequency. Lastly, students will build two identical IF (intermediate frequency) amplifiers to amplify the mixer output.

4.1.1 Receiving Circuit

The first component to be constructed in the superheterodyne reception system is the receiving circuit. The receiving circuit will be responsible for picking up various signals radiated to the air for broadcast from the transmission system of Chapter 3. The circuit will consist of three components: a variable capacitor, a transformer (acting as an antenna), and a biasing network for the mixer circuit. The variable capacitor and antenna will be pre-packaged, off the shelf components given to students. The output of the antenna circuit will be connected to the next stage in the superheterodyne receiver, the mixer. A schematic of the intended circuit is shown in Figure 4-2.

The receiving circuit will operate by creating an LC tank circuit (like the transmission antenna from Chapter 3) with the transformer primary winding and the variable capacitor. By changing the value of the variable capacitor, students will change the resonant frequency of the LC tank, and thus select the frequency to be received. The secondary side of the transformer will scale the amplitude of the received signal by the transformation ratio ($\frac{\mathrm{N}}{\mathrm{M}}$ in the case of Figure 4-2).

Using a transformer will allow for the easy addition of a bias network to the secondary of the transformer/antenna. At DC, the capacitor in the bias network will look like an open circuit, creating a fixed bias current, set by the 510k Ω bias resistor, at the output. This current will be the bias current for the base of the transistor in the mixer stage. For the AC signals of interest, the bias-network capacitor will act as a short to AC ground in parallel

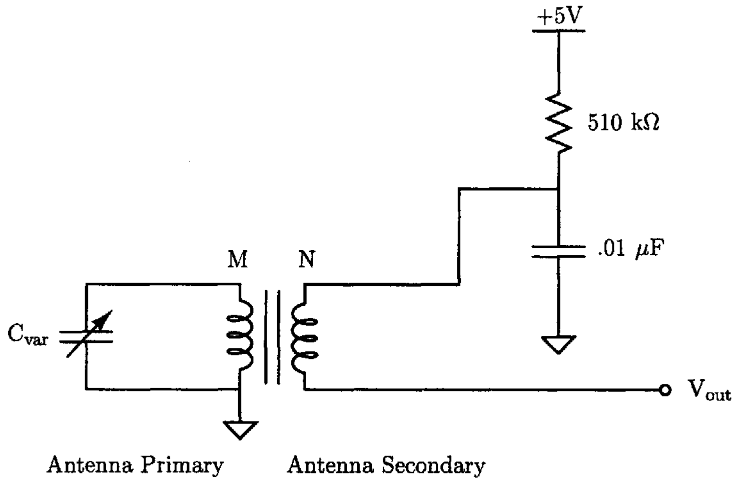


Figure 4-2: Complete reception antenna circuit. The variable capacitor allows the antenna to be tuned to a wide variety of broadcast frequencies. The biasing network at the secondary allows the antenna output to appropriately drive the mixer stage.

with the resistor, making the secondary appear as an AC source. The output of the AC source will be the received signal coupled in from the primary winding.

4.1.2 Mixer

After the antenna, the next stage of the superheterodyne receiver system will be the mixer. The mixer will change the frequency of the antenna-received signal to the lower, intermediate frequency of 455 kHz. The mixer is the most vital component of the superheterodyne system. By converting all incoming signals to the same frequency, the mixer allows all subsequent amplifiers and processing circuitry to be designed to work at a single frequency. This feature greatly reduces system complexity. The mixer will always be able to mix the received signal down to 455 kHz by using a companion circuit called the local oscillator. The local oscillator will be tied to the same tuning system as the receiving antenna, allowing it to produce a signal approximately 455 kHz above the frequency to which the antenna is tuned.

Ideally, the mixer would operate by multiplying the signal $r(t)$ from the receiving antenna and the local oscillator signal $l(t,$ set to be 455 kHz greater in frequency than $r(t)$. The multiplication of these two signals in time is the same as convolution in frequency. This convolution would create a signal that is exactly at 455 kHz as well as one at approximately twice the frequency of $r(t)$ as shown in Figure 4-3. The frequency components other than 455 kHz would be filtered out.

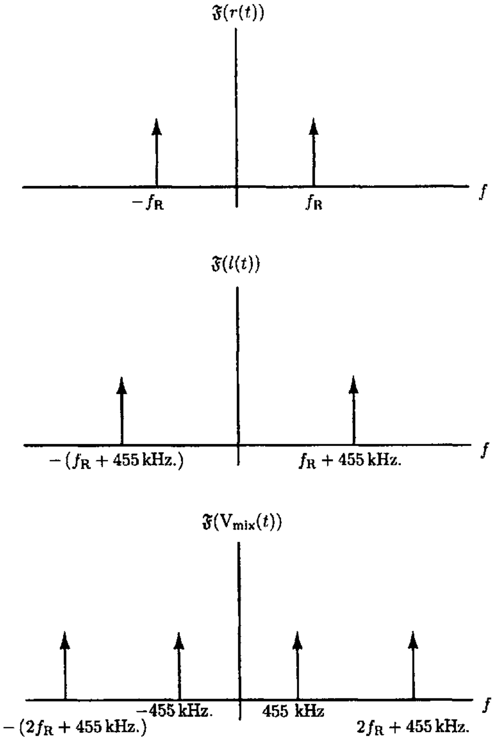


Figure 4-3: Frequency-domain convolution of the two mixer circuit inputs $r(t)$ and $l(t)$, the resulting signal $\mathrm{V}_{\mathrm{mix}}(t)$ includes a component at the intermediate frequency of 455 kHz as well as one at approximately twice the frequency of the received signal f_{R}

However, the single-transistor mixer that will be used does not simply multiply the two signals. Instead, the mixer circuit shown in Figure 4-4 will exploit the nonlinearity of the BJT to create an output that includes the resulting convolution of Figure 4-3 amongst several other frequency components. The output current I_{C} of the mixer is

$$\mathrm{I}_{\mathrm{C}} = \mathrm{I}_0 e^{-\frac{v_{\mathrm{BE}}}{v_{\mathrm{TH}}}}$$

where v_{TH} is the thermal voltage $\frac{\mathrm{kT}}{\mathrm{q}}$. The base-emitter voltage v_{BE} is the difference between the receiver and local-oscillator inputs $r(t)$ and $l(t)$ with some DC bias V_{BE},

$$v_{\mathrm{BE}} = \mathrm{V}_{\mathrm{BE}} + (r(t) - l(t))$$

Substituting the above equation into the output-current equation gives,

$$I_{\mathrm{C}} = e^{-\frac{\mathrm{V}_{\mathrm{BE}}+(r(t)-l(t))}{v_{\mathrm{TH}}}}$$

Using the Taylor series expansion of the output-current, the term of interest is (from [4]),

$$\frac{g_m}{2v_{\mathrm{TH}}} l(t) r(t)$$

A more detailed analysis of the mixer circuits operation can be found in [4], and [7].

In order to extract the output term of interest, a resonant element at the intermediate frequency will be used. This element will be a variable transformer component, the IF301, which includes a capacitor attached to the primary, creating a variable LC tank circuit. This transformer will be attached to the collector of the 2N3904 transistor in Figure 4-4. Students will be able to vary the resonant frequency of this tank circuit in order to tune the mixer to the intermediate frequency. Specific details on the mixer circuit design can be found in [4].

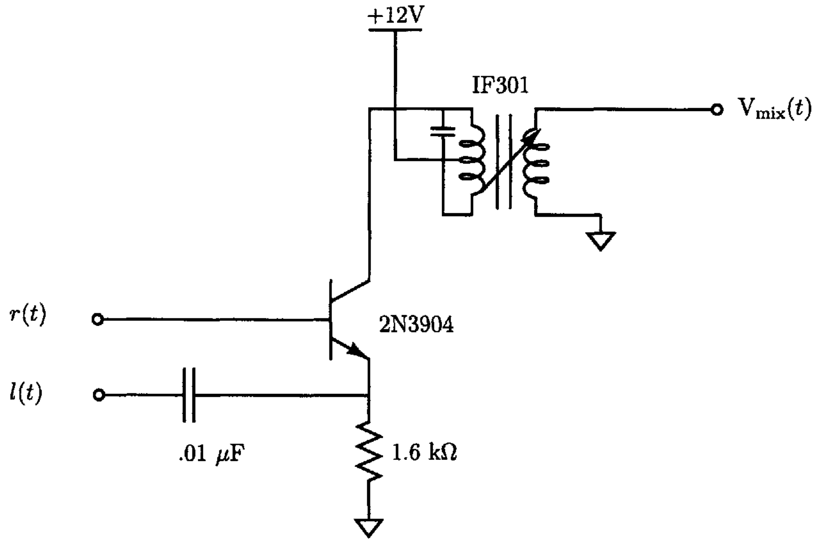


Figure 4-4: Single-transistor mixer circuit to be used in the laboratory kit reception module. The signal $l(t)$ can be taken from the local oscillator circuit or a function generator.

Local Oscillator

The mixer circuit will rely heavily on the signal $l(t)$ being 455 kHz above the received signal $r(t)$. In order to create this signal $l(t)$ with its particular frequency characteristics, students will construct the Hartley oscillator shown in Figure 4-5. The Hartley oscillator uses positive feedback to create an oscillating signal at a frequency set by an LC tank at the collector of its transistor, much like the mixer circuit. Because the oscillators output $l(t)$

must be related to the received signal $r(t)$, the capacitor used in the LC tank will be the mechanically coupled to the variable capacitor used in the receiving antenna in Section 4.1.1. Thus as the user tunes the receiving antenna, the local oscillator is also tuned to match. The transformer attached to the transformer is also adjustable, which will allow students to make minor adjustments to the frequency of $l(t)$. Further details on the Hartley oscillator can be found in [4].

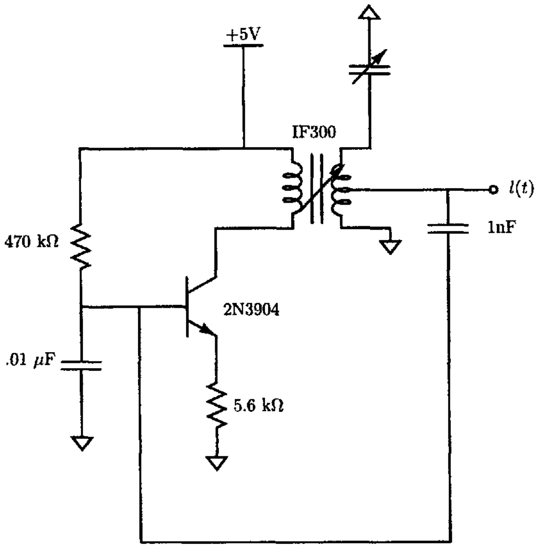


Figure 4-5: Local oscillator circuit used to generate the signal $l(t)$ for the mixer stage. A Hartley oscillator will be used with both a variable capacitor and transformer.

4.1.3 IF Amplifiers

The last stage in the superheterodyne receiver provides amplification. Two identical amplifier circuits will be used, both tuned (like the mixer stage) to the intermediate frequency. Each circuit will use 3 NPN bipolar transistors configured in a common-collector, common-base cascade (see [4] and [7]). At the collector of the output transistor, an adjustable transformer like the one used for the mixer stage will be used by students to tune each amplifier. The complete, dual-amplifier circuit is shown in Figure 4-6 with additional details on the operation of the differential amplifier available in [4]. An oscilloscope capture of the IF amplifiers working to amplify a received signal is shown in Figure 4-7

These last two amplifiers provided much-needed gain at moderate to long reception distances. Typically, a signal of 200 to 500 mV peak-to-peak maximum amplitude will be available as output from the mixer stage. In order for the mixed-down received signal to

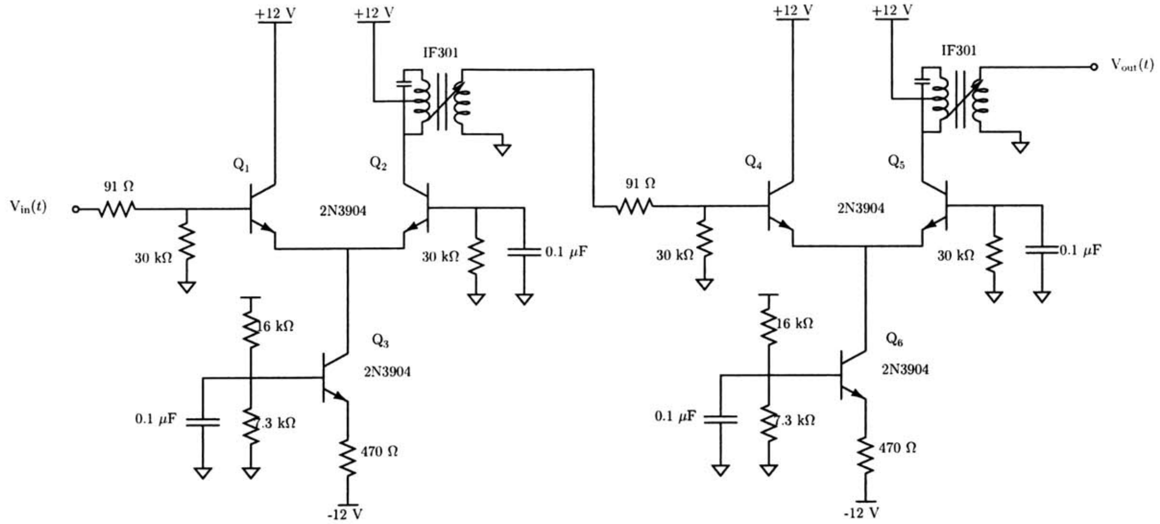
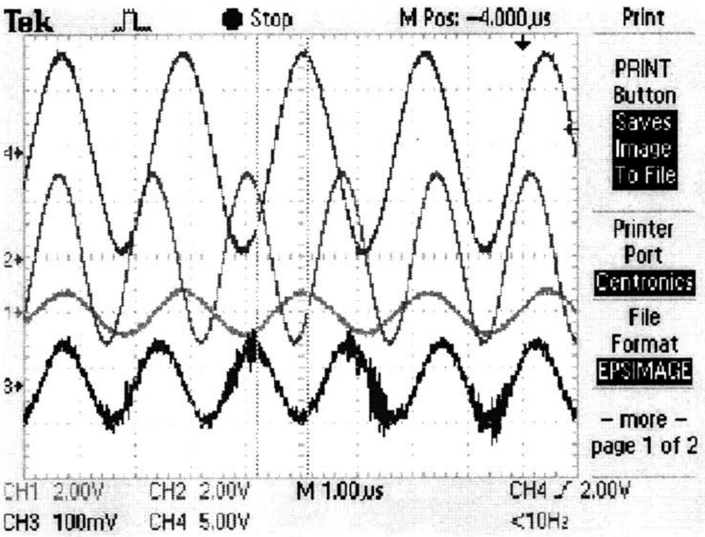


Figure 4-6: Intermediate-frequency amplifier stage, consisting of two differential amplifiers tuned (using the IF301 variable transformer) to 455 kHz



CH1 2.00V CH2 2.00V M 1.00μs CH4 ƒ 2.00V
CH3 100mV CH4 5.00V <10Hz

Figure 4-7: Intermediate-frequency amplifiers working to amplify a received signal. The bottom trace is a 455 kHz signal received at the antenna. The next trace up is the same signal amplified by the mixer stage (still at 455 kHz since the received signal is at the intermediate frequency). The third trace is the output of the first IF amplifier, and the top-most trace is the output of the last IF amplifier.

be made appropriate for processing by other demodulator circuits in the laboratory kit its amplitude must be approximately 5 V peak-to-peak. The two intermediate amplifiers will provide the needed gain of at least 20 to the mixer output to obtain the desired signal levels.

Moreover, it will also be desirable to maintain a constant 5 V peak-to-peak output no matter the magnitude of the mixer output. With a constant output, the superheterodyne receiver becomes more reliable and better able to provide consistently valid output signals for the other on-board demodulator circuits. In order to accomplish this, the gain of the intermediate amplifiers will have to be adjusted automatically as the strength of the received signal changes. A circuit for AGC (Automatic Gain Control) is recommended for this purpose by the ARRL (American Radio Relay League) in [8].

4.2 AM Demodulator

After constructing their superheterodyne reception system, students will use it to pick up a DSB/WC signal broadcast from the laboratory kit transmission module. Thus, prior to testing their receiver, students will construct an appropriate demodulator for the DSB/WC signal. The demodulator will consist of three components: a diode, a resistor, and a capacitor as shown in Figure 4-8. This demodulator is asynchronous, it does not require use of a copy of the carrier (without modulation) to function.

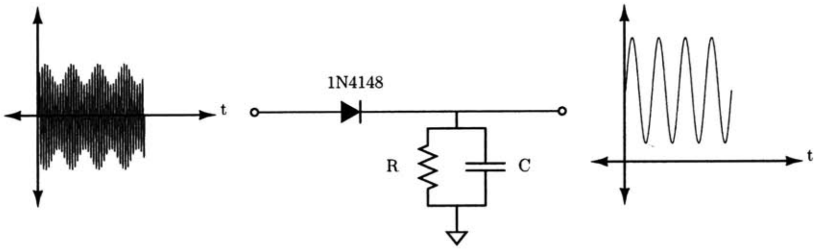


Figure 4-8: Asynchronous AM demodulator circuit. The diode acts as a half-wave rectifier followed by a parallel RC low-pass filter. The circuit output follows the envelope of its input signal.

More specifically, because a constant offset will have been added to the received signal at the transmitter as discussed in Section 3.2.1, the diode can be used as an effective half-wave rectifier. The offset allows the diode used in the demodulator to continue to conduct even when the modulating signal is zero in magnitude; preventing distortion due to cutoff during demodulation. Thus, the diode will effectively pass the positive half of the input signal (above the time axis in Figure 4-8.

Next, in order to extract the envelope of the received signal from the 455 kHz signal, a resistor and capacitor will be placed in parallel after the diode rectifier. This parallel combination will form a low-pass filter. The low-pass filter must pass the slower envelope of the received signal but not the carrier. The intended input signal $y(t)$ is

$$y(t) = e(t)\cos(\omega_c t)$$

with envelope function

$$e(t) = A(1 + m\cos(\omega_m t))$$

In order to follow the envelope, the filter output $V_{\text{dm}}(t)$ needs to be faster than the envelope. This "speed" is expressed using the derivative; that is, the derivative of the filter output voltage must be greater than the derivative of the envelope. This constraint is expressed by the inequality,

$$\frac{d}{dt}V_{\text{dm}}(t) \geq \frac{d}{dt}e(t)$$

However, the derivative of the output voltage must be less than that of the total signal, or the carrier will pass as well. Thus,

$$\frac{d}{dt}V_{\text{dm}}(t) \leq \frac{d}{dt}y(t)$$

Moreover, the filter output voltage is defined by two different equations, one for the capacitor discharge cycle and one for the charging cycle. Solving the derivative constraints for one equation will also solve the constraints for the other. Thus, the more simple discharge equation,

$$V_{\text{dm}}(t) = Ve^{-\frac{t}{\tau}} \quad \tau = RC$$

will be used below to solve for the resistor and capacitor values. The factor V is a constant, expressing the point at time $t = 0$ where the capacitor starts its discharge cycle.

The first constraint inequality will be simplified by first taking the derivative of the discharge equation,

$$\frac{d}{dt}V_{\text{dm}}(t) = \frac{d}{dt}\left(Ve^{-\frac{t}{\tau}}\right) = -\frac{V}{\tau}e^{-\frac{t}{\tau}}$$

Substituting the derivative into the first constraint inequality,

$$-\frac{V}{\tau}e^{-\frac{t}{\tau}} \geq \frac{d}{dt}(A(1+m\cos(\omega_m t)))$$

Taking the derivative of the right-hand side (assuming the discharge factor V is approximately constant over a charge period),

$$-\frac{V}{\tau}e^{-\frac{t}{\tau}} \geq -Am\omega_m\sin(\omega + mt)$$

Canceling the negative factor from both sides of the equation above leaves,

$$\frac{V}{\tau}e^{-\frac{t}{\tau}} \geq Am\omega_m\sin(\omega + mt)$$

To simplify the constraint inequality further, the constant term V will be solved for as follows. The capacitor discharges starting at the voltage V and approaches zero. The starting voltage is determined by the envelope function $e(t)$, as shown in Figure 4-9; the discharge voltage is shown in bold. Starting from the envelope, the capacitor voltage discharges until the next peak in the envelope, approximately following the envelope voltage.

When taking the derivative above, V was assumed to be constant, and now it will be assumed to vary as $e(t)$ varies. This seemingly contradictory set of assumptions is predicated on a third assumption that $e(t)$ varies much more slowly than $\mathrm{V}_{\mathrm{dm}}(t)$. Therefore, equating V to the envelope $e(t)$,

$$Ve^{-\frac{t}{\tau}} \approx A(1+m\cos(\omega_m t))$$

Multiplying both sides by the exponential factor,

$$V \approx A(1+m\cos(\omega_m t))e^{\frac{t}{\tau}}$$

Substituting this value for V into the constraint inequality,

$$\frac{A(1+m\cos(\omega_m t))}{\tau} \geq Am\omega_m\sin(\omega_m t)$$

Shifting terms to either side of the inequality yields,

$$\omega_m \tau \leq \frac{1 + m\cos(\omega_m t)}{m\sin(\omega_m t)}$$

The right-hand side of the inequality can be approximated as the reciprocal of the modulation index:

$$\frac{1 + m\cos(\omega_m t)}{m\sin(\omega_m t)} \approx \frac{1}{m}$$

Therefore, the inequality becomes,

$$\omega_m \tau \lessapprox \frac{1}{m}$$

Dividing both sides by the modulating frequency ω_m,

$$\tau \lessapprox \frac{1}{m\omega_m}$$

One last substitution for the time constant simplifies the inequality to (also found in [9]):

$$RC \lessapprox \frac{1}{m\omega_m}$$

Using this constraint approximation, students can choose appropriate values for the demodulator resistor and capacitor based upon the modulating signal frequency and modulation constant. For example, a 1 kHz modulating signal with modulation index of 0.25 will require a time constant no larger than 4 ms. For a capacitor value of 0.1 μF, the constraint inequality requires a resistor smaller than 40 kΩ.

4.2.1 Demodulating An Audio Signal

Lastly, to test the complete superheterodyne-demodulator system, students will demodulate an audio signal broadcast from the kit transmission module. The signal will be received and mixed-down using the superheterodyne receiver, and then demodulated using the asynchronous AM demodulator described above. To test the effectiveness of their reception

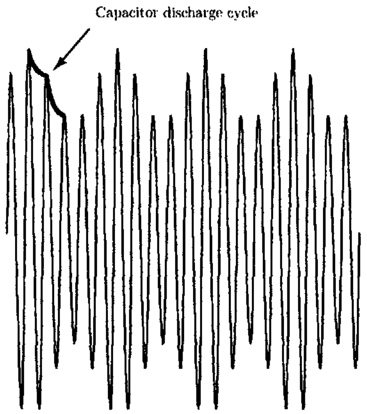


Figure 4-9: Asynchronous AM demodulators low-pass filter following the envelope of its input. The bold line is the capacitor discharge cycle.

system, students will then play the demodulated audio signal using an audio amplifier and a speaker.

The modulating signal will be a recorded audio signal that will have frequency content between 500 Hertz and 1.5 kilohertz. The modulation index will be assumed to be 0.5. The upper limit of the assumed frequency range will limit the demodulator filter time constant the most. Students will use these properties of the modulating signal to choose component values for their asynchronous AM demodulator. Using the constraint inequality from Section 4.2, the filter time constant will be:

$$RC \lessapprox \frac{1}{(0.5)(2\pi)(1500)} \approx 212 \text{ ns}$$

Therefore, with a capacitor of 0.1 μF, students will need to use a resistor of value less than aprooximately 2.12 kΩ. After using the newly configured (with correct resistor and capacitor values) demodulator, students will take the demodulator output and use an LM386 audio amplifier circuit to play the audio on a standard 8 Ω speaker. The circuit to be used by students is a configuration, recommended by the LM386 manufacturer in [10], it is shown in Figure 4-10.

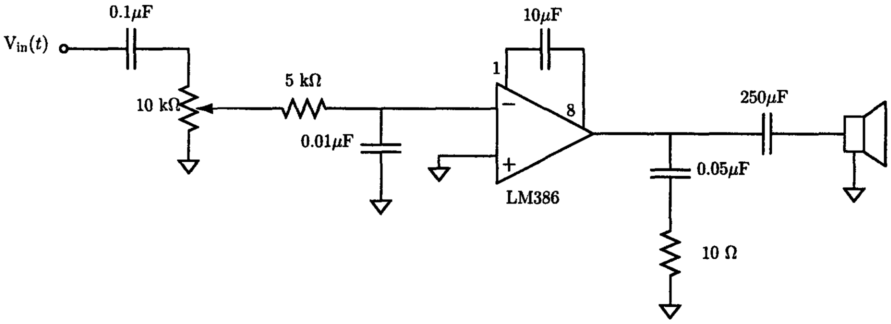


Figure 4-10: Audio amplifier circuit to be used by students to play back their demodulated audio signal.

Chapter 5

Lab 3: Phase-Locked Loops, Clock Recovery, and FM Demodulation

PLLs (Phase-Locked Loops) are valuable tools in commercial wireless receivers. PLLs are used to perform a wide range of functions including clock signal recovery and FM demodulation. In this third laboratory assignment students will explore both of these PLL applications. They will use two different PLL integrated circuits, one for each of the two application areas. First, students will investigate the three building blocks of any PLL: the phase detector, loop filter, and voltage-controlled oscillator. Each component is vital to designing and constructing a PLL for specific applications.

After familiarizing themselves with the key PLL components, students will employ the CD4046 integrated circuit to explore PLL design techniques. Students will be charged with the task of designing PLL circuits to meet specific criteria and discuss the use of PLLs for recovering a valid clock signal from digital signals with varying periods. Next, students will learn about FM demodulation using the LM565 integrated circuit. The modulated signals from the XR2206 of Chapter 3 can be used as inputs for the FM demodulator in this chapter. The LM565-based demodulator will then be used as the primary FM signal demodulator for the lab kit reception module.

Thus, once this laboratory assignment is completed students should have a general familiarity with the operation of a PLL. Additionally, they will have experienced the design and construction of PLL circuits for different operating constraints including a major application in communication systems, FM demodulation.

5.1 PLL Overview

A phase-locked loop is a collection of circuits placed in a feedback loop. This loop acts to reduce the difference in phase between its input and output. The phase of a signal is defined as

$$\phi(t) = \omega t + \theta$$

The derivative of the phase is the frequency ω (with θ considered a constant):

$$\frac{d\phi}{dt} = \omega$$

Students in this course will use this definition of phase and a linearized feedback model to study PLLs [4]. The linearized feedback model consists of three blocks as shown in Figure 5-1. The first block is a phase detector, calculating phase differences between its inputs. The second block is called the loop filter; responsible for several different loop operations depending on the choice of phase detector. The last block is a VCO (Voltage-Controlled Oscillator). The VCO generates a frequency at its output that is proportional to the DC voltage at its input.

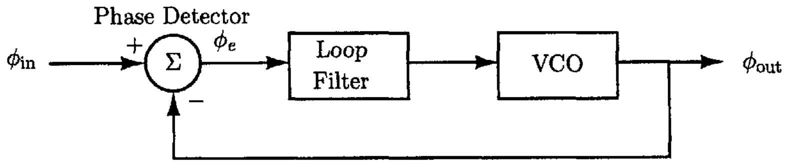


Figure 5-1: Block diagram of a linearized feedback model for a phase-locked loop.

The PLL model operates by comparing the phase of its input and output using the phase-detector block. The output of the phase detector is processed by the loop filter to create an appropriate DC voltage for the VCO block. The VCO block creates an output signal that has a frequency proportional to the phase difference. This output is fed back to the phase detector, driving the input and output phases to be equal (a condition known as being "in lock"). The operation of each block is explained in greater detail below.

5.1.1 Phase Detector

In the block diagram of Figure 5-1 the phase detector, sometimes called a phase comparator, is modeled as a subtractor operating on the phases of its two inputs. The difference in phases, called the phase error ϕ_e, is its output. The phase detector has a constant gain associated with it, $\mathrm{K_D}$, that multiplies the error. This gain constant varies depending on the phase detector implementation.

Phase detectors are created using several different types of devices and circuits. Two of the most common phase detectors are analog multipliers and digital XOR gates. For PLLs that work with sinusoidal signals, analog multipliers have become the predominant choice for a phase detector [4]. Digital PLLs, used for tasks such as clock recovery, often employ an exclusive-or gate for phase detection. XOR gates are effective digital phase detectors and have greater simplicity in construction for integrated circuits as compared to analog multipliers. Students will use an analog multiplier phase detector for FM demodulation and two circuits for their CD4046 designs: an XOR gate and another, different, digital circuit with different output conditions.

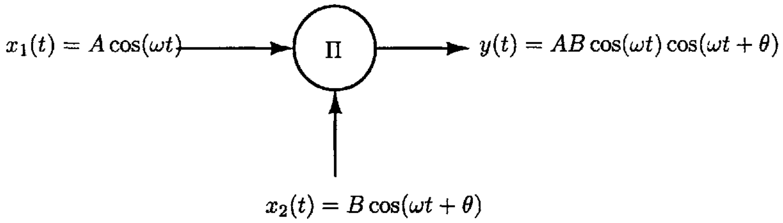
The analog multiplier, applied to two sinusoidal signals $x_1(t)$ and $x_2(t)$ as shown in Figure 5-2, produces as output the signal $y(t) = x_1(t)x_2(t)$. Assuming that $x_1(t)$ and $x_2(t)$ only differ by a shift in phase θ, the output can be represented (using trigonometric manipulations) by a constant term and a term at twice the frequency of the input. For the signals $x_1(t)$ and $x_2(t)$, the multiplication results in

$$y(t) = \frac{AB}{2}\left[\cos(\theta) - \cos(2\omega t + \theta)\right]$$

The time-average value of this result is the constant term $\frac{AB}{2}\cos(\theta)$ which is directly proportional to the difference in phase between the two inputs. This constant term is the input to a VCO, thus it must be extracted from $y(t)$ by way of a loop filter as described later in the section on loop filter design. Interestingly, the constant term is reduced to zero only if the phase difference between the two inputs is $\frac{\pi}{2}$. Because of this property, the analog multiplier is known as a quadrature phase detector. Moreover, when in lock, the incremental gain constant K_D of the analog multiplier is

$$K_D = \frac{d}{d\phi}\left(\frac{AB}{2}\cos(\theta)\right) = -\frac{AB}{2}\sin(\theta)\bigg|_{\theta=\frac{\pi}{2}} = -\frac{AB}{2}\ \frac{\text{V}}{\text{rad}}$$

A version of the analog multiplier phase detector is included in the LM565 PLL IC to be used by students later in this laboratory assignment [11].



$x_1(t) = A\cos(\omega t)$
Π
$y(t) = AB\cos(\omega t)\cos(\omega t + \theta)$
$x_2(t) = B\cos(\omega t + \theta)$

Figure 5-2: Analog multiplier acting as a phase detector. The output signal $y(t)$ contains a constant term directly related to the phase difference between the inputs.

Another common type of phase detector, the XOR gate, is available in the CD4046, called PDI [12]. The XOR behaves exactly as the multiplier would for antipodal square waves but acts on digital square waves (e.g. taking values of either 0 or 5 V). For two digital inputs $x_1(t)$ and $x_2(t)$ as shown in Figure 5-3, the output is also a square wave of approximately twice the frequency (as the antipodal output would have been using an analog multiplier). The duty cycle of the output square wave is directly related to the difference in phase of the inputs. Moreover, the duty ratio directly affects the average value of the output waveform, much like the analog multiplier. Thus, the average of the square wave output varies with the phase difference as shown in Figure 5-4. Furthermore, a phase difference of $\frac{\pi}{2}$ rad results in zero average output, thus the XOR gate is also a quadrature phase detector, with incremental gain constant $K_D = \frac{V_{dd}}{\pi}\ \frac{\text{V}}{\text{rad}}$ [4].

Lastly, it is prudent to briefly mention the second phase detector included with the CD4046. Labeled PDII, the second phase detector is a sequential phase detector of complex design [12]. In contrast with the quadrature phase detectors mentioned above, PDII will cause the PLL to lock under 0 rad of phase difference. While the details of its implementation will not be discussed here, students will be encouraged to compare it to the other two phase detectors discussed and incorporate it into their designs.

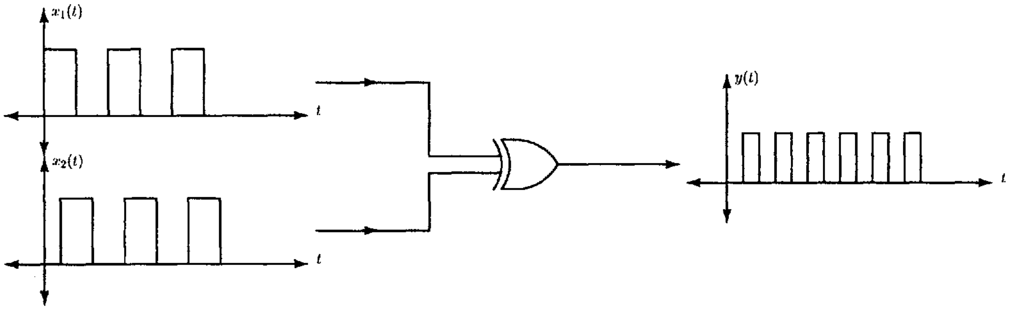


Figure 5-3: XOR phase detector working on two square wave inputs. The output waveform, like the analog multiplier, is at twice the frequency and has a component proportional to the phase difference of the inputs (in this case the duty ratio).

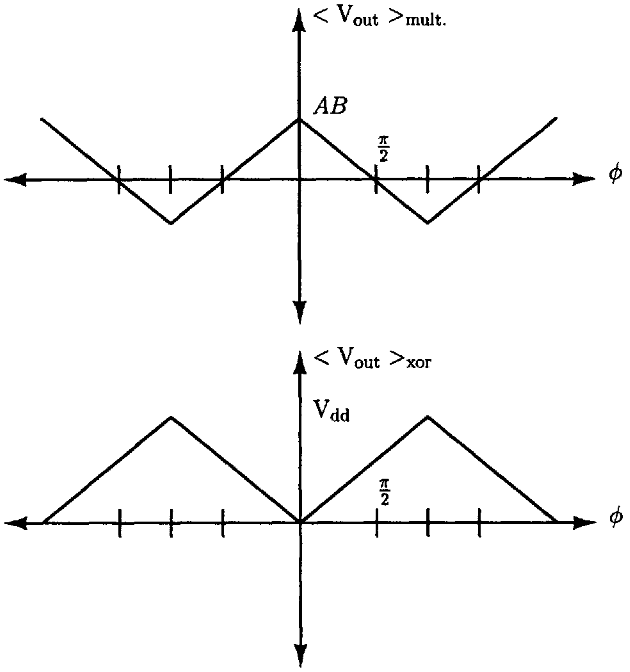


Figure 5-4: Average phase detector output voltage for both an analog multiplier (using antipodal square waves) and an XOR gate (using digital signals) from [4]. The XOR average output is an inverted, offset version of the multiplier output. Although it does not seem so, it still behaves like a quadrature phase detector, producing minimal output with a phase difference of $\frac{\pi}{2}$ rad.

5.1.2 Loop Filter

Each of the three phase detectors mentioned previously has an output signal consisting of two components: a constant term proportional to the phase difference of the inputs and a second term that is periodic (often a double-frequency term). The loop-filter component in a PLL is responsible for processing this phase detector output waveform and filtering out its periodic portion.

Since only the average value or constant term is desired, the loop filter will be either a passive or active low-pass filter. A low-pass filter will be designed to severely attenuate the periodic component from the phase detector output and pass only the DC term. Passive low-pass filters such as a single resistor and capacitor network are effective, but it will be difficult to decouple the feedback loops bandwidth from its damping ratio. More complex passive filters and also active filters will allow for independent control of bandwidth and damping by adding an additional zero to the loop transfer function. More on these factors will be discussed in Section 5.2.

5.1.3 Voltage-Controlled Oscillator

The voltage-controlled oscillator is responsible for generating the oscillating output of the PLL. The VCO acts as a voltage-to-frequency converter. A input DC voltage produces as output a periodic signal whose frequency is proportional to the input. The ratio of the change in output frequency to the change in input voltage is known as the VCO constant, $\mathrm{K_O}$, with units of $\frac{\mathrm{rad}}{\mathrm{Vs}}$.

The VCO is modeled as an integrator with a gain of $\mathrm{K_O}$ in Figure 5-1. Since instantaneous phase is the integral of frequency, an integration must be built into the VCO model to maintain consistency in the loop variables. This integration is vital to the PLL dynamics discussed in Section 5.2 on PLL design.

The implementation details of a VCO vary wildly from application to application. As a result, the VCO constant $\mathrm{K_O}$ must be measured on each different PLL device used. Students will be required to measure $\mathrm{K_O}$ for both the LM565 and CD4046 prior to working with them in the laboratory.

5.2 PLL Design and Application

The following two design exercises will explore two separate applications of PLLs in communication systems. First, students will use a CD4046 PLL to create a clock recovery system. This system will be used to explore the concept of using a data signal with varying duty cycle and/or period to recreate a single-frequency clock signal. The PLL design properties of lock-in range and phase margin will be major factors in clock recovery design. Next, students will create an FM demodulator. The LM565 PLL will be used, operating at the superheterodyne intermediate frequency of 455 kHz from Chapter 4. The maximum phase error of the PLL will be a significant design factor in the demodulator design.

5.2.1 PLL Design Using The CD4046

The first set of PLL exercises will be used to get students comfortable with PLL design techniques. Because of the predominant PLLs in digital systems, the CD4046 will be used in these exercises. The CD4046 includes 2 phase detectors and a VCO suitable for working with digital, square wave, signals. In this set of laboratory exercises, students will also explore simple clock recovery situations that will use square wave signals of varying duty cycle to test the CD4046 PLLs ability to recover a clock signal.

Students will undergo two different PLL design exercises with the CD4046. The first design will employ a passive loop filter and an XOR gate phase detector. The second design will use the non-quadrature PDII phase detector (discussed in Section 5.1) and an active loop filter.

Each designs success is directly related to the lock-in range of the PLL design. If the input signal to the PLL is close enough to the free-running frequency of its VCO the PLL will "lock up" [13]. That is, the PLL will follow the input frequency after a transient [13]. The lock-in range of both PLLs will be calculated and compared in reference to their ability to establish and hold a clock signal. Then, the clock recovery abilities of each design will be tested using variable duty cycle square waves in hardware.

XOR Phase Detector and Passive Loop Filter

To start gaining familiarity with the CD4046 IC, students will complete a preliminary clock recovery design using an XOR gate phase detector (called PDI) and a passive, first-order, low-pass loop filter. A modified version of the PLL block diagram from Section 5.1 shows the intended design with the salient circuit components for each functional block included, similar to [4]. The block diagram is shown in Figure 5-5. The resistor R_3 and capacitor C_1 make up the first order loop filter. The VCO of the CD4046 has its operating frequency set by 3 circuit components: R_1, R_2, and C_2. The frequency of oscillation for the VCO, from [4], is approximated as:

$$\omega_{\text{osc}} \approx \frac{2\left(\frac{\text{V}_\text{C}-1}{\text{R}_1} + \frac{4}{\text{R}_2}\right)}{\text{C}_2} \quad \frac{\text{rad}}{\text{sec}}$$

Where V_C is the control, or input voltage to the VCO. Consequently, the VCO gain constant is the derivative of ω_{osc} with respect to the control voltage V_C. Thus, the constant is defined as:

$$\text{K}_\text{O} = \frac{d}{d\text{V}_\text{C}}\omega_{\text{osc}} \approx \frac{2}{\text{R}_1\text{C}_2} \quad \frac{\text{rad}}{\text{Vs}}$$

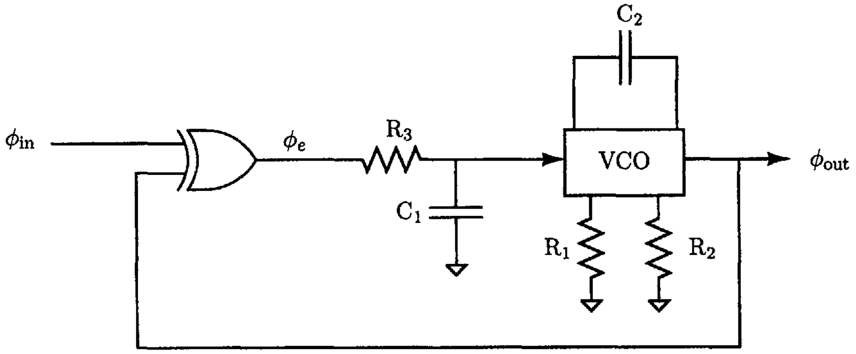


Figure 5-5: Block diagram of the CD4046 circuit using a passive loop filter and XOR phase detector.

Using these circuit components and parameters, students will design and construct a PLL circuit following the guidelines from [4]. The PLL should have a crossover frequency (ω_c) of 1000 $\frac{\text{rad}}{\text{sec}}$, phase margin ($\Phi_m$) of 45°, and VCO oscillating frequency (ω_{osc}) of $40\pi \times 10^3 \frac{\text{rad}}{\text{sec}}$:

$$\omega_c = 1000 \ \frac{\text{rad}}{\text{sec}} \qquad \phi_m = 45^\circ \qquad \omega_{\text{osc}} = 40\pi \times 10^3 \ \frac{\text{rad}}{\text{sec}}$$

Students should begin their design by calculating the PLL loop transmission as follows:

$$L(s) = \text{K}_\text{D}H(s)\frac{\text{K}_\text{O}}{s} = \frac{\text{K}_\text{D}\text{K}_\text{O}}{s\left(\text{R}_3\text{C}_1 s + 1\right)}$$

The closed-loop transfer function, will be calculated from the loop transmission using Black's formula:

$$P(s) = \frac{L(s)}{1+L(s)} = \frac{\frac{\mathrm{K_D K_O}}{s(\mathrm{R_3C_1}s+1)}}{1+\frac{\mathrm{K_D K_O}}{s(\mathrm{R_3C_1}s+1)}} = \frac{\frac{\mathrm{K_D K_O}}{(\mathrm{R_3C_1})}}{s^2+\frac{s}{(\mathrm{R_3C_1})}+\frac{\mathrm{K_D K_O}}{(\mathrm{R_3C_1})}}$$

Thus, the closed-loop system is of second-order, with natural frequency and damping ratio equal to:

$$\omega_n = \sqrt{\frac{\mathrm{K_D K_O}}{\mathrm{R_3C_1}}}\ \frac{\mathrm{rad}}{\mathrm{sec}} \qquad \zeta = \frac{1}{2}\sqrt{\frac{1}{\mathrm{R_3C_1K_DK_O}}}$$

The properties of the closed-loop function bring to light the negative effects of using a first-order filter. The loop dynamics, as defined by the natural frequency and damping ratio, are both related to the filter components $\mathrm{R_3}$ and $\mathrm{C_1}$. The natural frequency and the damping ratio cannot be controlled independently, leaving the PLL designer with limited choices in terms of PLL performance. Using a second-order, active filter, as discussed in the next design will improve this.

Additionally, from the loop transfer function, students will observe that the two poles will drive the phase of the system to $-180°$. In order to achieve a phase margin of 45°, the pole controlled by $\mathrm{R_3}$ and $\mathrm{C_1}$ must be placed at the desired crossover frequency of 1000 $\frac{\mathrm{rad.}}{\mathrm{sec.}}$. Thus, the time constant $\mathrm{R_3C_1}$ must be set to 1 millisecond. Typical values for $\mathrm{R_3}$ and $\mathrm{C_1}$ will be 100kΩ and .01μF respectively. With these components now specified, the necessary VCO constant can then be calculated by evaluating the magnitude of $L(s)$ at the crossover frequency:

$$|L(j\omega_c)| = \left|\frac{\mathrm{K_D K_O}}{-\omega_c^2\mathrm{R_3C_1}+j\omega_c}\right| = 1$$

For a time constant of,

$$\mathrm{R_3C_1} = 10^{-3}\ \mathrm{s}$$

The loop transmission becomes

$$|L(j\omega_c)| = \frac{\mathrm{K_D K_O}}{\omega_c\sqrt{2}}$$

For the time cosntant,

$$R_1C_2 \approx .582 \text{ ms}$$

Therefore, the VCO constant is,

$$K_O = \frac{2}{.582 \text{ ms}} \approx 3.44 \quad \frac{\text{rad}}{\text{Vs}}$$

Typical values for the timing resistor R_1 and timing capacitor C_2 are 582kΩ and 1nF respectively. With 2 of its 3 circuit components constrained, the last specification on free-running frequency ω_{osc} sets the last of the VCOs components, R_2. Using the formula for ω_{osc}, and assuming a nominal control voltage of half the intended supply rail of 5 V:

$$\omega_{\text{osc}} = 40\pi \times 10^3 \quad \frac{\text{rad}}{\text{sec}} \approx \frac{2\left(\frac{V_C - 1}{R_1} + \frac{4}{R_2}\right)}{C_2}$$

Solving for the unknown resistor R_2 gives,

$$R_2 \approx \frac{8}{\omega_{\text{osc}}C_2} - \frac{4R_1}{1.5} \approx 66 \text{ k}\Omega$$

After selecting R_2, the first PLL design exercise is complete. The circuit schematic of the CD4046 configured with PDI and the first-order filter is shown in Figure 5-6. The tuning range of the VCO is quite limited in this configuration, so limited that it predominates the lock-in range. Thus, in this design example, the CD4046 circuit will lose lock once the input reaches a frequency at the limit of ω_{osc}. Students will be encouraged to use a function generator to sweep the input frequency to the CD4046 circuit and observe the PLL acquiring and losing lock. This preliminary example should solidify the importance of good locking range and the need to have independent control of the loop dynamic properties ω_n and ζ.

PDII and Active Loop Filter

Next, students will design a second PLL circuit using the sequential phase detector PDII and an active, second-order, loop filter. The second-order filter takes the place of the RC

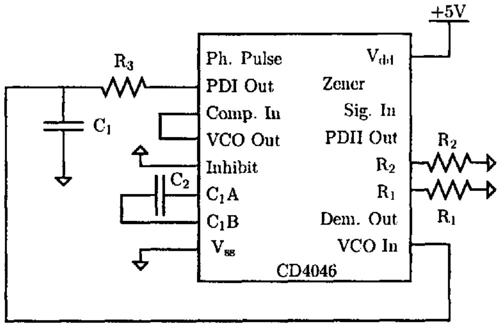


Figure 5-6: CD4046 PLL circuit using a passive loop filter and XOR phase detector.

passive network in the block diagram of Figure 5-7. R_1, R_2, and capacitor C_2 will still set the operating point for the VCO as in the previous design exercise. Resistors R_3, R_4 and capacitor C_1 set the dynamics of the loop filter. Consequently, this loop filter design introduces a signal inversion at its output that must be corrected by using the unity-gain inverter shown in Figure 5-7. Also, the active filter requires its positive input to be set to half the supply voltage to increase the range of output voltages [4]. The phase detector, PDII, has no additional components to add from the previous design, but does have a different gain constant, approximately one-fourth as large as that of the XOR, making K_D equal to $\frac{V_{dd}}{4\pi}$.

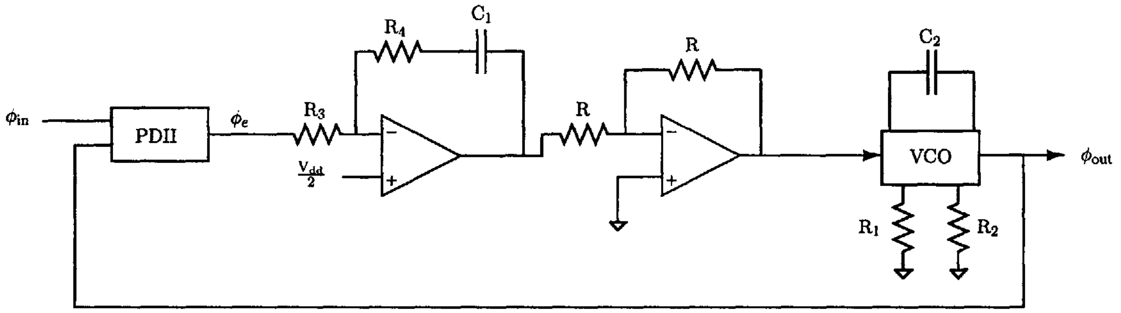


Figure 5-7: Block diagram of the CD4046 circuit using an active loop filter and phase detector PDII from [4].

Continuing on, this design exercise has similar target specifications to the last, namely:

$$\omega_c = 1000 \ \frac{\text{rad}}{\text{sec}} \qquad \phi_m = 45^\circ \qquad \omega_{osc} = 40\pi \times 10^3 \ \frac{\text{rad}}{\text{sec}}$$

However, a new requirement will be added on the lock-in range. Specifically, the lock-in range should be ±10 kHz about the intended oscillation frequency of 20 kHz.

The new loop filter transfer function will be:

$$L(s) = \frac{\mathrm{K_D K_O}(\mathrm{R_4 C_1}s + 1)}{\mathrm{R_3 C_1}s^2}$$

With closed loop transfer function:

$$\tau_1 = \mathrm{R_3C_1} \qquad \tau_2 = \mathrm{R_4C_1}$$

$$P(s) = \frac{\mathrm{K_DK_O}(\tau_2 s + 1)/\tau_1}{s^2 + \frac{\mathrm{K_DK_O}\tau_2}{\tau_1 s} + \frac{\mathrm{K_DK_O}}{\tau_1}}$$

From the transfer function, as in the previous design, the natural frequency and damping ratio are:

$$\omega_n = \sqrt{\frac{\mathrm{K_DK_O}}{\tau_1}}\ \frac{\mathrm{rad}}{\mathrm{sec}} \qquad \zeta = \frac{1}{2}\tau_2\omega_n$$

Although the damping ratio depends on the natural frequency, the second time constant τ_2 affords an independent control on its value in contrast with the previous design with the first-order filter.

The added zero in the loop transfer function gives a boost to the system phase margin. With two poles, the system phase would approach $-180°$. Thus, the zero must be set at the desired crossover frequency to meet the 45° phase margin specification. The product $\mathrm{R}_\tau\mathrm{C}_1$ must therefore equal 1ms, with typical values for R_4 and C_1 of 100kΩ and .01μF respectively.

Evaluating the loop transfer function at crossover constrains the last resistor in the active filter, R_3,

$$|L(j\omega_c)| = \left|\frac{10^{-3}((j\omega_c) + 1)}{\mathrm{R_3C_1}(j\omega_c)}\right|$$

Solving for R_3,

$$\mathrm{R}_3 = \frac{\mathrm{K_OK_D}\sqrt{2}}{\omega_c^2} = \frac{\mathrm{V_{dd}K_D}\sqrt{2}}{\pi\omega_c^2} \quad \Omega$$

The calculation for R_3 involves $\mathrm{K_O}$, whose value is restricted by the new lock range requirement. Specifically, the control voltage of the VCO has a narrowing effect on its oscillation frequency ω_{osc} [4]. The control voltage is effective only from 1.2V to 5V. The VCO must be able to vary its frequency by 10 kHz in either direction over this range in order to maintain lock. Using 2.5V as the voltage that corresponds to the center of the VCO frequency range, the VCO must change by 10 kHz, constraining K_O as follows:

$$K_O \geq \frac{(2\pi)10\text{ kHz}}{1.3\text{V}} \approx 4.8 \times 10^4 \ \frac{\text{rad}}{\text{Vs}}$$

Typical component values that meet this specification are $.001\mu$F, 42kΩ, and 130kΩ for C_2, R_1, and R_2 respectively. With K_O set, R_3 will then take on a value of 2.8MΩ, completing the design. A circuit schematic of the new CD4046 design is shown in Figure 5-8.

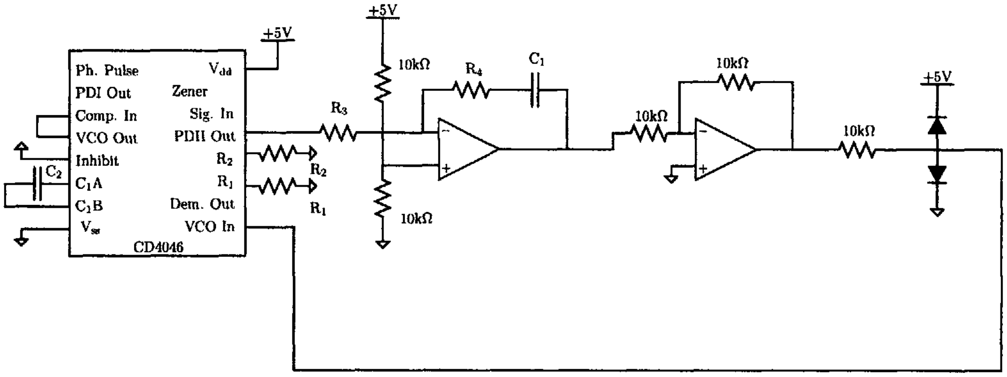


Figure 5-8: CD4046 circuit using an active loop filter and sequential phase detector PDII.

Like the first design, the VCO tuning range sets the lock-in range of the PLL. Moreover, the lock-in range requirement significantly improves the PLLs performance by reducing its sensitivity to frequency changes at the input. Students will be asked to perform the same frequency sweep on this PLL design they performed on the last. Then, they can experiment with square wave signals of varying duty cycle to test the PLLs ability to maintain an output frequency equal to the inputs fundamental, effectively recovering a clock signal.

5.2.2 FM Demodulation Using The LM565

The second PLL application to be studied is FM demodulation. Frequency modulation deliberately introduces a phase disturbance in the carrier frequency to be broadcast. This phase disturbance is proportional to the modulating signal. A PLL can be used to detect this phase difference using its phase detector. The loop filter processes the phase detector signal leaving the modulating signal at its output. The loop filter output is therefore considered the PLL output signal in this application. A block diagram of this process is shown in Figure 5-9.

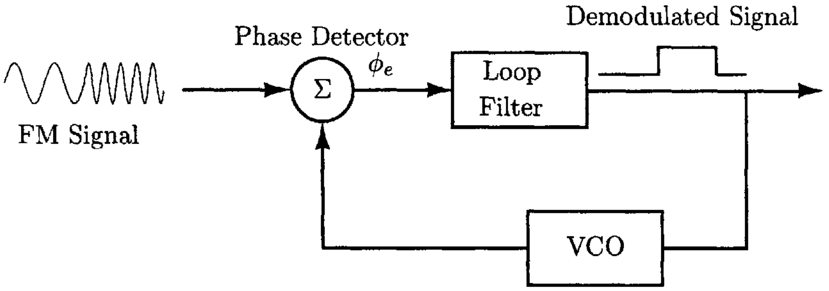


Figure 5-9: Block diagram of a PLL acting as an FM demodulator. The loop filter is now considered the output of the loop, in contrast to the canonical loop displayed in Figure 5-1.

Students will use the LM565 PLL to design an FM demodulator circuit. This circuit will be included in the laboratory kit reception module. Being integrated into the reception module, the FM demodulator will be designed to process signals from the superheterodyne receiver of Chapter 4. As a result, the LM565 will be set to operate at a fixed frequency of 455 kHz Moreover, the superheterodyne output signals will all be sinusoidal, which will be processed effectively by the LM565 analog multiplier phase detector. The demodulator design will use a first-order, passive loop filter.

Unlike the first-order filter used in Section 5.2.1, the filter that will be used for the FM demodulator will contain one pole and one zero. The additional zero in the filter results from a built-in resistor, R_b, at the phase detector output included by the manufacturer. Instead of simply adding a capacitor after this built-in resistor to create a single-pole filter, another resistor R_1 will be added to create the complete loop filter shown in Figure 5-10.

This pole-zero filter will provide a significant advantage over the single-pole filter used previously. It will allow for independent control of the close-loop parameters ω_n and ζ. The time constant associated with the pole will control the natural frequency while the time constant associated with the zero will control the damping ratio. A complete circuit schematic for the overall design is shown in Figure 5-11.

The design specifications for the PLL will require a center frequency equal to the superheterodyne receivers intermediate frequency of 455 kHz A maximum modulating frequency of 5 kHz will be assumed along with a maximum possible deviation of ±5 kHz from the center frequency. Under these conditions, the peak phase error, ϕ_e, will be required to be less than $\frac{\pi}{2}$ rad, reducing noise modulation at high noise levels [14]. The PLL closed loop

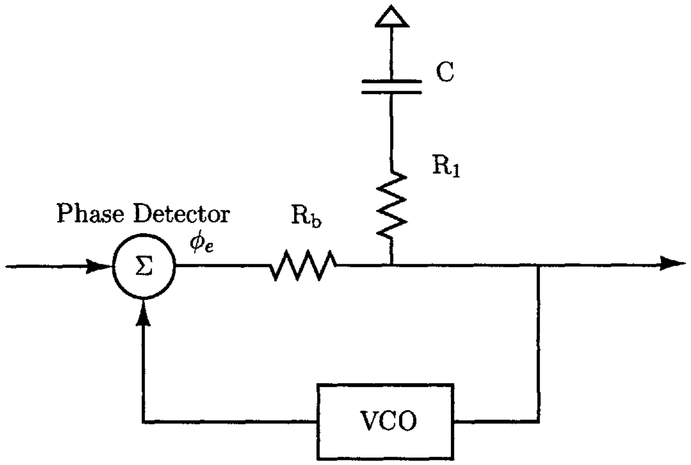


Figure 5-10: Pole-zero loop filter to be used for the FM demodulator. Resistor R_b is built into the IC by the manufacturer. Using R_b along with R_l and capacitor C will give students independent control of ω_n and ζ.

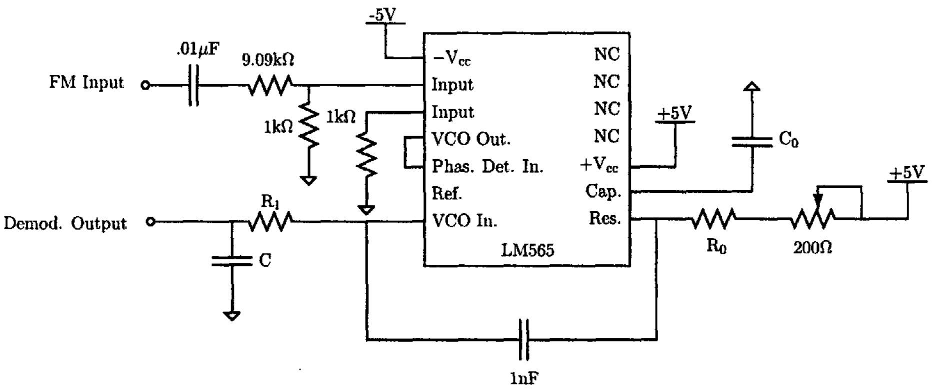


Figure 5-11: LM565 FM demodulator circuit. An AC coupling input capacitor and attenuation network is included at the input.

dynamics should be designed to have a damping ratio of $\frac{\sqrt{2}}{2}$ and a worst-case natural frequency equal to the maximum modulating signal frequency. Thus, the design specifications can be summarized as:

$$\omega_{\text{osc}} = 910\pi \times 10^3 \ \frac{\text{rad}}{\text{sec}} \qquad \Delta\omega = \pm 10^4\pi \ \frac{\text{rad}}{\text{sec}} \qquad \omega_{m_{\max}} = 10^4\pi \ \frac{\text{rad}}{\text{sec}}$$

$$\phi_{e_{\max}} \leq \frac{\pi}{2} \qquad \zeta = \frac{\sqrt{2}}{2} \qquad \omega_n = \omega_m$$

Students should begin their design by consulting the LM565 datasheet and an application note by National Semiconductor on PLL design ([11] and [14] respectively). The LM565 datasheet provides a formula for determining the product of the phase detector constant and VCO constant. Using the 10V total supply voltage for the LM565, the formula yields:

$$\mathrm{K_D K_O} = \frac{33.6 f_0}{\mathrm{V_C}} = \frac{33.6\left(455 \times 10^3\right)}{10} \approx 1.529 \times 10^6 \quad \mathrm{Hz}$$

The datasheet also provides a very rough formula for setting the VCO free-running frequency using the timing resistor $\mathrm{R_0}$ and timing capacitor $\mathrm{C_0}$,

$$f_{\mathrm{osc}} \approx \frac{0.3}{\mathrm{C_0 R_0}} \quad \mathrm{Hz}$$

Students will be encouraged to use a 1 nF timing capacitor for their design. Thus students will calculate an appropriate timing resistor value as follows:

$$f_{\mathrm{osc}} = \frac{\omega_{\mathrm{osc}}}{2\pi} = 455 \times 10^3 \ \mathrm{Hz} = \frac{0.3}{\mathrm{R_0} \times 10^{-9}} \quad \mathrm{Hz}$$

For a 455 kHz oscillating frequency, the timing resistor is,

$$\mathrm{R_0} = 659\Omega$$

However, this calculation will not produce the correct center frequency in the laboratory. The formula provided by [11] allows for too much error (approximately ±30%) in the resulting center frequency for a choice of timing resistor and capacitor. As a result, students will have to adjust their resistor value slightly and include a potentiometer for on-site adjustment if necessary. It will be recommended that students use a 390 Ω resistor and 200 Ω potentiometer in series for their timing resistance. This combination will meet the center frequency specification.

Next, in order to meet the natural frequency and damping ratio specifications, students will start by determining the loop transfer function of the LM565 PLL circuit. The loop transfer function is

$$L(s) = \frac{(\mathrm{K_D K_O})(\mathrm{R_1 C}s + 1)}{s\left((\mathrm{R_b} + \mathrm{R_1})\mathrm{C}s + 1\right)}$$

Using the loop transmission in conjunction with Black's formula, the closed-loop transfer function will be

$$P(s) = \frac{L(s)}{1+L(s)} = \frac{\frac{(\mathrm{K_D K_O})(\tau_1 s+1)}{s(\tau_2 s+1)}}{1+\frac{(\mathrm{K_D K_O})(\tau_1 s+1)}{(\tau_2 s+1)}} = \frac{\mathrm{K_D K_O}(\tau_1 s+1)}{s^2+\left(\frac{1}{\tau_2}+\frac{\mathrm{K_D K_O}\tau_1}{\tau_2}\right)s+\frac{\mathrm{K_D K_O}}{\tau_2}}$$

Where the time constant τ_1 is $\mathrm{R_1 C}$ and τ_2 is $(\mathrm{R_1}+\mathrm{R_b})\,\mathrm{C}$. From the closed-loop transfer function, the natural frequency will be

$$\omega_n = \sqrt{\frac{\mathrm{K_D K_O}}{\tau_2}} \approx \sqrt{\frac{\mathrm{K_D K_O}}{\mathrm{R_b C}}} \quad \frac{\mathrm{rad}}{\mathrm{sec}} \qquad \mathrm{R_1} \ll \mathrm{R_b}$$

Using the natural frequency formula, the damping ratio will be calculated from the middle term in the denominator of the closed-loop transfer function,

$$2\zeta\omega_n = 2\zeta\sqrt{\frac{\mathrm{K_D K_O}}{\tau_2}} = \left(\frac{1}{\tau_2}+\frac{\mathrm{K_D K_O}\tau_1}{\tau_2}\right)$$

Dividing both sides by the natural frequency leaves,

$$2\zeta = \sqrt{\frac{1}{\tau_2^2}\frac{\tau_2}{\mathrm{K_D K_O}}} + \sqrt{\frac{(\mathrm{K_D K_O})^2\tau_1^2}{\tau_2^2}\frac{\tau_2}{\mathrm{K_D K_O}}} = \sqrt{\frac{1}{\mathrm{K_D K_O}\tau_2}} + \sqrt{\frac{\mathrm{K_D K_O}\tau_1^2}{\tau_2}} \approx \omega_n\tau_1$$

Solving for ζ,

$$\zeta \approx \frac{\omega_n\tau_1}{2} = \frac{\omega_n \mathrm{R_1 C}}{2}$$

Therefore, with the pole-zero loop filter, the natural frequency and damping ratio will be determined by two different time constants giving students more control over their design.

Moreover, the natural frequency specification constrains the time constant $\mathrm{R_b C}$,

$$\omega_n = \omega_m = 10^4\pi \approx \sqrt{\frac{\mathrm{K_D K_O}}{\mathrm{R_b C}}}$$

Solving for the time constant $\mathrm{R_b C}$,

$$\mathrm{R_b C} = \frac{\mathrm{K_D K_O}}{\omega_n^2} \approx 1.55 \times 10^{-3}\,\mathrm{sec.}$$

Since the built-in resistor $\mathrm{R_b}$ is known to be 3.6kΩ, the filter capacitor C must be approximately 0.47μF. Similarly, the damping ratio specification constrains the time constant $\mathrm{R_1C}$:

$$\zeta = \frac{\sqrt{2}}{2} \approx \frac{\omega_n \mathrm{R_1C}}{2}$$

Solving for $\mathrm{R_1C}$ gives

$$\mathrm{R_1C} = \frac{2\zeta}{\omega_n} \approx 4.50 \times 10^{-5}\,\mathrm{sec.}$$

With the filter capacitor already chosen in order to meet the natural frequency specification, $\mathrm{R_1}$ must be approximately 96Ω.

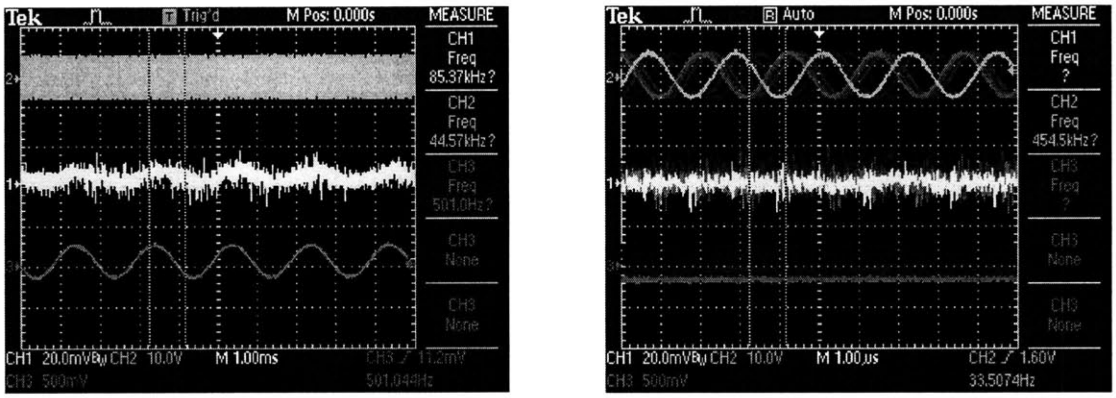
Lastly, the natural frequency and damping ratio parameters can be used to determine the maximum phase error. Students will use a graph plotting the magnitude of $\frac{\phi_e}{\frac{\Delta\omega}{\omega_n}}$ versus $\frac{\omega_m}{\omega_n}$ parameterized by ζ [13]. With the natural frequency of the loop and the modulating frequency assumed to be equal (in the worst case) and a damping ratio of $\frac{\sqrt{2}}{2}$, the resulting phase error ratio will be:

$$\frac{\phi_e}{\frac{\Delta\omega}{\omega_n}} \approx 0.702$$

Since $\Delta\omega$ and the natural frequency are also equal by the design specifications

$$\frac{\phi_e}{\frac{\Delta\omega}{\omega_n}} = \phi_e \approx 0.702 < \frac{\pi}{2}$$

With this last specification met, the FM demodulator design is complete. Students will be able to test their FM demodulator using signals originating from the FM modulator circuit and transmission system constructed in Chapter 3, received by the superheterodyne receiver system constructed in Chapter 4. An oscilloscope trace of the FM demodulator operating on an FM signal is shown in Figure 5-12.



(a) Closeup view of the demodulated signal (middle trace). The same signal is shown below it, amplified by 100. The top trace is the faster carrier signal, used as input to the FM demodulation circuit from Figure 5-11.

(b) Same capture as Figure 5-12(a), but with different time scale showing the 455 kilohertz carrier signal.

Figure 5-12: Oscilloscope captures of the LM565 acting as an FM demodulator.

Chapter 6

Lab 4: DSSS Broadcast Techniques Using CDMA

The exercises in the fourth laboratory assignment focus on DSSS (Direct Sequence Spread Spectrum) broadcast techniques. Specifically, students construct a CDMA (Code Division Multiple Access) communication system and explore topics related to signal orthogonality and autocorrelation. Signal orthogonality and autocorrelation are at the heart of all modern CDMA systems including cellular telephones and IEEE standard 802.11-based technology. Students begin by exploring these signal processing concepts in MATLAB to help in visualizing the process of spreading a signal and despreading it as well as synchronizing the transmission of these signals.

The core CDMA concepts built-up in the MATLAB exercises will then yield to an actual implementation of a CDMA system in hardware using Microchip PIC microcontrollers. One PIC will be responsible for spreading and transmitting a message over a wire and the other will be responsible for receiving and decoding the message. These exercises involve a translation of the various MATLAB routines into the C programming language including adapting several data structures and spreading techniques for the microcontroller environment.

Once this wired solution is in place the next task for the student is to make the system work wirelessly over a radio link. The first step towards this end is the introduction of extremely reliable asynchronous communication methods using a serial communication interface once again over a wire. This interface will be built as a layer on top of the CDMA transmission scheme in the previous exercises. The synchronization techniques discussed in the earlier MATLAB exercises will continue to be reinforced in this part of the assignment with discussion on the positive and negative aspects of synchronizing two geographically separate systems.

Finally, after asynchronous data transfer has been established, the system will go truly wireless. Students will use the same FM radio transmission system constructed in labs 1 through 3 to broadcast their CDMA data from one PIC to the other. With this final communication layer in place, students can explore topics related to end-to-end communication and reliability.

6.1 DSSS Overview

Modern wireless communication systems rely on their ability to give multiple users access to the same physical medium. This medium is used to convey and receive all of the messages on a network. In order to allow more than one user to communicate at a time, various multiple-access methods are used. These techniques include FDMA (Frequency Division Multiple Access), TDMA (Time Division Multiple Access), CDMA, and hybrids of the previous three techniques.

Moreover, in terms of the wireless communication system discussed in chapters 3 and 4, an FDMA system is not applicable. FDMA relies on giving different users individual carrier frequencies, which is an anathema to the single-carrier system developed previously. TDMA uses one carrier frequency and gives users access to the communication medium at specified time intervals. CDMA also employs a single carrier, but allows all users to communicate simultaneously by combining each message with a code unique to each individual.

Several multiple-access implementations are limited in their effectiveness to communicate by noise within the communication medium. For example, high-power noise at the system carrier frequency may distort or predominate over the broadcast signal of a modern cellular telephone system. To add resilience to noise, spread-spectrum techniques were developed.

The idea behind spread-spectrum communication is to take a data signal of narrow bandwidth and spread it over a larger bandwidth. With the data now spread over a larger set of frequencies, a communication system is less susceptible to noise at any one particular frequency. Two spread-spectrum techniques are commonly used: FHSS (Frequency-Hopping Spread Spectrum), and DSSS (Direct Sequence Spread Spectrum). FHSS techniques periodically switch the signal carrier frequency, often switching to frequencies chosen in a pseudo-random fashion. DSSS techniques spread a data signal over a wider set of frequencies prior to that signal being used to modulate a carrier.

Consequently, both spread-spectrum techniques use a method of CDMA to spread the bandwidth of a data signal. Therefore multiple access to a medium (e.g. a carrier signal) and resilience to noise (through spreading the spectrum of that carrier) can be applied to the same communication system. Both FHSS and DSSS systems use a code to spread the bandwidth of a signal prior to broadcasting it and then despread the received signal once it's been received, as shown in Figure 6-1. The spreading code is used differently based upon the spread-spectrum method used (FHSS or DSSS).

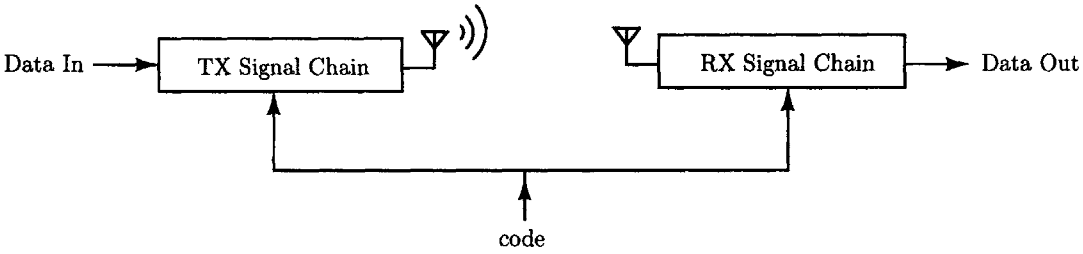


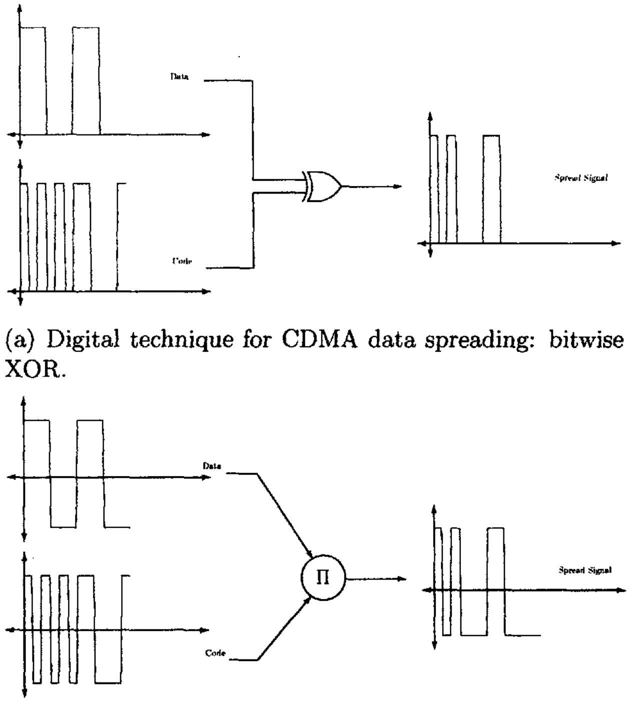
Figure 6-1: Block diagram of a typical spread-spectrum communication system.

6.1.1 Spread-Spectrum Implementations

Several commercial implementations of spread-spectrum communication systems use FHSS and DSSS. The popular Bluetooth® communication standard employs CDMA via an FHSS implementation (sometimes called FH-CDMA). The IS-95 and CDMA2000 cellular telephone standards use DSSS-based CDMA. In this laboratory assignment, students will explore CDMA based upon the ideas in IS-95 and CDMA2000. Students will implement CDMA by having a code operate directly on a data signal (i.e. DSSS) as shown in Figure 6-2.

Spreading Antipodal and Digital Signals

Two mathematical representations exist for implementing and analyzing DSSS-based CDMA. The first method employs digital representations of the data and code. The code, or "chip-



(a) Digital technique for CDMA data spreading: bitwise XOR.

(b) Antipodal technique for CDMA data spreading: multiplication.

Figure 6-2: Digital and antipodal methods for spreading a data signal with a code or chipping sequence.

ping sequence" has a much higher data rate than the data to be spread. Both the data and chipping sequence are represented by signals that have values of either 0 or 1. When the data signal and the code signal are XOR'd together, the resulting signal has a larger frequency spectrum than the original data signal. An example of this process is shown in Figure 6-2(a).

The second method employs antipodal representations of the data and code as shown in Figure 6-2(b). Antipodal signals take values of either 1 or -1. The code signal is again at a higher rate than the data signal. In this case, the data signal is multiplied by the chipping sequence to create the spread signal. After the multiplication, the spread signal contains frequency content over a larger band than the original data signal.

Moreover, these two representations are mathematically equivalent. That is, performing an XOR on two digital signals is the same as multiplying two antipodal signals. One representation can be converted to the other using the equivalence table, Table 6.1. For example, if all of the 1's in a digital signal were replaced with -1's and all the 0's replaced with 1's, the digital signal would be converted to its antipodal representation. This new signal could then be spread using the antipodal multiplication described in Figure 6-2(b). Likewise, an antipodal signal could be converted to a digital signal using the opposite approach (replacing -1's with 1's and 1's with 0's) and then spread using the digital approach of Figure 6-2(a).

Table 6.1: Digital values and their corresponding antipodal equivalent values. A handy table for converting digital signals to their antipodal equivalents.

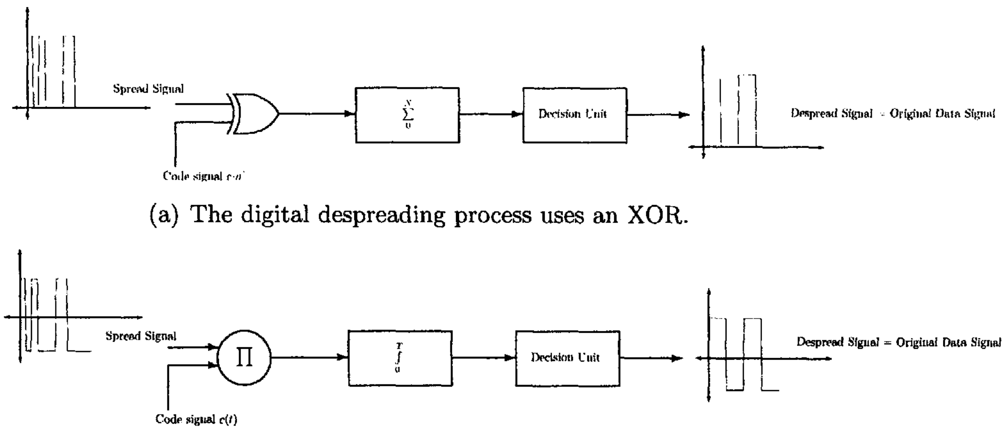
Digital	Antipodal
0	1
1	-1

Despreading Antipodal and Digital Signals

The despreading process involves performing the spreading process from Figure 6-2 again, but with an added summation step. For a single-user system, the digital and antipodal despreading processes are described in the block diagrams of Figure 6-3. The spread signal in either the antipodal or digital system is multiplied or XOR'd as appropriate. For antipodal signals this resulting signal is integrated over the length of the code signal $c(t)$. The length of $c(t)$ is labeled T in Figure 6-3(b). For digital signals, the resulting signal is summed over the length of the code signal $c[n]$ which is labeled N in Figure 6-3(a). After integration or summation, a decision unit looks at the result and outputs the appropriate antipodal or digital value. If the integral/sum is below a certain threshold value the decision unit sends an antipodal or digital "low" value per Table 6.1. If the integral/sum is above the threshold, it sends an antipodal or digital "high". This process accurately recreates the user's original data signal.

Multiple Access and Code Orthogonality

Multi-user CDMA systems employ a modified approach to the single-user system described above. Figure 6-4 shows an example of the multi-user spreading and despreading process using antipodal signals. In this 2-person example, each user spreads his or her data, $y_n(t)$, with a particular code, $c_n(t)$. The two resulting spread signals are added together to create



(a) The digital despreading process uses an XOR.

(b) The antipodal despreading process uses an analog multiplier.

Figure 6-3: Block diagram of the digital and antipodal despreading processes.

an aggregate signal. The aggregate signal, $a(t)$, is

$$a(t) = y_1(t)c_1(t) + y_2(t)c_2(t)$$

The combined information of both users in the aggregate signal presents a problem for the despreading process previously discussed in the single-user context. In the multi-user system, the same aggregate signal is despread by both users' despreading systems. That is, the information conveyed in $a(t)$ by the first user (namely $y_1(t)c_1(t)$) will be used as part of the second user's despreading process. This additional information will lead to incorrect despreading of the second user's data and vice-versa.

In order to counteract this problem, orthogonal codes can be used by both users to spread their data. Assuming codes $c_1(t)$ and $c_2(t)$ are orthogonal in Figure 6-4, then the output of the first user's integration block will be

$$s_1(t) = \frac{1}{T_1}\int_0^{T_1} a(t)c_1(t)dt$$

Inserting the equation for $a(t)$ gives

$$\frac{1}{T_1}\int_0^{T_1} \left(y_1(t)c_1(t) + y_2(t)c_2(t)\right)c_1(t)dt$$

Expanding the multiplication from above,

$$\frac{1}{T_1}\left(\int_0^{T_1} y_1(t)c_1(t)c_1(t)dt + \int_0^{T_1} y_2(t)c_2(t)c_1(t)dt\right)$$

Since $c_1(t)$ and $c_2(t)$ are orthogonal, the average value of their multiplication is zero, that is

$$\frac{1}{T_1}\int_0^{T_1} c_1(t)c_2(t)dt = \frac{1}{T_2}\int_0^{T_2} c_1(t)c_2(t)dt = 0$$

Therefore, the integral becomes

$$s_1(t) = \frac{1}{T_1}\int_0^{T_1} y_1(t)c_1(t)c_1(t)dt$$

The orthogonality of the two codes eliminates the unwanted information from the integration. This property of the codes allows multi-user data to remain in the aggregate data stream but keeps users from despreading all but their own data. Therefore, to allow multiple access to the CDMA communication medium, all user codes must be orthogonal to each other.

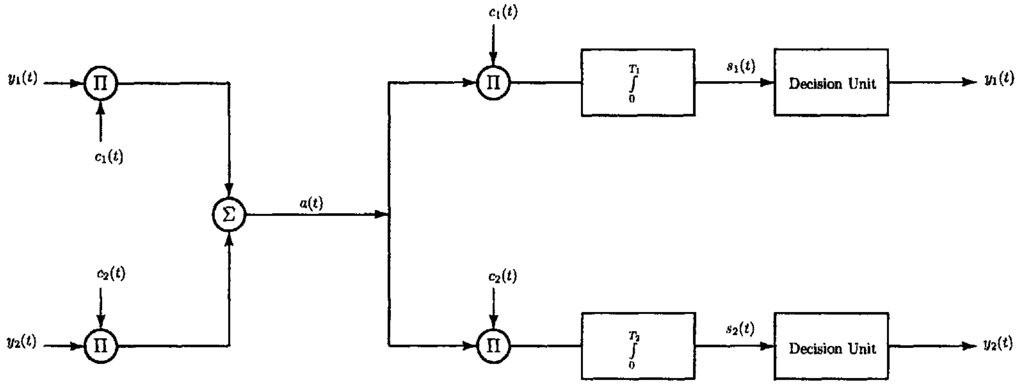


Figure 6-4: Antipodal multi-user CDMA system. Both the spreading and despreading processes are shown.

Using an aggregate signal also presents a different problem for digitally-spread systems. Even with orthogonal digital codes, the aggregate signal created from the sum of two or more digitally-spread signals cannot be digitally despread. The digital XOR that would take the place of the antipodal multiplier in Figure 6-4 cannot process $a(t)$ since it is not a digital signal. Therefore, in a multi-user system, the despreading process must be performed using antipodal methods.

Consequently, a multi-user digital system will have to convert its signals to their antipodal equivalents at some time prior to despreading. This can be done prior to creating the aggregate signal $a(t)$ as shown in Figure 6-5. Two digital signals are spread using a bitwise XOR. These two signals are then converted to their antipodal equivalents. The same despreading process from Figure 6-4 can now be used. The despread signals are then converted back to their digital equivalents. This method allows the digital, single-user tools discussed previously to be used in a multi-user context. Without the addition of antipodal tools, multi-user digital systems cannot be used.

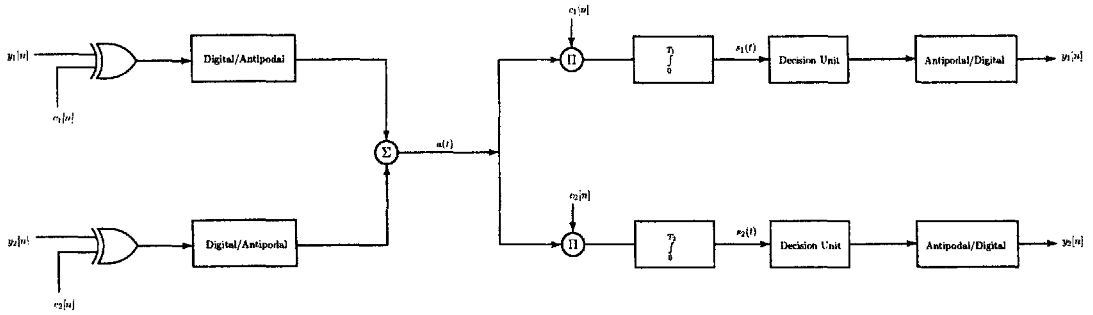


Figure 6-5: Digital multi-user CDMA system. Note the use of antipodal methods for despreading of the aggregate signal $a(t)$.

Therefore, to create a multi-user CDMA communication system, code orthogonality is essential and digital systems must employ antipodal despreading methods. Code orthogonality eliminates the problem of using an aggregate data signal where every user's data is included in the despreading process. Antipodal methods allow digital systems to work with the non-digital aggregate signal $a(t)$ described in both Figure 6-4 and Figure 6-5.

6.1.2 The IS-95 Standard and Its Lab 4 Implementation

Two of the most widely used CDMA multi-user implementations are the IS-95 and CDMA2000 cellular telephone standards. IS-95 is in wide use as a 2G (second generation) network while CDMA2000 is a burgeoning standard used in the increasingly popular 3G (third generation) networks. Students will implement a simplified version of the IS-95 standard using 2 PIC microcontrollers.

IS-95 CDMA Communication

IS-95 (Interim Standard 95) uses the same principles of spreading and despreading signals discussed in Section 6.1.1. However because it is the basis for a cellular telephone service, IS-95 adds several additional elements to the basic spreading/despreading communication method of Figure 6-4. Most notably, customers who send a CDMA signal (i.e. spread data with their own code and transmit) must communicate with a BTS (Base Transceiver Station) prior to having their transmission reach other users who can then despread incoming call data.

This BTS communication system is described in Figure 6-6. For transmission, several cellular phone customers use their hand-held units to spread voice data with their own code. The codes used are a set of orthogonal codes that resemble pseudo-random noise (known as PN sequences). After spreading their data, each mobile unit transmits its data to the BTS. This is known as reverse channel communication.

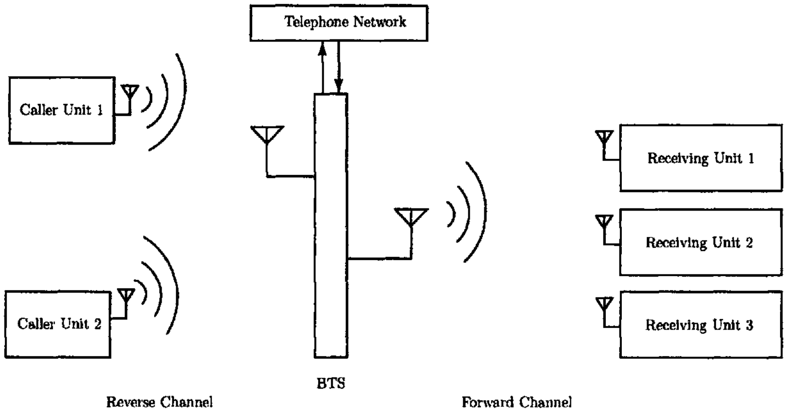


Figure 6-6: IS-95 CDMA communication system using a BTS. The communication chain between the caller units and the BTS is known as the reverse channel while the chain between the BTS and the receiving units is called the forward channel.

The BTS receives these reverse channel transmissions and despreads each callers' data using each individuals' code. This despread data is sent to the telephone company which reroutes the data to the intended receiving cellular phone. The BTS closest to the intended call receiver then spreads the call data with the receiver's code. Prior to transmission, another spreading process is performed on the (already spread) data before it is transmitted. This additional spreading process is added to allow adjacent BTS's to use the same set of broadcast frequencies. Therefore, after the second spreading process, the BTS transmits the call data to the receiver's hand-held unit. The receiving unit can then despread the BTS transmission and gain access to the voice data. This is known as forward channel communication.

In contrast to the reverse channel, the forward channel portion of IS-95 communication uses a different code for spreading and despreading data. While the reverse channel used PN sequences, the forward channel uses a set of orthogonal codes called Walsh codes. Walsh codes allow for more users to occupy the same carrier frequency but these users must be synchronized. Up to 64 users can receive data on one carrier frequency using Walsh codes. The asynchronous nature of the reverse channel therefore does not allow for the use of Walsh codes.

Lab 4 IS-95 Implementation

In this laboratory assignment, students will implement a modified version of the IS-95 forward channel component. The BTS role in the forward channel will be implemented using a PIC microcontroller. In software, the PIC will have several messages that will be spread using orthogonal codes. These will then be added together and broadcast using the FM transmission system from Chapter 3. Although the spreading of multiple signals in software does not fully utilize the noise-rejection properties of the spread spectrum system, it does emphasize the multi-user aspects of CDMA allowing multiple data streams to be broadcast at the same time.

Furthermore, the superheterodyne reception system from Chapter 4 will receive the broadcast signal and send the spread data to another PIC. This second PIC will despread the received data to gain access to a specific message transmitted from the first PIC. The entire forward channel system that will be implemented is shown in Figure 6-7. The details of the entire IS-95 implementation are discussed in Section 6.3.

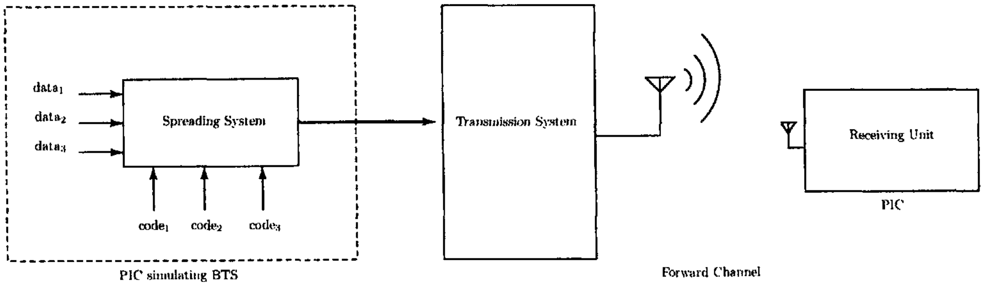


Figure 6-7: IS-95 forward channel version to be implemented in Lab 4.

6.1.3 Synchronization

A vital component to any CDMA implementation and notably the IS-95-based system that will be constructed by students is the need for synchronization. Two kinds of synchronization are necessary to ensure that data is properly received and despread so as to faithfully reproduce the original data streams that were broadcast. The first type of synchronization is bit synchronization which will allow transmission and reception to occur without data loss. The second is frame synchronization which wil allow receiving units to know where the beginning of the despread data should begin.

Bit synchronization is necessary to prevent data loss in the transmission process. It will allow the receiving unit to know when a transmission begins and ends. If a receiving unit were to lose this type of synchronization, part of the spread message may go missing resulting in improper despreading. In order to achieve bit synchronization, an asynchronous serial communications link will be used. The details of this communication link will be discussed in Section 6.3.

Furthermore, a portion of the spread data will contain vital information for frame synchronization. One of the codes to be used for spreading will be a Barker code. Barker codes are a set of sequences that vary in length but all have a high autocorrelation [15]. The inner product of a Barker code with itself is maximized, while the inner product of a Barker code with a shifted version of itself is minimal (in most cases equal to zero). This autocorrelation (represented as Φ_{bb}) property is

$$\Phi_{bb}[m] = \sum_{0}^{N} b[n]b[n+m] \quad \text{and} \quad \Phi_{bb}[0] \gg \Phi_{bb}[m] \quad \text{for} \quad m > 0$$

For frame synchronization, this behavior of the Barker codes can be exploited to determine if the despreading process is out of sync. By seeing if the Barker code used to spread a signal is aligned with the same Barker code used to despread it, loss of synchronization can be detected and corrected. An example of the consequences of losing frame synchronization is shown in Figure 6-8. Further details of frame synchronization using Barker codes are discussed in Section 6.3.

Thus, using the spreading and despreading framework described in Section 6.1.1 students will construct a modified version of the IS-95 standard. The communication system to be constructed will mimic the base station and receiving units of the IS-95 forward channel as

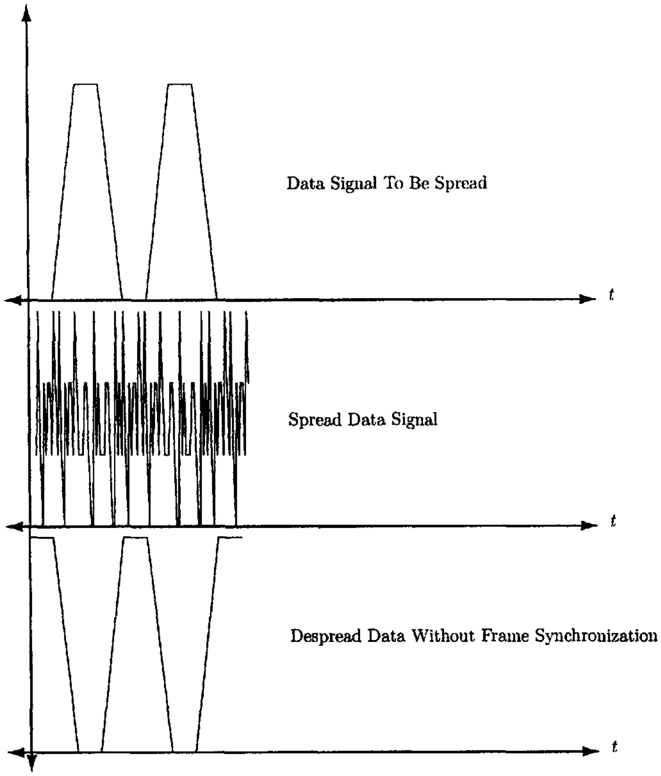


Figure 6-8: This plot demonstrates the effect of despreading without proper frame synchronization, the data signal to be spread is shown at the top. After spreading, the signal in the middle is created. Despreading without frame synchronization produces the inverted version of the original signal shown in the plot at the bottom.

described in Section 6.1.2. Students will explore the concepts of multiple-access using this system and gain further understanding of the key synchronization issues inherent to most CDMA systems.

6.2 Introductory MATLAB Exercises

Prior to constructing the CDMA communication system based on IS-95, students will explore several concepts related to CDMA using MATLAB. Since students are expected to already have a strong familiarity with MATLABs core software and several toolboxes, it can become a strong aid in teaching the vital concepts of the system. Thus, with the several coding exercises described in this section, students should be able to gain significant familiarity with the programming concepts necessary to implement CDMA. Later, it will be easy for students to export these MATLAB routines to the C programming environment of the PIC16F628.

6.2.1 Antipodal Versus Digital Signals

In this first exercise, students explore spreading antipodal and digital signals. Students begin by creating two signals in MATLAB, one that is antipodal and one that is digital. Each signal will be spread according to a corresponding antipodal or digital procedure. The end goal of this exercise is to show that the two procedures are equivalent. This equivalence must be reinforced as the PIC system to be constructed in later exercises relies upon this relationship between the two techniques.

Two signals (one antipodal, the other digital) are shown in Listing 6.1. Prior to spreading each of these signals, students should look at a truth table for the **xor** function and compare it to the four possible outcomes of multiplying the numbers 1 and -1. The two functions are equivalent when a digital 0 is replaced by an antipodal 1 and a digital 1 is replaced by an antipodal -1 as described in Table 6.2.

```
antipodal=[1 -1 1]; # antipodal signal
digital=[0 1 0];    # digital signal
a_code=[-1 1 -1 -1 -1 1 1 1 -1 1 1]; # antipodal chipping sequence
d_code=[1 0 1 1 1 0 0 0 1 0 0]; # digital chipping sequence
a_spread=[]; # holder for antipodal spread sequence
d_spread=[]; # holder for digital spread sequence

# Spread the antipodal signal
for i=1:3
   for j=1:11
     temp=antipodal(i)*a_code(j);
     a_spread=horzcat(a_spread,temp);
   end
end

# Spread the digital signal
for i=1:3
   for j=1:11
     temp=xor(digital(i),d_code(j));
     d_spread=horzcat(d_spread,temp);
   end
end
```

Listing 6.1: Sample MATLAB code for spreading antipodal and digital signals

Table 6.2: Digital operations and their equivalent antipodal operations.

Digital	Antipodal
$1 \oplus 1 = 0$	$-1 \times -1 = 1$
$1 \oplus 0 = 1$	$-1 \times 1 = -1$
$0 \oplus 0 = 0$	$1 \times 1 = 1$

The antipodal data stream is spread using multiplication in the first for loop. Similarly, the digital stream is spread using a bitwise **xor** in the second for loop. Students should be able to compare the two resulting data streams and use Table 6.2 to convert a_spread into a digital data stream. The converted a_spread and its partner signal d_spread should be identical.

Once the spreading process is complete, students can write a routine to recover the signals antipodal and digital from a_spread and d_spread respectively. The despreading process is much like the spreading process, with an added summation as discussed in Section 6.1. This new routine cannot function without knowledge of the code used to spread a signal (making it impossible to despread it). Therefore, the new routine described in Listing 6.2 already has the codes a_code and d_code built into it. To despread the antipodal signal

```
# Despread the antipodal stream
for j=1:11
      despread1=horzcat(despread1, a_spread(j)*a_code(j));
      despread2=horzcat(despread2, a_spread(j+11)*a_code(j));
      despread3=horzcat(despread3, a_spread(j+22)*a_code(j));
end
a_1=sum(despread1)/10
a_2=sum(despread2)/10
a_3=sum(despread3)/10
# Despread the digital stream
for j=1:11
      despread1=horzcat(despread1, xor(d_spread(j),d_code(j)));
      despread2=horzcat(despread2, xor(d_spread(j+11),d_code(j)));
      despread3=horzcat(despread3, xor(d_spread(j+22),d_code(j)));
end
d_1=sum(despread1)/11
d_2=sum(despread2)/11
d_3=sum(despread3)/11
```

Listing 6.2: Sample MATLAB code for despreading antipodal and digital spread data streams.

we break up a_spread into sections that are of equal length to the code. Each section will correspond to one "bit" of the despread message. This property comes from knowing that our CDMA system creates a spread message that is MN bits long where M is the size of the original signal (in this case antipodal and digital) and N is the size of the code used to spread it.

In the first for loop, the antipodal spread stream is broken up into these sections and each bit is multiplied by its corresponding bit in the code (i.e. the first bit of each section is multiplied by the first bit of the code, and so on). Once the loop has completed, each of these despread sections: despreadl, despread2, and despread3 should have 11 components in them. These 11 components are added together using the sum command. This process emulates the despreading process for antipodal signals from Section 6.1.1.

The summation of each sections' 11 elements will either be a negative number or a positive number corresponding to an antipodal "-1" bit or "1" bit respectively. The magnitude of this number in the example listing is 10, dividing each bit by 10 gives the corresponding antipodal bit. Therefore, antipodal bits a_1, a_2, and a_3 should be the same as the three bits in antipodal. A similar process is used for despreading the digital stream d_spread except it uses the xor function instead of multiplication. The result however, should be the same, with digital bits d_1, d_2, and d_3 being identical to the bits in digital from Listing 6.1.

While these simple spreading and despreading procedures help to introduce students to the idea of spread spectrum techniques they do not emphasize the multi-user aspects of the CDMA system. In the next MATLAB routine, students will explore adding multiple spread signals together to create an aggregate data signal. They will then have to create pieces of code that work with this aggregate signal to reproduce a given set of data. This exercise reinforces the multiple-access discussion of Section 6.1.1.

A multiple-access system involving two users can be constructed using MATLAB with minimal modification to the previous two routines created in this section. In the next exercise, instead of spreading just one signal like digital with one code (d_code), several signals will be spread with their own codes. However, the codes that are used for spreading cannot be chosen at random, they must be orthogonal to each other. Without code orthogonality, the original signals cannot be despread from the aggregate signal [15].

The MATLAB example code in Listing 6.3 implements the spreading of two different digital signals data_a, and data_b. The signal data_a will be spread with the code code_a, and the signal data_b will be spread with the code code_b. Both codes are chosen such that their inner product (code_a · code_b) is equal to zero. Each code is spread in the for loop. The resulting two spread signals are then added together to create an aggregate signal. This new aggregate signal contains all of the information from the previous two spread signals, but is now inaccessible to the digital despreading tools developed in Listing 6.2 (essentially becoming an analog signal). Those tools relied on the spread data being binary in value (either 0 or 1), the new aggregate signal has values of 0, 1, or 2. Using a bitwise xor function for despreading is no longer an option. However, because of the equivalence of the antipodal

```
# Two digital signals
data_a=[0 1 0];
data_b=[0 0 0];

# Two orthogonal codes, one for each signal
code_a=[1 0 1 1 1 0 0 0 1 0 0];
code_b=[0 1 0 0 0 1 1 1 0 1 1];


# Place holders for the two spread data streams
spread_a=[];
spread_b=[];

# Spread data_a and data_b with code_a and code_b
for i=1:3
  for j=1:11
    temp_a=xor(data_a(i),code_a(j));
    temp_b=xor(data_b(i),code_b(j));
    spread_a=horzcat(spread_a,temp1);
    spread_b=horzcat(spread_b,temp2);
  end
end

# Create the aggregate signal
aggregate=spread_a+spread_b;

# Convert ‘‘digital’’ aggregate to ‘‘antipodal’’ aggregate
lookup=[2 0 -2];
m=length(aggregate);

a_sum=[]; # a place holder for the ‘‘antipodal’’ aggregate signal
for i=1:m
  a_sum(i)=lookup(aggregate(i)+1);
end
```

Listing 6.3: MATLAB code for spreading two digital signals and creating an aggregate spread signal

and digital methods, the antipodal tools *can* be used to despread the signal.

Consequently, one last transformation of the aggregate signal is required before the antipodal tools can be used. This requirement stems from using values of -1 or 1 for antipodal signals. If the signals spread_a and spread_b were antipodal, the aggregate stream resulting from their sum would take values of 2, 0, or -2 as opposed to the current 0, 1, or 2 in Listing 6.3.

Instead of going back and re-spreading the signals data_a and data_b with antipodal tools, we can quickly convert the aggregate data using a lookup table. The benefits of this technique over re-spreading are two-fold. First, using digital tools to spread and antipodal tools to despread reinforces the idea that the digital and antipodal representations of these signals are equivalent. Second, the digital tools for spreading create rich, high-performance routines for use with microcontrollers, since they rely on digital operations. These routines are good teaching tools for learning about microcontroller systems. Thus, they will be of great benefit in the exercises discussed in Section 6.3.

The conversion routine is found at the end of Listing 6.3. The array called lookup contains the 3 possible values for an antipodal aggregate signal. The value of the digital aggregate signal is used as an index into this array. The value selected from the lookup array corresponds to the antipodal equivalent of the digital aggregate value. Using a **for** loop, the entire digital aggregate signal is converted to the antipodal aggregate signal a_sum. This new aggregate signal can be despread exactly as in Listing 6.2. The signal data_a can be recovered by using the despreading routine in Listing 6.2 replacing the code used with code_a. Similarly, the signal data_b can be despread using the same routine but replacing the code used with code_b.

In summary, the three listings provided in this section introduce the concept of representing signals in their antipodal and digital forms, spreading them individually using the CDMA method described in [15], and then creating an aggregate signal appropriate for transmission. This aggregate signal cannot be despread with the digital methods in Listing 6.2 because it is no longer in a binary format. As a result, it must be converted to an antipodal aggregate and then despread using antipodal means. Antipodal tools work because they do not rely on assumptions about the binary format of the aggregate spread data.

6.2.2 Barker Codes and Synchronization

Synchronization is a critical component for despreading a CDMA signal. When receiving data for despreading, it is assumed that it is in order, with the data spread by the first bit of the code received first. However, many CDMA systems have delays and shifts associated with their implementations. A system must be created to account for these sort of "frame shifts" and still create proper despread output as. One such system employs the use of Barker codes discussed in Section 6.1.3.

To demonstrate the "self-syncing" ability of the Barker code, students will experiment with one such sequence using MATLAB. The code in Listing 6.4 creates a vector containing the 11 bit Barker sequence barker and takes the inner product of that signal with itself. The Barker code barker is dotted with itself using the **dot** command. The output inner is 5 for a perfectly aligned dot product. However, the next several lines shift the Barker code by one and take the inner product once again, this time of the Barker code with its shifted version. This new inner product is 2, significantly less than the aligned dot product.

```
# The 11 bit Barker code
barker=[0 1 0 0 1 0 0 0 1 1 1];

# Find its inner product with itself
inner=dot(barker,barker)

# Shift the Barker code by one bit, with rotation
barker_shift=shift(barker,1);

# Now take the inner product of the Barker code
# with its shifter version, inner should be a lot smaller
inner=dot(barker,barker_shift)
```

Listing 6.4: MATLAB code demonstrating the high autocorrelation of the Barker code.

This property also holds true for signals spread with a Barker code using CDMA. For example, the MATLAB code in Listing 6.5 spreads a signal of all 1's with the 11 bit Barker code used previously in Listing 6.4. This spread signal is then despread with a properly aligned version of the Barker code. The sum of this despread signal is 5. The next several lines proceed to despread the signal with a shifted (misaligned) version of the Barker code. The resulting sum of this despread signal is significantly reduced at 2.3.

This Barker-spreading system can be used to detect when the code used to despread a signal is aligned correctly. If we include a signal of all 1's spread with a Barker code to the aggregate signal created in Listing 6.3, we can add synchronization detection at the point

```
# 11 bit Barker code used for spreading
barker=[0 1 0 0 1 0 0 0 1 1 1];

# Data signal to be spread
data=[1 1 1 1 1];

# Place holder for the spread data
spread =[];

# Spread data with barker
for i=1:5
  for j=1:11
    temp=xor(data(i),barker(j));
    spread=horzcat(spread ,temp);
    end
end

despread1 =[];
despread2 =[];
despread3 =[];
despread4 =[];
despread5 =[];
# Despread the digital stream
for j=1:11
      despread1=horzcat(despread1 , xor(spread(j),barker(j)));
      despread2=horzcat(despread2 , xor(spread(j+11),barker(j)));
      despread3=horzcat(despread3 , xor(spread(j+22),barker(j)));
      despread4=horzcat(despread4 , xor(spread(j+33),barker(j)));
      despread5=horzcat(despread5 , xor(spread(j+44),barker(j)));
end

despread_total=horzcat((sum(despread1)/11),(sum(despread2)/11), ...
(sum(despread3)/11),(sum(despread4)/11),(sum(despread5)/11));

sum(despread_total)

# Now shift the Barker code by one
barker_shift=shift(barker ,1);

despread1 =[];
despread2 =[];
despread3 =[];
despread4 =[];
despread5 =[];

# Despread the digital stream again
for j=1:11
      despread1=horzcat(despread1 , xor(spread(j),barker_shift(j)));
      despread2=horzcat(despread2 , xor(spread(j+11),barker_shift(j)));
      despread3=horzcat(despread3 , xor(spread(j+22),barker_shift(j)));
      despread4=horzcat(despread4 , xor(spread(j+33),barker_shift(j)));
      despread5=horzcat(despread5 , xor(spread(j+44),barker_shift(j)));
end

despread_total=horzcat((sum(despread1)/11),(sum(despread2)/11),...
(sum(despread3)/11),(sum(despread4)/11),(sum(despread5)/11))

sum(despread_total)
```

Listing 6.5: MATLAB code employing the the Barker code to spread and despread data.

of reception by despreading the aggregate with the same Barker code. If the Barker code at the receiving end is shifted by any amount, the expected maximum value will not be present when taking the sum of all the bits of the despread signal.

However, while this system allows for detection of synchronization problems, it does not correct for these problems. Adding one last feature to the receiving end will not only detect synchronization problems but also correct for them. If a misalignment is detected, the Barker code at the receiving end can be shifted by 1 to compensate. After shifting by 1 the synchronization detection process is started again. If synchronization is still not detected, the system will continue to shift the Barker code by one at the receiver until the system is synchronized. This correction mechanism will be explored in further detail in Section 6.3.

Therefore, Barker codes are an invaluable tool in building a reliable CDMA communication system. The high auto-correlation of these codes makes them ideal for synchronization detection and correction. Including a Barker code as one of the spreading codes in an aggregate CDMA signal introduces a measure of synchronization error detection and correction.

Moreover, all of the MATLAB routines developed as introductory material for students in this section will be the building blocks for creating a CDMA communication system in hardware. Digital and antipodal PIC code will be used in addition to the synchronization detection and correction routines developed in Section 6.2.2. Each of the components created in subsequent sections draw upon the students' understanding of the fundamental concepts of CDMA described in Section 6.1 and explored in this section.

6.3 Hardware CDMA Implementation

The next step in exploring CDMA systems is to implement one using a hardware system driven by the CDMA software routines developed in Section 6.2. The intended system takes the form of the block diagram in Figure 6-9. The software routines for spreading a signal are included in a PIC16F628 at the transmission side. The output of these routines will be changed to conform to the specifications of the PICs built-in SCI (Serial Communications Interface) and transmitted over the air to a receiver unit. This receiver unit then decodes the SCI transmission and sends the CDMA signals to a set of despreading routines running on another PIC16F628.

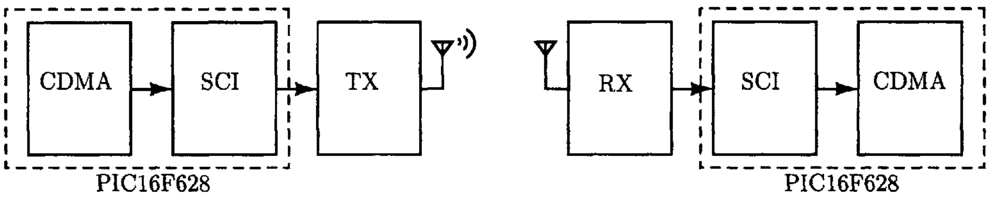


Figure 6-9: Block diagram of the CDMA hardware system.

The system will be constructed in three stages. The first set of exercises for students involves the translation of the MATLAB routines created in Section 6.2 to the PIC platform in the C programming language. After translation, both the transmission PIC and the reception PIC are made to communicate with clock and CDMA signals transmitted directly via wires. The second stage of exercises introduces SCI and the elimination of the need for a clock to ensure synchronization of reception. The last stage of exercises adds a wireless component using the transmission and reception techniques developed in Chapters 3, 4, and 5.

6.3.1 Signal and Clock Transmission/Reception

The MATLAB routines in the first section received their input signals all at once. That is, neither the spreading nor despreading routine received its input in a serialized fashion. However, the hardware transmission of even 55 bits of spread data in parallel (as in Listing 6.5) is prohibitive in terms of cost and complexity of implementation. Therefore the systems constructed in this and subsequent exercises will transmit the spread data serially.

An effect of this new serial transmission requirement is the need to notify the receiving system when a bit has been transmitted. Without knowledge of the rate of transmission, a receiving system cannot synchronize properly to the transmitter. The PIC implementation described in this section includes a clock signal as a separate output in addition to the aggregate CDMA signal described in Section 6.2. A circuit schematic of the complete "clock and signal" system is shown in Figure 6-10.

This set of exercises, starting with the CDMA transmitter, involves translating the routines of Listing 6.5 into the C programming language for use with the PIC16F628. The aggregate signal transmitted will contain three orthogonal codes: an 11-bit barker code used for CDMA synchronization as discussed in Subsection 6.2.2, and two other 11-bit codes used for transmitting data. These three codes along with the data to be transmitted are defined

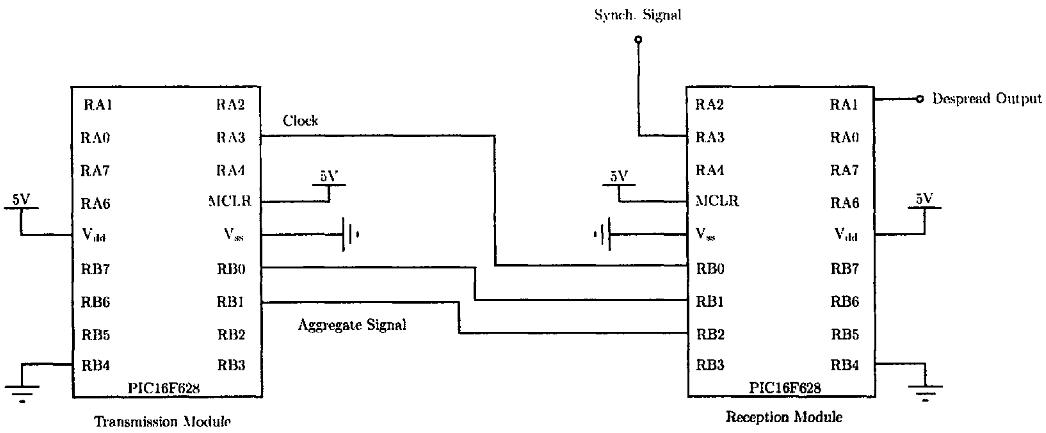


Figure 6-10: Circuit schematic of the complete clock-and-signal CDMA reception/transmission system.

first in the PIC transmission code in Listing 6.6.

The Interrupt Service Routine (ISR) contains the actual code used to spread the stored data. The code uses a timer interrupt to ensure a regular interval of transmission. The main routine sets up the timer interrupt and pins on I/O ports A and B as interrupts using the tristate registers [16]. The timer interrupt is programmed to create a clock edge on pin A3 in Figure 6-10. One value of the aggregate signal is sent on each clock edge. Since the aggregate signal can take values of 0, 1, or 2, two bits need to be transmitted per aggregate value. These bits are transmitted on pins B0 and B1. The interrupt is arbitrarily programmed to transmit the clock and one aggregate signal value (two bits in size) at approximately 31.50 kHz (using the timer register value in TMR0).

The spreading is performed the same way as the MATLAB code of Section 6.2.2 with a bitwise XOR. Each signal is spread with its corresponding code and summed together before transmission. Upon reaching the end of the code the ISR starts at the beginning again by setting the variables i and j to zero.

The CDMA system receiver has three input pins: the clock signal from the transmitter (used as an edge-triggered interrupt) on B0 and the two bits of spread aggregate data on pins B1 and B2 in Figure 6-10. The receiver has two outputs: pin A1 for the despread data and pin A3 to signal that the CDMA despreading process is synchronized. Using an edge-triggered interrupt on the clock will ensure greater accuracy in receiving valid inputs than typical I/O port polling techniques.

Like the CDMA despreading code in Listing 6.5, the receiving PIC system must translate

```
#include <pic.h>
#include <pic16f6x.h>

__CONFIG(0x3FF0);

unsigned char barker[11] = {0,1,0,0,1,0,0,0,1,1,1};
//Barker code specific data, for receiver syncing
unsigned char bdata[10] = {0,0,0,0,0,0,0,0,0,0};
unsigned char code[11] =  {1,0,1,1,1,0,0,0,1,0,0};
unsigned char orth[11] = {0,0,0,1,0,1,0,0,1,0,1};
unsigned char data[10] = {1,1,0,0,1,1,0,0,1,1}; //some data
unsigned char i,j;

void main (void) {
  TRISA=0x00;
  TRISB=0xF0;
  CMCON=0x07;
  TMR0=0x7F;
  T0IE=0x01;
  GIE=0x01;
  i=0;
  j=0;
  while (1) {
    // Just sit and wait
  }
}

void interrupt ISR (void) {
  if (T0IF){
    RA3=0x01;
    PORTB = ((data[i]^orth[j])+(data[i]^code[j])+(bdata[i]^barker[j]));
    j++;
    if (j==11) {
      j=0;
      RA0=data[i];
      i++;
      if (i==10){
        i=0;
      }
    }
    T0IF=0x00;
    TMR0=0x00;
    RA3=0x00;
  }
}
```

Listing 6.6: PIC C code for transmitting serial CDMA data using a clock signal.

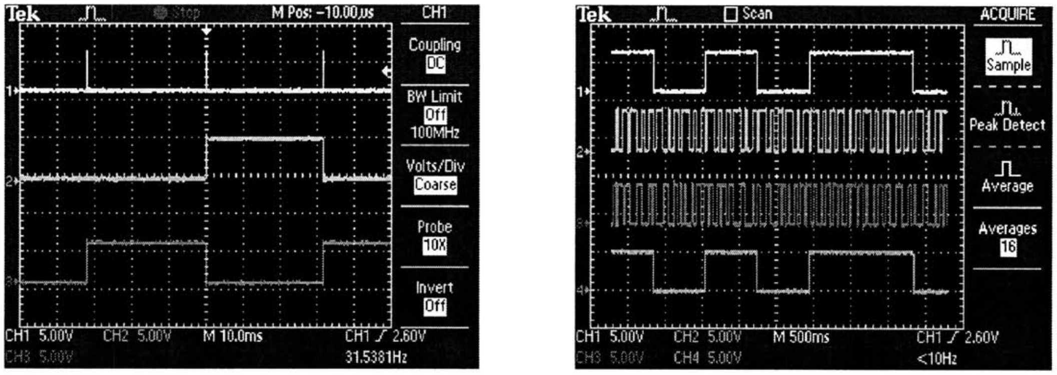
the digital aggregate signal to an antipodal one using a lookup array of values. Additionally, it must store an antipodal version of the Barker code and an antipodal version of the code it will use to despread the data signal. The C code in Appendix A.1 lists the antipodal Barker code, antipodal code, and lookup array first.

Following these variable definitions is a routine for rotating an array of values (much like the shift function in MATLAB). The code uses this routine to rotate the Barker code in order to synchronize the CDMA despreading process as discussed in Section 6.2.2. After the rotation routine is the main routine used for setting I/O pins and interrupts appropriately. Below the main routine is one that converts an antipodal signal to a binary one. This is necessary for the receiver to properly output the despread data signal digitally on an I/O pin.

Finally, the main components of the despreading process are located in the ISR. Triggered on a clock edge, the ISR reads in the aggregate value from pins B1 and B2, using the Barker code to despread the aggregate data first to ensure synchronicity. This synchronicity value, syncvalue is added to a running total called syncsum. That sum will be maximized when the copies of the Barker code at the transmitter and receiver are aligned properly as in Section 6.2.2. A check on this maximal value is done after 11 values are received (the length in bits of the Barker code data, bdata, from Listing 6.6).

If the value is at its maximum, the despreading process is deemed valid and a binary bit of despread data is sent at the output, along with a high value on pin A3 to signal synchronicity. Should the value not be maximal, pins A1 and A3 are reset to zero as well as all counting variables. Then, the Barker code is rotated by one using the rotate() function per the algorithm specified in Section 6.2.2. This process continues indefinitely, performing "on the fly" synchronicity checks and outputting valid values as long as clock edges continue to be received on pin B0. Oscilloscope captures of the entire system performing the spreading and despreading process are shown in Figure 6-11.

Therefore, this PIC-based system emulates the process of spreading several digital signals in a fashion similar to Section 6.2. These signals are then added together in software to create an aggregate signal which is then transmitted via a wire serially to a receiving circuit. The receiving circuit receives this aggregate data using a separately transmitted clock signal and despreads it. The receiver PIC system uses antipodal tools (equivalent to their digital counterparts) to properly despread the transmitted data and give a digital



(a) Clock signal (top trace) being transmitted along with two bits of aggregate data (bottom two traces).

(b) The transmitter data signal is on the top trace, followed by the two bits of aggregate data, with the reception modules despread output on the bottom trace.

Figure 6-11: The reception module despreading the aggregate data and following the transmitter's data. Subfigure 6-11(a) shows the clock signal and two aggregate bits being transmitted. Subfigure 6-11(b) shows the reception module properly outputting despread data.

despread signal on its output pin. As the majority of their work in developing the clock and signal system, students can be expected to translate the CDMA MATLAB routines into C routines.

6.3.2 Transmission/Reception Using A Serial Communications Interface

The clock and signal system described in the previous section relies heavily on the transmitted clock for proper operation. Without receiving valid rising edges on the interrupt pin B0, the reception PIC cannot properly despread the spread data. Having a dependence on a clock creates significant complexity, particularly with regards to the end goal of wireless communications. For example, the clock must be transmitted separately to the aggregate data, creating the need for possibly two superheterodyne receivers of the type described in Chapter 4. Including another such receiver in the laboratory kit would significantly increase its size and circuit complexity.

Consequently, two different design approaches emerge as alternatives to a more complex wireless transmission/reception system. One such solution is to use a clock recovery circuit using PLLs as discussed in Chapter 5. This solution is prohibitive however, due to inconsistencies in the performance of monolithic PLL ICs such as the CD4046 and the LM565. Both of these chips proved too unstable to keep the despread data consistently synchronized at the PIC output. Therefore, a second solution is to use an asynchronous method of transmission. Several serial protocols exist that transmit data in this fashion, one of which is used by the the PIC16 family of microcontrollers.

The PIC16F628 model microcontroller contains a serial communications interface known as a USART (Universal Synchronous Asynchronous Receiver Transmitter). The USART uses a serial communication protocol created to facilitate communications with peripheral chips and especially for communications with other PIC microcontrollers [16]. Therefore, the USART is uniquely suited for reliable communication between the transmission and reception modules in the hardware CDMA implementation. The circuit schematic from Figure 6-10 is now simplified to just one wire for communications between the two modules as shown in Figure 6-12.

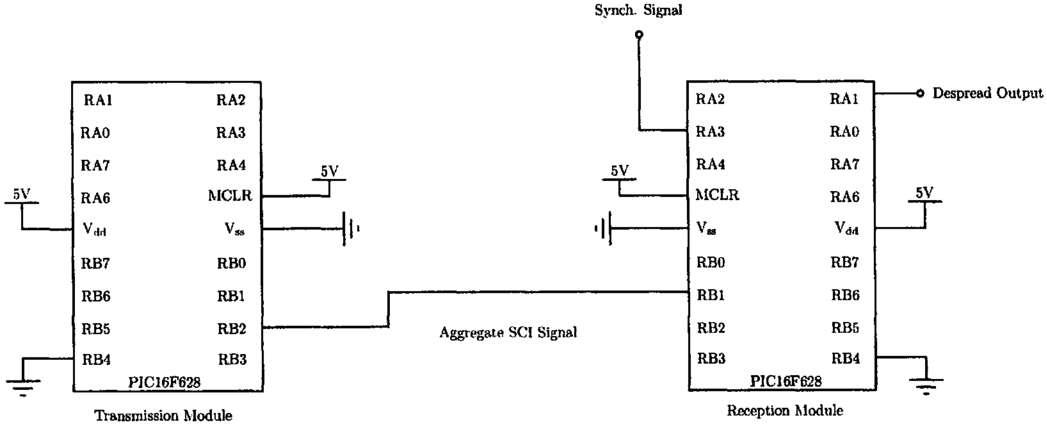


Figure 6-12: Circuit schematic of the CDMA reception/transmission system using the built-in PIC USART.

While the circuit for communications using the USART is less complex, the serial protocol itself creates some additional complexity in software. The USART transmits serial data in 8-bit bursts, it does not support the 2-bit transmission protocol used in Section 6.3.1. Working within this new 8-bit framework, 4 aggregate values could be packed into each of these new serial packets. Throughput could be significantly increased using this method. Instead of packing values, the remaining 6 bits in these serial packets will be used to increase the reliability of the communication channel.

The USART-based system performs adequately provided that the wire between the 2 PICs is not subject to significant sources of noise or other interference. However, when a wireless system is added, noise issues become increasingly important factors. Dropped packets in particular are a bane to reliable communication. The clock-and-signal system does not properly detect these types of errors. If the clock signal is received, but the 2 bits of aggregate data are lost, or full of noise, the despreading mechanism will assume its Barker code is out of sync. It will then respond by shifting its Barker code by one, but this will not correct the problem. Therefore, the additional 6 bits of the USART serial packets will be used to detect packet loss.

The new SCI PIC transmission code in Appendix A.2 incorporates the spreading method of the previous clock-and-signal system with an additional packet loss detection technique. The serial packet called packet consists of the 2 bit aggregate value, 2 zeros (used as spacer bits), and a 4 bit count as shown in Figure 6-13. The 4 bit count corresponds to the jth value of the codes used to spread each of the data signals. This value gives each packet a sequential numerical ID that indicates which portion of a spread bit is being transmitted.

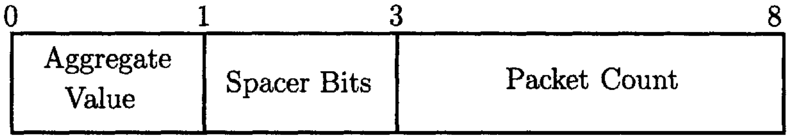


Figure 6-13: Structure of the serial transmission packet for use with the PIC16F628 USART.

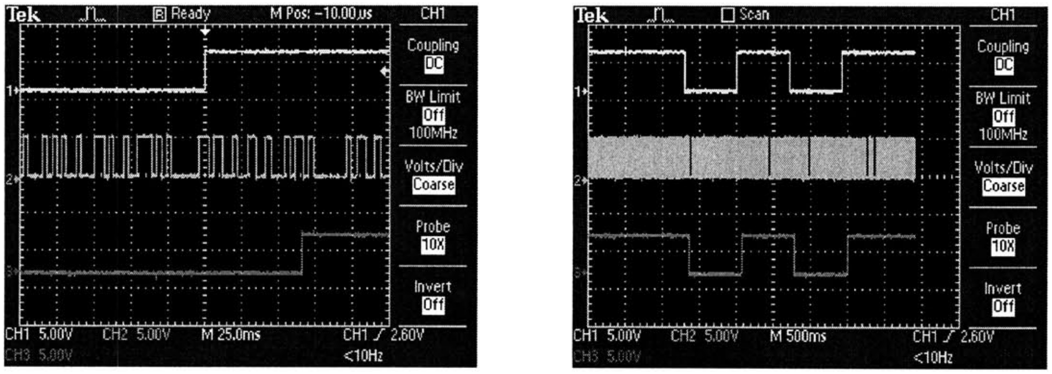
Each spread bit is despread using 11 continuous values of aggregate data. Therefore, each spread bit consists of 11 serial packets. The receiver expects these packets to have sequential IDs numbered from 1 to 11 in the Packet Count field in Figure 6-13. If the sequence is not received in order, than a packet is assumed lost and the received packets corresponding to that spread bit are discarded. This "all or nothing" algorithm keeps the serial transmission reliable by eliminating incomplete and invalid communications.

Using the serial_put function, the transmission module transmits a serial packet at 300 baud. The baud rate is set by the the SPBRG, and BRGH registers in the init_serial routine. The main routine calls serial_put after assembling a packet. A serial packet is assembled by taking the spread aggregate data and joining it with the counter variable, j, used to create the sequential packet ID. The transmission continues to transmit the spread data in an infinite loop, repeating after the 10th data bit is spread.

Like the transmission module, the reception module is configured for 300 baud serial communications using the init_serial routine in the SCI reception code of Appendix A.3. The main routine of this code also uses an infinite loop to continuously receive serial data from the transmission module. Using the serial_get function, the reception code blocks until a serial packet is received.

Once the packet is received, the code pulls apart the serial packet into its two components from Figure 6-13. The aggregate value is stored in the variable index to be used as the index for the antipodal translation as in Section 6.2.2 and identical to the clock-and-signal code of Appendix A.1. The synchronization ID is stored in the count variable which is compared to the expected count. The result of the comparison is stored in the sersync variable which is 1 when the serial transmission is in sync and 0 when it is not.

Upon successful serial synchronization the SCI reception code performs the same despreading process as the clock-and-signal code of Appendix A.1. It uses the Barker code to synchronize the CDMA despreading process and only output a valid value when the Barker codes of the reception and transmission modules are aligned. The output is once again on pin A1 with a synchronization signal on pin A3. Figure 6-14 shows oscilloscope captures of this system in operation.



(a) USART serial signal being transmitted (middle trace), the transmitter data signal, and despread data signal (top trace and bottom trace respectively). The despread signal is delayed slightly due to computation time.

(b) The despread signal (bottom trace) following the transmitter data signal (top trace). The middle trace is the same USART signal of 6-14(a).

Figure 6-14: The USART-based reception module despreading the aggregate data and following the transmitter's data. Subfigure 6-14(a) shows the transmitter data signal, despread data, and USART serial signal. Subfigure 6-14(b) shows the reception module properly outputting despread data.

As a result, using the USART serial communications interface, a system of sending CDMA data from a transmission PIC to a reception PIC is made feasible for one-wire transmission. In addition, communication is made more reliable by including a packet ID number that ensures the sequential data is received in order with no dropped packets; otherwise the data is discarded. Finally, using one wire for communications enables easy integration with the wireless transmission system of Chapter 3 and the wireless reception system of Chapter 4.

6.3.3 Serial Communication With A PC

With a more reliable communication system between the two PIC modules in place, longer pieces of data can be sent and received without the worry that they will be corrupted or lost. As a result a small, hard-coded set of data for spreading is no longer necessary. In order to allow for more dynamic data, the user of the transmission PIC module should be prompted for data to send. A data link to a PC would provide a terminal for text input that can then provide that input to the transmitting PIC. On the receiving side, another PC link could then display that received data on another terminal for ease of reading.

Communication with a PC takes the form of another serial communication channel for each PIC. Since the USART is already being used for PIC-to-PIC communications, the behavior of some I/O pins has to be modified to provide serial port capability (known as a "bit-bang" serial implementation). In addition, personal computers conform to the RS-232 serial communication specification. This specification is incompatible with the voltage and current output specifications of the PIC I/O pins. Therefore, a line driver circuit will also be necessary for a PC serial link. The overall system now takes the form of the block diagram in Figure 6-15.

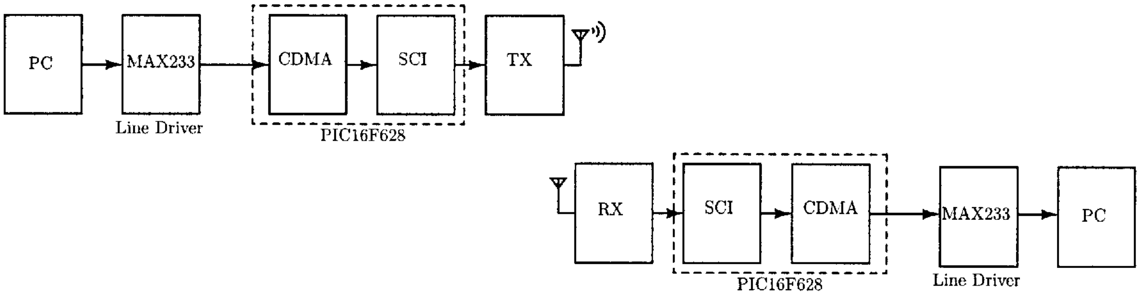


Figure 6-15: CDMA hardware block diagram including PC communication system.

The transmission PIC module code must be modified to include this bit-bang serial functionality. This modified code is listed in Appendix A.4. The program runs an initial `while` loop to collect a message from the user up to 8 characters (each of length 1 byte) long (the message can be shortened by entering the period character). After the message is acquired from the PC, the 8 corresponding bits from each character byte are individually spread and broadcast to the PIC reception module. The spreading process is exactly the same as Section 6.3.2 and the code in Appendix A.2.

The PIC reception module receives the aggregate data using the same routines as Appendix A.3. After despreading the data, it collects sets of 8 despread bits together to form each message byte. These message bytes are then sent to the PC terminal connected to the PIC reception module. The PC reception module communicates with its PC via its own set of bit-bang serial routines; listed along with the rest of the modified reception code in Appendix A.5.

Both the transmission and reception module need a line driver chip to communicate with the PC terminal using the RS-232 serial specification. Both PICs use the same driver circuit shown in Figure 6-16. The circuit can be configured with either the necessary reception module lines or transmission module lines as shown.

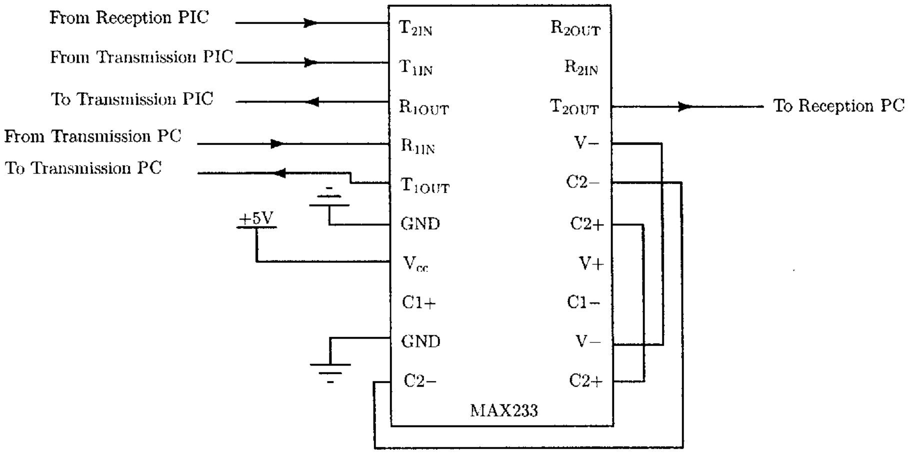


Figure 6-16: MAX233 line driver circuit for PC serial communication via the RS-232 standard. This circuit includes all possible connections for both the CDMA transmission and reception modules.

With the line driver circuit and the additional bit-bang serial code for both the transmission and reception module, two PCs can now communicate using the CDMA hardware. Although the CDMA transmission/reception method may not have changed, PC communication adds a vital human interaction component necessary for the success of any communication system.

6.3.4 Wireless Transmission

Wireless communication between the two PICs is now easier to facilitate with the one-wire serial system developed in Section 6.3.2. The transmission tools using the XR2206 from Chapter 3 require one signal to be broadcast with either AM or FM. That signal can now be the USART CDMA aggregate signal. Similarly, the reception framework involving the superheterodyne receiver from Chapter 4 can demodulate that AM or FM signal to recover the aggregate signal and pass it along to the reception PIC module for despreading. This application uses FM transmission in favor of AM due to the extra added resiliency to amplitude noise that FM provides for the broadcast signal.

The laboratory kit has a dedicated XR2206 input for the CDMA system. Once the input switch is set to "CDMA", and the modulation switch set to "FM" on the laboratory kit transmission board, the CDMA wireless transmission circuit is the one shown in Figure 6-17. The SCI CDMA signal will be connected to an LF356 operational amplifier for buffering and optional gain. The output of the opamp will then be sent to the XR2206 FM input. The FM broadcast system described in Chapter 3 will then transmit its signal.

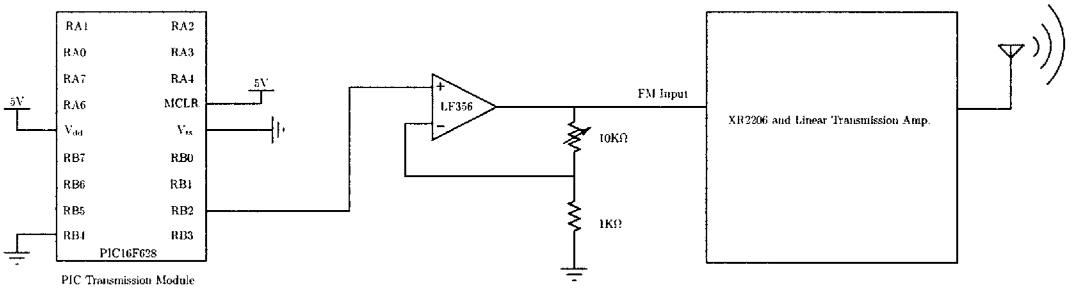


Figure 6-17: CDMA circuit for wireless FM transmission

On the reception side, the FM signal will be picked up by the resonant antenna circuit of Chapter 4. After being mixed down to the intermediate frequency of 455 kHz using the laboratory kit superheterodyne receiver system, the received FM signal will be sent to the LM565-based FM demodulator. The demodulator operates using the same circuit described in Chapter 5. However, after demodulation the CDMA aggregate signal does not conform to the voltage level specification of the reception PIC module's USART. As a result, a signal conditioning circuit will be necessary to bring the FM demodulator output up to the specification.

The maximum output voltage for the demodulation pin on the LM565 is 400 mV peak-to-peak [11]. For the demodulated SCI CDMA signal, this means that a logic high is no larger than 400 mV which is well out of the acceptable voltage range for the PIC reception module's USART input (which is a minimum of 2 V [16]). To bring the demodulated signal up to the USART specification, three different signal conditioning circuits are used.

The first operational amplifier circuit in Figure 6-18 subtracts any voltage offset present in the demodulated signal, allowing a logic low to be set to zero volts. The amount of offset to subtract will be provided by the LM565 itself on its reference pin (pin number 6). The circuit multiplies each input (the demodulated signal and the reference pin output) by a gain and subtracts the two. The gain applied to each input signal is related to the ratio of feedback resistor $\mathrm{R_f}$ and the input resistor to each input. In the case of the circuit shown, the demodulated signal has an adjustable gain of $\frac{\mathrm{R_f}}{\mathrm{R_1}}$ while the reference voltage has a gain of $\frac{\mathrm{R_f}}{\mathrm{R_f}} = 1$. The resistor $\mathrm{R_2}$ must be equal to $(2 - \frac{\mathrm{R_f}}{\mathrm{R_1}})$ in order for the circuit to operate properly.

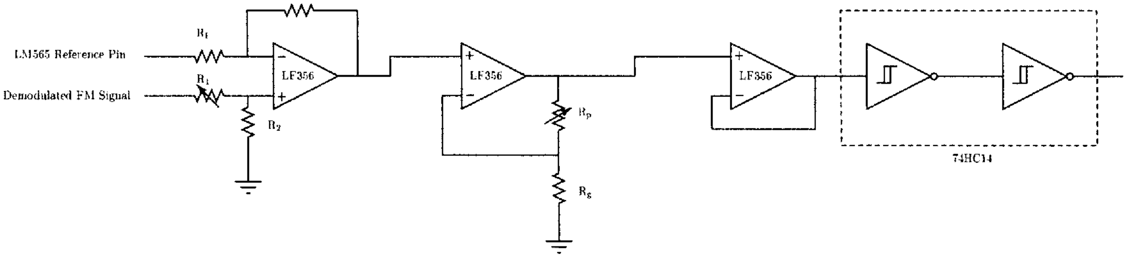


Figure 6-18: Circuit for conditioning the demodulated CDMA signal from the superheterodyne receiver to meet the PIC USART voltage specification.

Once the voltage offset has been removed and some gain applied, another gain stage is provided to add any remaining amplification necessary to the now offset-free signal. The gain of this next stage is $(1+\frac{\mathrm{R_p}}{\mathrm{R_g}})$, which is adjustable using the potentiometer $\mathrm{R_p}$. The last circuit contains three separate components: an opamp buffer and two Schmitt trigger, digital inverters. The buffer is a safety precaution to prevent the inverter following it from loading the opamp output. The inverters serve to "clean up" sharp edges present in the demodulated signal due to the built-in hysteresis of the 74HC14 components. The inverters also serve to adjust any properties of the signal that may not satisfy the static discipline. A second inverter is needed in order to keep the polarity of the demodulated signal correct. Two inverters were used over a single conventional TTL buffer because of their hysteretic properties. After the signal has been properly conditioned by these three circuits, it can be sent to the USART input pin of the PIC reception module just as in Section 6.3.2 or Section 6.3.3.

Consequently, adding a wireless component to the CDMA Hardware implementation requires no changes to PIC software and few additional circuits to the same PIC hardware. Moreover, the wireless system uses the same PIC transmission module with its output connected to the same FM transmission system discussed in Chapter 3 The system also uses the same PIC reception module, but does require an additional complement of circuits to help condition the demodulated FM signal to meet the specifications of its USART.

Chapter 7

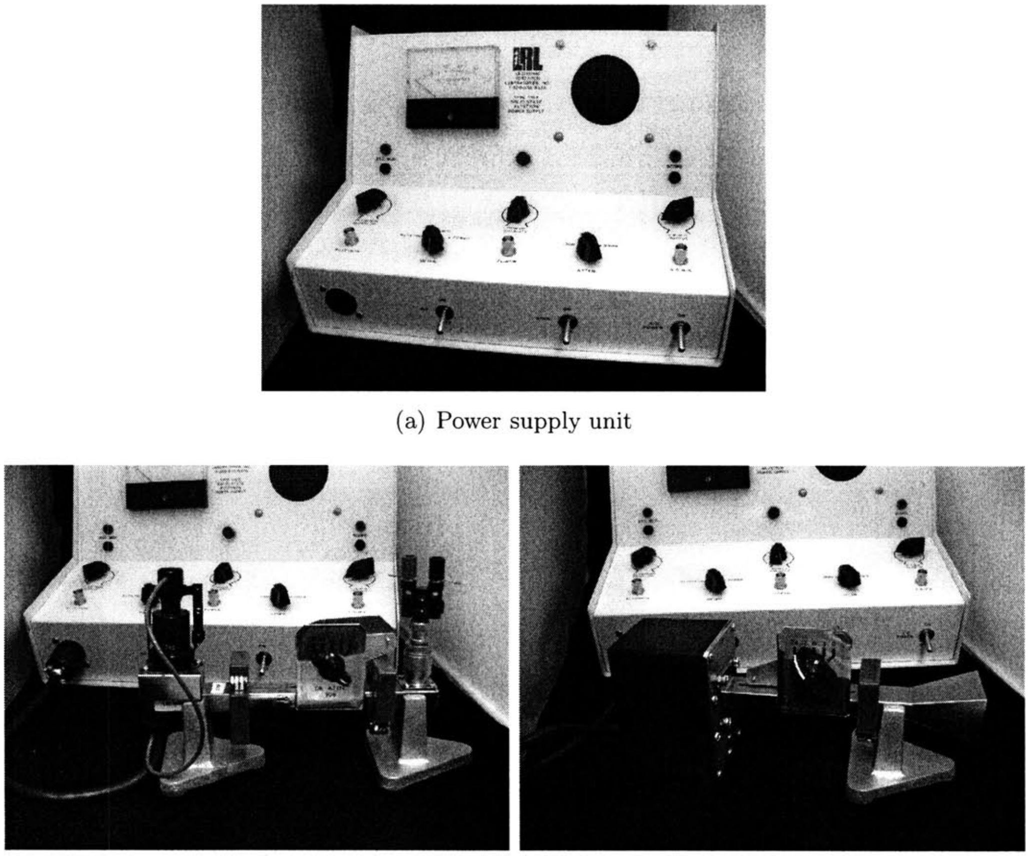
Lab 5: Antenna Electrodynamics

This fifth, last laboratory assignment will delve into topics related to the transmission and reception of signals on the order of 1 GHz in frequency. Several popular wireless technologies including cellular phones and devices based on IEEE standard 802.11 all transmit and receive signals in the 1 to 10 GHz range. Moreover, communication using signals in this frequency range is convenient due to the relatively small antenna sizes required (a property related to the wavelength of the electromagnetic signals used). Thus, because of the popularity of communication at these frequencies and the ease with which system components can be constructed, students will study Gigahertz-frequency transmission and reception. Specifically, they will use a pre-packaged microwave kit as a broadcast signal generator and compare the effectiveness and directionality of different antenna designs in reference to their use for high-frequency communication.

Students will begin by familiarizing themselves with the microwave trainer kit made by Lectronic Research Laboratory, the LRL550B-SS. The trainer will use a Reflex Klystron generator to create an electromagnetic signal at approximately 10 GHz. The Klystron will be attached to a waveguide in order to channel the signal into an antenna for broadcast. Students will tune the Klystron generator according to a setup procedure and then practice constructing an appropriate waveguide from given components. The final set of exercises will involve the construction of two different antenna types: a horn antenna and a dipole. The antenna patterns of each type will then be determined and measured in the laboratory using a pickup circuit adapted for use with the microwave trainer kit.

7.1 The LRL550B-SS Microwave Trainer

The LRL550B-SS microwave trainer kit is a pre-packaged platform for teaching students about electromagnetic waveguides. The kit is built by Lectronic Research Laboratories and consists of two main components: a power supply/calibration unit and a microwave generator (two distinct types of microwave generator are actually included with the kit), both components are shown in Figure 7-1. The power supply unit provides power for the microwave generator and contains several measurement instruments and power supply adjustments for optimizing the microwave generators output characteristics. The trainer kit comes packaged with two separate microwave generators, a Reflex Klystron tube generator and a solid-state generator. Students will predominantly use the Klystron tube generator which has a higher output amplitude.



(a) Power supply unit

(b) Reflex Klystron generator with waveguide

(c) Solid state generator with waveguide

Figure 7-1: LRL550B-SS microwave trainer with both of the included microwave generator units. Figure 7-1(b) shows the Reflex Klystron generator and Figure 7-1(c) shows the solid state generator.

7.1.1 Reflex Klystron Microwave Generator

To aid in students' familiarization with microwave generation techniques, a basic model of the Reflex Klystron microwave generator is shown in Figure 7-2. The Klystron generates a beam of electrons using a cathode at one end of the tube. As the electrons travel across the tube they will be accelerated by an accelerating grid. Once in the center of the tube, the electrons are attracted to a second, resonant cavity inside the tube by a separate modulating voltage (V_{mod}). The modulating voltage causes the electronics to bunch together or space apart (depending on the magnitude and polarity of the modulating voltage) as they exit the cavity. As the electrons approach the other end of the tube, a second electrode known as the repeller or reflector causes the electronics to reverse direction (the "reflex" action of the Reflex Klystron). The direction reversal is created by the power supply potential (V_{sup}) imposed on the reflector plate.

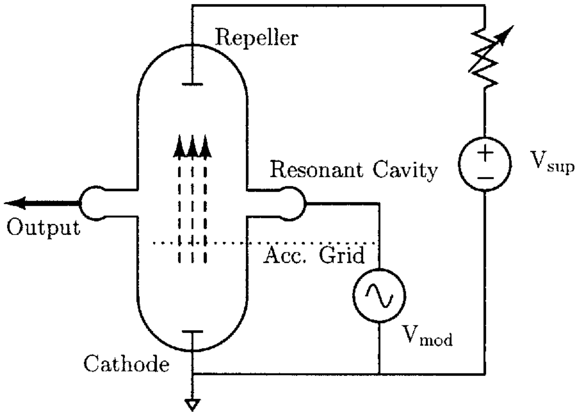


Figure 7-2: Simplified model of the Reflex Klystron microwave generator.

After reversing direction, the electrons pass through the cavity again and release electromagnetic energy at a specific frequency determined by their velocity and the resonant cavity dimensions. This energy released will be in the form of microwaves (the desired output) which are then emitted from the tube. The electrons are recollected in the resonant cavity after this energy release before the process repeats. Through this technique, the Reflex Klystron creates a fixed, Gigahertz-frequency output for use in microwave-based waveguide experiments.

Moreover, the frequency range of the Reflex Klystron included with the LRL microwave trainer kit can be determined using the dimensions of its included waveguide components. Each of the waveguide components included with the trainer kit has the rectangular dimensions shown in Figure 7-3. According to [17], the lowest-order mode that propagates inside a rectangular-metallic waveguide is the TE$_{10}$ mode, which has a spatial cutoff frequency ($k_{c_{10}}$) of

$$k_{c_{10}} = \frac{\pi}{b}$$

Substituting the dimension b from Figure 7-3,

$$k_{c_{10}} = \frac{\pi}{1.42 \times 10^{-2}}$$

The dispersion relation relates the spatial frequency $k_{c_{10}}$ to time by [17]:

$$k = \omega\sqrt{\mu\epsilon}$$

In the case of the air-filled waveguides of the kit, the permitivity ϵ and permeability μ are:

$$\mu = \mu_0 = 4\pi \times 10^{-7}\frac{\mathrm{H}}{\mathrm{m}} \qquad \epsilon = \epsilon_0 = 8.85 \times 10^{-12}\frac{\mathrm{F}}{\mathrm{m}}$$

Thus, the dispersion relation for the kit waveguides becomes,

$$k = \omega\sqrt{\mu_0\epsilon_0}$$

Substituting the dispersion relation into the equation for the fundamental mode,

$$k_{c_{10}} = \omega\sqrt{\mu_0\epsilon_0} = \frac{\pi}{1.42 \times 10^{-2}}$$

Solving for the frequency, the first step is to divide both sides by the square-root term,

$$\omega = \frac{\pi}{(1.42 \times 10^{-2})\sqrt{\mu_0\epsilon_0}}$$

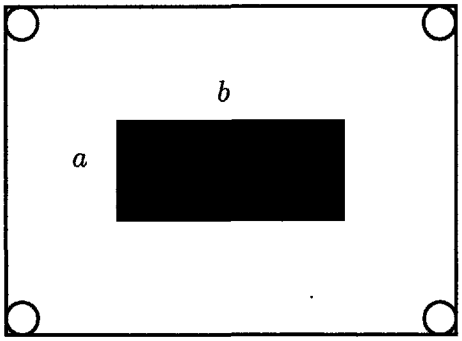
The equation above leaves the result in units of radians-per-second, to convert to units of Hertz, divide by 2π:

$$f = \frac{\omega}{2\pi} = \frac{\pi}{(1.42 \times 10^{-2})\, 2\pi\sqrt{\mu_0\epsilon_0}} \quad \text{Hz}$$

Solving the resulting equation for f leaves a maximum cutoff frequency of,

$$f \approx 10.56 \times 10^9 \quad \text{Hz}$$

Thus, using the calculation above and the included trainer kit manual, the fundamental operating frequency of the Reflex Klystrons operation is approximated to be nominally 10 GHz.



$a \approx 0.635$cm $\qquad$ $b \approx 1.42$cm

Figure 7-3: Rectangular waveguide dimensions for components included with the LRL microwave trainer kit.

7.2 Antenna Electrodynamics

After familiarizing themselves with the basic operation of the Reflex Klystron microwave generator and its adjustments using the power supply unit, students will study high frequency communication and antenna design using two sets of exercises. The first set will involve the use of the horn antenna attachment provided with the microwave trainer kit. Students will use the horn antenna to broadcast the microwave signal generated by the Reflex Klystron to the air. Then, they will use a specially-designed pickup circuit to measure the horn antennas field-strength pattern. From this measured pattern and knowledge of the antenna dimensions, students will gauge its effectiveness (i.e. gain and directionality) for ranged communication. Similarly, the second set of exercises will involve broadcasting the Reflex Klystron output using a dipole antenna. Students will construct an appropriate dipole antenna out of copper wire. They will then use the same pickup circuit from the first set of exercises to measure its field-strength pattern. Using the measured patterns from both antennas, students will be able to compare and contrast these two antennas in their ability to broadcast microwave signals.

7.2.1 Pickup Circuit

Prior to beginning either the horn antenna or the dipole antenna exercises, students will need to familiarize themselves with the strength-pattern measurement equipment they will be using. The measurement device, called a pickup circuit, is a printed circuit board that will attach to a waveguide component packaged with the microwave trainer kit, shown in Figure 7-4. Specifically, the waveguide component consists of a horn receiver antenna attached to a crystal diode. The crystal diode acts as a rectifier circuit which will be used to demodulate the approximately 10 GHz received signal. The modulation of the received signal is imparted by the modulation voltage V_{mod} in the Reflex Klystron generator. The rectified, crystal diode output will then be passed to the pickup circuit, the schematic of which is shown in Figure 7-5.

Moreover, the input stage of the pickup circuit completes the demodulator circuit with the crystal diode from the waveguide component. The time constant of the input resistor and capacitor should be set assuming a 10 kHz modulating frequency, thus typical values for R_C and C_C are 1.1 kΩ and 0.01 μF respectively (which will set the pass band just above 10 kHz). After the input stage, a buffer stage (using an LF356 operational amplifier) isolates the input from the intermediate, amplification stage. The amplification stage will allow students to amplify the demodulated signal up to a factor of 100 using another LF356 operational amplifier. This gain may be necessary depending on the distance the student will stand from the antenna to be measured as well as the output signal strength of the Reflex Klystron (which can vary from unit to unit).

Next, another buffer stage isolates the amplifier from the last stage of the pickup circuit, a resonant LC tank. The LC tank stage has components set to have a resonant frequency at approximately 10 kHz. This last stage provides both extra amplification (due to resonance)

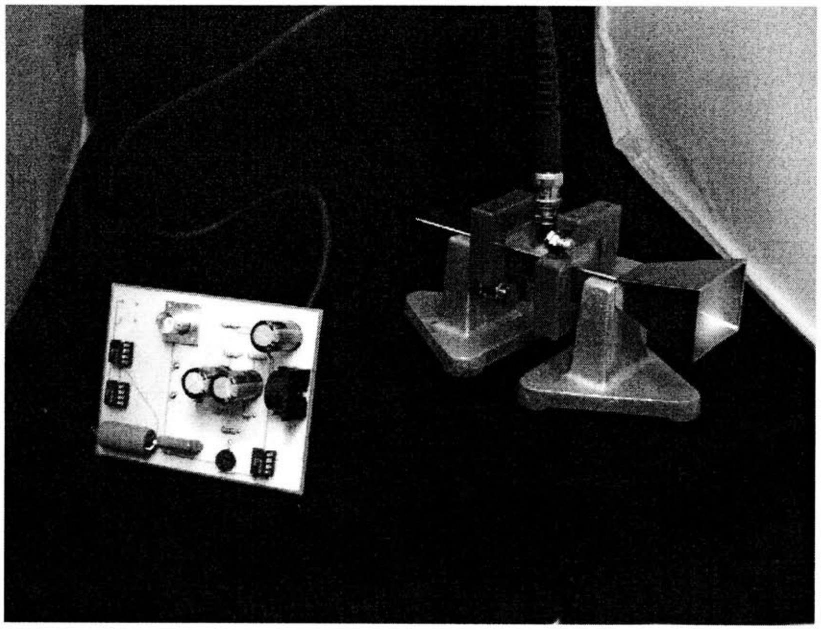


Figure 7-4: Pickup circuit printed circuit board alongside the pre-packaged horn receiver antenna and crystal diode (attached to BNC cable).

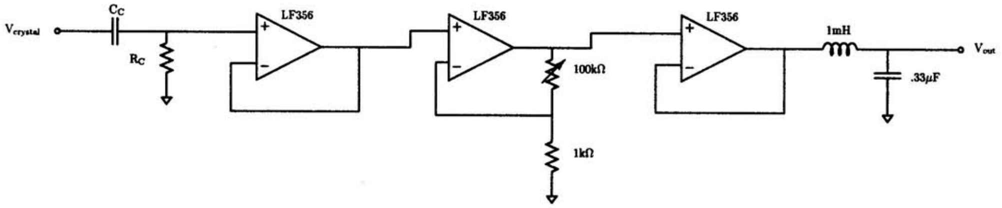


Figure 7-5: Schematic diagram of the pickup field-strength measurement circuit, laid out on the printed circuit board shown in Figure 7-4.

and a measure of selectivity for the expected 10 kHz demodulated signal. The selectivity in particular will aid in the attenuation of noise at other frequencies that may interfere with proper field-strength readings. Typical values for the inductor and capacitor of the tank are 1 mH and 0.33 μF respectively.

Thus, using this pickup circuit, students can make an effective measurement of the field-strength of their antenna designs. The included BNC connector output on the PCB will allow students to connect the pickup circuit to an oscilloscope. As they move about the laboratory with the pickup circuit in hand, students will make measurements of the field strength of the antenna by observing the magnitude of the sine wave at the circuits output. This magnitude will correlate directly to the strength of the antennas output field in a particular direction. By plotting the strength of the field at a particular direction, students can empirically recreate the field-strength pattern of any antenna. The PCB layout for the pickup circuit is included in Appendix C.3.

7.2.2 Horn Antenna

Using the pickup circuit and receiver antenna from Figure 7-4, students will first measure the field-strength pattern of the pre-packaged horn antenna attachment for the Reflex Klystron generator. The expected antenna pattern students should plot is shown in Figure 7-6. Approximating the horn antenna as a rectangular aperture, it should (in the far-field) create a Fraunhofer diffraction pattern [17]. The top-down view in the figure shows the expected pattern in the far-field, approximately a sinc function emanating from the horn antenna and extending outward. Students should be able to verify this pattern using their pickup circuit measurements.

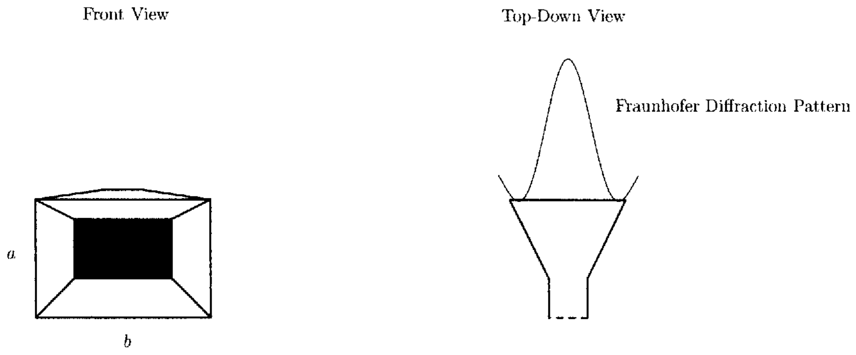


Figure 7-6: Horn antenna from two points of view. The frontal view shows the rectangular horn antennas vertical and horizontal dimensions (a and b respectively). The top-down view shows the expected field-strength pattern arising from Fraunhofer diffraction in the far-field.

Moreover, the Fraunhofer pattern exhibits significant directionality in front of the horn antenna with field-strength diminishing rapidly as the angle from the center increases. Note that no field should be measurable directly behind the antenna. Students will be expected to calculate the maximum magnitude of the horn antenna output, known as the directivity of the antenna D. Students will use the following derivation from [18] as a mathematical reference. The far-field electric field for a rectangular aperture is (from [18]):

$$\bar{E}_{ff}(r) = -\hat{\theta}\frac{jab}{\lambda r}E_0 e^{-jkr_0}\left(\frac{\sin(\frac{\pi\alpha_y a}{\lambda})}{\frac{\pi\alpha_y a}{\lambda}}\right)\left(\frac{\sin(\frac{\pi\alpha_x b}{\lambda})}{\frac{\pi\alpha_x b}{\lambda}}\right)$$

Where α_x and α_y are the angle away from the center of the antenna, λ is the wavelength of the output field, and a and b are the dimensions listed in Figure 7-6. The power radiated is given as:

$$P_R = \frac{ab|E_0|^2}{2\eta_0}$$

Therefore, the gain of the antenna is:

$$G(\alpha_x, \alpha_y) \approxeq \frac{\frac{|\bar{E}_{ff}|^2}{2\eta_0}}{\frac{P_R}{4\pi r^2}} = \frac{ab4\pi}{\lambda^2}\left(\frac{\sin^2(\frac{\pi\alpha_y a}{\lambda})}{(\frac{\pi\alpha_y a}{\lambda})^2}\right)\left(\frac{\sin^2(\frac{\pi\alpha_x b}{\lambda})}{(\frac{\pi\alpha_x b}{\lambda})^2}\right)$$

The maximum gain is at $G(0,0)$ which is:

$$G(0,0) = D = \frac{ab4\pi}{\lambda^2}$$

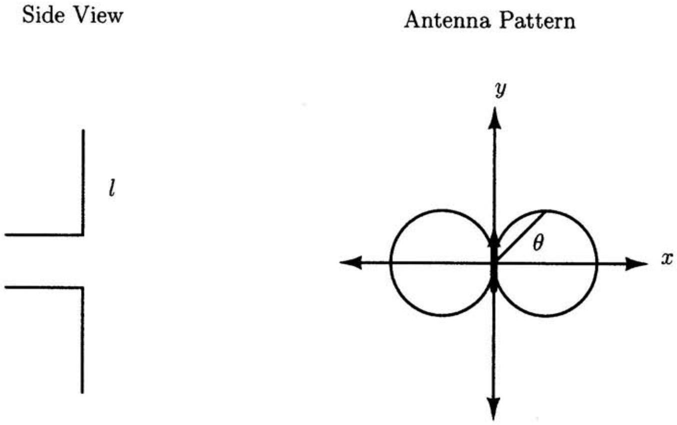
For the pre-packaged horn antenna with 10 GHz operating frequency the directionality will be:

$$D = \frac{(4.60\,\text{cm})(3.50\,\text{cm})4\pi}{(3\,\text{cm})^2} \approx 22.5$$

7.2.3 Dipole Antenna

With the horn antenna pattern measured and the its gain calculated, students will move on to performing the same set of exercises on a dipole antenna of their own construction. Assuming a 10 GHz operating frequency as with the horn antenna, students will need to construct a dipole antenna as pictured in Figure 7-7. The length l of each segment of the dipole should be one-half the wavelength of the output wave. Therefore, the length l of each segment for this set of dipole exercises should be approximately 1.5 cm. After constructing their dipole antenna, students should fix it to the waveguide output of the Reflex Klystron generator as shown in Figure 7-8.

Like the horn antenna exercise, students should use their pickup circuit and receiver antenna to measure the field-strength pattern of the dipole antenna. The expected antenna pattern should look like the two-lobed pattern shown in Figure 7-7. Unlike the horn antenna, the Hertzian dipole antenna is completely mathematically characterized. Its electric field,


Side View

Antenna Pattern

Figure 7-7: Dipole antenna configuration and field-strength pattern. The dipole should be created from two equal pieces of copper wire, each of length l. The antenna pattern consists of two lobes in the x-y plane with zero amplitude in the z plane.

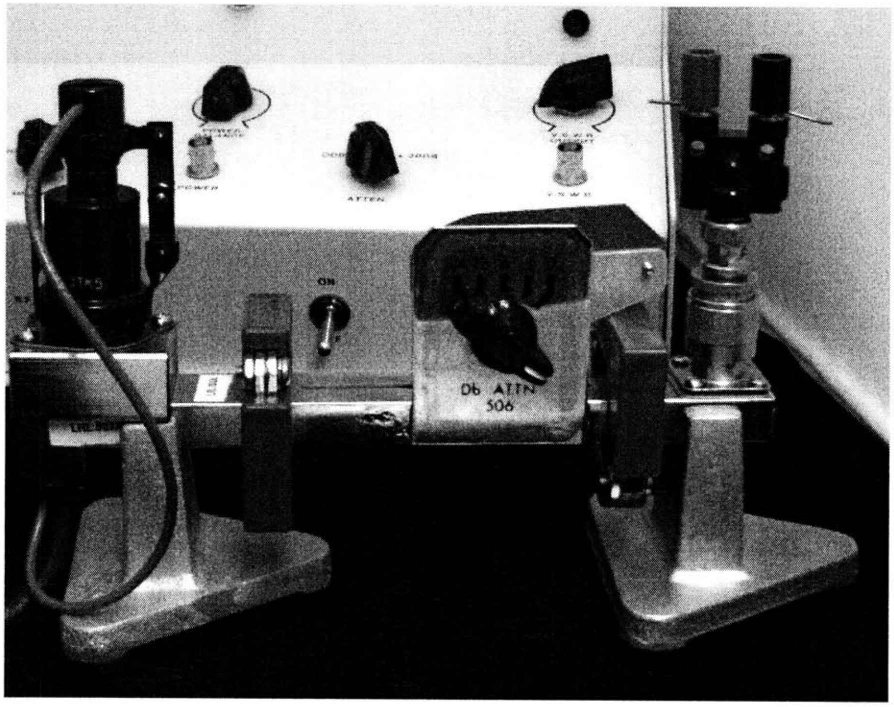


Figure 7-8: Dipole connection to the Reflex Klystron generator waveguide. The dipole wire segments should be attached using a BNC adaptor to the LRL502 termination.

as described in [17], is

$$\bar{E}_{ff}(r) = \hat{\theta} j\omega\mu I l \frac{e^{jkr}}{4\pi r} \sin(\theta)$$

Where I is the current through the dipole and l is the dipole length shown in Figure 7-7 The power radiated is:

$$P_R = \frac{4\pi}{3}\eta_0 \left(\frac{kIl}{4\pi}\right)^2$$

Therefore the gain is

$$G(\theta) = \frac{3}{2}\sin^2(\theta)$$

Therefore the directionality, D, of the dipole antenna is

$$D = G(\frac{\pi}{2}) = 1.5$$

Comparing the two antennas, students should recognize the horn antenna has a high directivity granting it significant gain in its forward direction compared to the dipole. However, the dipole sacrifices some directionality to make its signal available over a wider area. Thus, each antenna is suited for different applications. For example, the horn antenna could be well-suited for site to site communications while the dipole antenna could be better suited for wide-area broadcasts. Students will be encouraged to discuss these different applications and weigh the pros and cons of each type of antenna.

Chapter 8

Conclusions

This thesis proposes a curriculum with accompanying hardware and software to teach students about various wireless technologies. These technologies include RF power amplifiers, amplitude modulation, frequency modulation, superheterodyne receivers, phase-locked loops, CDMA, and antenna analysis. The intent in including so many technologies is to give students as diverse an experience as possible in one semester. This diversity will better prepare them for the design challenges they may have working on wireless systems in the field.

In order to teach such a broad spectrum of technologies, the laboratory kit and assignments are all designed around the notion of "permanent topologies". The main circuits students will use, lab after lab, are built directly into the laboratory kit. That is, the system topologies are fixed, but the component values are not. Key components won't be soldered into the kit, rather they will be replaced by sockets. Students will be required to learn how to design a system, select the right components, and place them in their kit to get the circuits to work. This approach shifts the emphasis of the laboratory course away from construction and more towards the finer points of design.

Furthermore, this thesis in no way sets out a complete curriculum to be taught. The exercises suggested are not exhaustive, several topics can be expanded upon and branched out from the assignments in Chapters 3 through 7, including wideband FM, synchronous AM demodulation, and IEEE 802.11-based networks. The assignments listed in this thesis are intended to be merely a starting point in the creation of a wireless laboratory course.

Appendix A

PIC C Code

A.1 Clock and Signal PIC Reception Code

```
#include <pic.h>
#include <pic16f6x.h>
#include <stdlib.h>

__CONFIG(0x3FF0);

// Use RB0/INT interrupt, which is an edge triggered interrupt.  Use
// falling edge (INTEDG=0).  INTF bit is set on interrupt.  Clear INTE
// to disable, re-enable by clearing INTF and then setting INTE.

signed char barker[11] = {1,-1,1,1,-1,1,1,1,-1,-1,-1};
signed char code[11] = {-1,1,-1,-1,-1,1,1,1,-1,1,1};
signed char analog[4] = {3, 1, -1, -3};
unsigned char j;
signed char sigsum,syncsum;


// rotates antipodal barker code by one with carry
void rotate (void) {
  unsigned char i;
  signed char temp[11];
  for(i=0; i<10;  i++) temp[i+1]=barker[i];
  temp[0]=barker[10];
  for(i=0;i<11; i++)barker[i]=temp[i];
```

```
}



void main (void) {
  unsigned int i;
  TRISA=0x00;
  TRISB=0x07;
  CMCON=0x07;
  INTE=0x01;
  INTEDG=0x00;
  GIE=0x01;
  j=0;
  while (1) {
    // sit and wait
  }
}


// convert antipodal summed signal to binary for output
unsigned char binary (signed char num) {
  if (num < 0) {
    return 1;
  }
  return 0;
}

void interrupt ISR (void) {
  unsigned char index;
  signed char syncvalue,sigvalue;
  if (INTF) {
    INTF=0x00;
    INTE=0x00;
    index= (RB2<<1) | RB1;
    syncvalue=analog[index] * barker[j];
    sigvalue=analog[index] * code[j];
    syncsum+=syncvalue;
    sigsum+=sigvalue;
    j++;
    if (j==11){
      if ((abs(syncsum))>=11){
        RA3=1;
        RA1=binary(sigsum);
      }
      else {
        RA3=0;
        RA1=0;
        rotate();
      }
      j=0;
      syncsum=0;
      sigsum=0;
    }
    INTE=0x01;
  }
}
```

A.2 SCI PIC Transmission Code

```
#include <pic.h>
#include <pic16f6x.h>

__CONFIG(0x3FF0);

unsigned char barker[11] = {0,1,0,0,1,0,0,0,1,1,1};
//Barker code specific data, for receiver syncing
unsigned char bdata[10] = {0,0,0,0,0,0,0,0,0,0};
unsigned char code[11] =  {1,0,1,1,1,0,0,0,1,0,0};
unsigned char orth[11] = {0,0,0,1,0,1,0,0,1,0,1};
unsigned char data[10] = {1,1,0,0,1,1,0,0,1,1}; //some data
unsigned char i,j,k;

void init_serial (void) {
  TRISB1=1;
  TRISB2=0;
  BRGH=0;
  SPBRG=207;
  SYNC=0;
  SPEN=1;
  TX9=0;
  TXIE=0;
  TXEN=1;
}

/* Send byte, blocking */
void serial_put(unsigned char x) {
  while(!TXIF);
  TXREG = x;
}

void main (void) {
  unsigned char value, packet;
  TRISA=0x00;
  TRISB=0x02;
  CMCON=0x07;
  init_serial();
  i=0;
  j=0;
  k=0;
  GIE=0x01;
  PEIE=0x01;
  while (1) {
    value=((data[i]^orth[j])+(data[i]^code[j])+(bdata[i]^barker[j]));
    packet=(unsigned char)(j<<4) | (value);
    serial_put(packet);
    j++;
    if (j==11) {
      j=0;
      RA0=data[i];
      i++;
      if (i==10){
        i=0;
      }
    }
  }
}
```

A.3 SCI PIC Reception Code

```
#include <pic.h>
#include <pic16f6x.h>
#include <stdlib.h>

__CONFIG(0x3FF0);

signed char barker[11] = {1,-1,1,1,-1,1,1,1,-1,-1,-1};
signed char code[11] = {-1,1,-1,-1,-1,1,1,1,-1,1,1};
signed char analog[4] = {3, 1, -1, -3};
unsigned char j;
signed char sigsum,syncsum;
bit sersync;

// Initialize the serial port
void init_serial (void) {
  TRISB1=1;
  TRISB2=0;
  BRGH=0;
  SPBRG=207;
  SYNC=0;
  SPEN=1;
  RX9=0;
  CREN=1;
  SREN=0;
  RCIE=0;
}

// rotates antipodal barker code by one with carry
void rotate (void) {
  unsigned char i;
  signed char temp[11];
  for(i=0; i<10;  i++) temp[i+1]=barker[i];
  temp[0]=barker[10];
  for(i=0;i<11; i++)barker[i]=temp[i];
}

// convert antipodal summed signal to binary for output
```

```
unsigned char binary (signed char num) {
  if (num < 0) {
    return 1;
  }
  return 0;
}

unsigned char serial_get(void) {
  for(;;) {
    if(RCIF) {
      return RCREG;
    }
    if(OERR) {
      CREN = 0;
      CREN = 1;
    }
  }
}

void main (void) {
  unsigned char index, packet, count;
  signed char syncvalue,sigvalue;
  TRISA=0x00;
  CMCON=0x07;
  init_serial();
  j=0;
  sersync=0;
  while (1) {
    packet=serial_get();
    index=(packet & 0x03);
    count=(packet >> 4);
    sersync=(j==count);
    if (sersync){
      syncvalue=analog[index]*barker[j];
      sigvalue=analog[index]*code[j];
      syncsum+=syncvalue;
      sigsum+=sigvalue;
      RA0^=1;
      if (j==10){
        if ((abs(syncsum))>=10){
          RA3=1;
          RA1=binary(sigsum);
        }
        else {
          RA3=0;
          RA1=0;
          rotate();
        }
        j=0;
        syncsum=0;
        sigsum=0;
      }
      else {
        j++;
      }
    }
    else {
      RA3=0;
      RA1=0;
      j=0;
    }
  }
}
```

A.4 SCI PIC Transmission Code With PC Communication

```
#include <pic.h>
#include <pic16f6x.h>

__CONFIG(0x3FF0);

// Define timings for bit-bang serial reception and transmission
#define BIT_RX 84
#define BIT_TX 104

// Macro for delaying a set number of cycles
#define delay_cycles(x) do { \
        unsigned char _i; \
        _i = ((x) - 1) / 3; \
        while(--_i != 0) continue; \
        if((((x) - 1) % 3) >= 1) asm("nop"); \
        if((((x) - 1) % 3) >= 2) asm("nop"); \
} while(0)

// Bit-bang serial initialization
void bb_init(void){
  TRISB5=0;
  TRISB6=1;
  RB5=0x01;
}

// Get a byte via bit-bang serial
unsigned char bb_get(void){
  unsigned char i,temp;
  temp=0;
  // Wait for a start bit
  while(RB6==1) continue;
  delay_cycles(BIT_RX+50);
  for(i=0;i<8;i++){
    temp|=(RB6<<i);
    delay_cycles(BIT_RX);
  }
  delay_cycles(BIT_RX);
  if(RB6==1){
    return temp;
  }
  else{
    return 0;
  }
}

// Put out a byte via bit-bang
void bb_put(unsigned char x){
  /* Start Bit */
  RB5=0;
  delay_cycles(BIT_TX-1);

  /* Code borrowed from Jim Paris: */
/* This compiles as 4 cycles regardless of whether setting or clearing RB5. */
#define do_set(cond) do { if((cond))RB5=1; if(!(cond)) RB5=0; } while(0)

/* Data */
  do_set(x & 0x01); delay_cycles(BIT_TX - 4);
  do_set(x & 0x02); delay_cycles(BIT_TX - 4);
  do_set(x & 0x04); delay_cycles(BIT_TX - 4);
  do_set(x & 0x08); delay_cycles(BIT_TX - 4);
  do_set(x & 0x10); delay_cycles(BIT_TX - 4);
  do_set(x & 0x20); delay_cycles(BIT_TX - 4);
  do_set(x & 0x40); delay_cycles(BIT_TX - 4);
  do_set(x & 0x80); delay_cycles(BIT_TX - 4);

  /*Stop*/
  RB5=1;
  delay_cycles(BIT_TX-4);
}


// Put out a string via bit-bang
void bb_put_string(char *s){
  while(s && *s)
    bb_put(*s++);
}
```

```
// Global variables
unsigned char barker[11] = {0,1,0,0,1,0,0,0,1,1,1}; // Digital Barker code
unsigned char bdata[10] = {0,0,0,0,0,0,0,0,0,0}; //Barker-code-specific data
unsigned char code[11] =  {1,0,1,1,1,0,0,0,1,0,0}; //Orthogonal code
 // Code orthogonal to the other two
unsigned char orth[11] = {0,0,0,1,0,1,0,0,1,0,1};
// array to store an eight character message to transmit
unsigned char message[8];
//Startup message:
bank1 unsigned char start_msg[]="\n␣Enter␣up␣to␣8␣characters,␣end␣with␣a␣'.'\n>";
unsigned char data[8]; //place holder for each bit of each message byte
unsigned char i,j;

// Break up an 8-bit ASCII character into 8 separate bits, store in data[]
void ascii_break (unsigned char a){
  unsigned char i;
  for(i=0;i<8;i++){
    data[i]=(a>>i)&0x01;
  }
}


// USART serial initialization
void init_serial (void) {
  TRISB1=1;
  TRISB2=0;
  BRGH=0;
  SPBRG=207;
  SYNC=0;
  SPEN=1;
  TX9=0;
  TXIE=0;
  TXEN=1;
}

// Send byte, blocking
void serial_put(unsigned char x) {
  while(!TXIF);
  TXREG = x;
}



void main (void) {
  unsigned char value,packet,n,k,temp;
  TRISA=0x00; // Port A all outputs
  CMCON=0x07; //Turn off comparators
  bb_init(); //initialize bit-bang
  init_serial(); //initialize USART
  while(1){
    // Put out the startup message
    bb_put_string(start_msg);
    k=0;
    n=0;
    // Get 8 characters, or stop when you received a period char.
    while(k!=8){
      temp=bb_get();
      bb_put(temp);
      if(temp=='.') break;
      message[k]=temp;
      k++;
    }
    while (n<8) {
      // Break up ASCII character into 8 individual bits
      ascii_break(message[n]);
      // Spread each bit:
      value=((data[i]^orth[j])+(data[i]^code[j])+(bdata[i]^barker[j]));
      packet=(unsigned char)(j<<4) | (value); //Assemble a serial packet
      serial_put(packet); //Put out the packet using USART
      j++;
      if (j==11) {
        j=0;
        RA0=data[i]; //Output unspread data to pin A0 for debugging
        i++;
        if (i==8){
          i=0;
```

```
                n++;
              }
            }
          }
        }
      }
```

A.5 SCI PIC Reception Code With PC Communication

```
#include <pic.h>
#include <pic16f6x.h>
#include <stdlib.h>

__CONFIG(0x3FF0);

// Define timings for bit-bang serial transmission
#define BIT_TIME 104

// Macro for delaying a set number of cycles
#define delay_cycles(x) do { \
        unsigned char _i; \
        _i = ((x) - 1) / 3; \
        while(--_i != 0) continue; \
        if((((x) - 1) % 3) >= 1) asm(”nop”); \
        if((((x) - 1) % 3) >= 2) asm(”nop”); \
} while(0)

// Bit-bang serial initialization
void bb_init(void){
  TRISB5=0;
  RB5=0x01;
}

// Put out a byte via bit-bang
void bb_put(unsigned char x){
  /* Start Bit */
  RB5=0;
  delay_cycles(BIT_TIME-1);

  /* Code borrowed from Jim Paris: */
  /* This compiles as 4 cycles regardless setting or clearing RB5. */
#define do_set(cond) do { if((cond))RB5=1; if(!(cond)) RB5=0; } while(0)
  /* Data */
  do_set(x & 0x01); delay_cycles(BIT_TIME - 4);
  do_set(x & 0x02); delay_cycles(BIT_TIME - 4);
  do_set(x & 0x04); delay_cycles(BIT_TIME - 4);
  do_set(x & 0x08); delay_cycles(BIT_TIME - 4);
  do_set(x & 0x10); delay_cycles(BIT_TIME - 4);
  do_set(x & 0x20); delay_cycles(BIT_TIME - 4);
  do_set(x & 0x40); delay_cycles(BIT_TIME - 4);
  do_set(x & 0x80); delay_cycles(BIT_TIME - 4);

  /*Stop*/
  RB5=1;
  delay_cycles(BIT_TIME-4);
}


// Initialize the serial port
void init_serial (void) {
  TRISB1=1;
  TRISB2=0;
  BRGH=0;
  SPBRG=207;
  SYNC=0;
  SPEN=1;
  RX9=0;
  CREN=1;
  SREN=0;
  RCIE=0;
}

// rotates antipodal Barker code by one with carry
void rotate (void) {
  unsigned char i;
  signed char temp[11];
  for(i=0; i<10;  i++) temp[i+1]=barker[i];
  temp[0]=barker[10];
  for(i=0;i<11; i++)barker[i]=temp[i];
}

// convert antipodal summed signal to binary for output
unsigned char binary (signed char num) {
  if (num < 0) {
    return 1;
  }
  return 0;
}

// Get a serial byte using USART
unsigned char serial_get(void) {
  for(;;) {
    if(RCIF) {
      return RCREG;
    }
    if(OERR) {
      CREN = 0;
      CREN = 1;
    }
  }
}


// Global variables
signed char barker[11] = {1,-1,1,1,-1,1,1,1,-1,-1,-1};
signed char code[11] = {-1,1,-1,-1,-1,1,1,1,-1,1,1};
signed char analog[4] = {3, 1, -1, -3};
unsigned char j,n,letter;
signed char sigsum,syncsum;
bit sersync;



void main (void) {
  unsigned char index, packet, count;
  signed char syncvalue,sigvalue;
  TRISA=0x00; // Port A is all outputs
  CMCON=0x07; // Turn off comparators
  bb_init(); //initialize bit-bang
  init_serial(); //initialize USART
  j=0;
  n=0;
  sersync=0;
  letter=0x00;
```

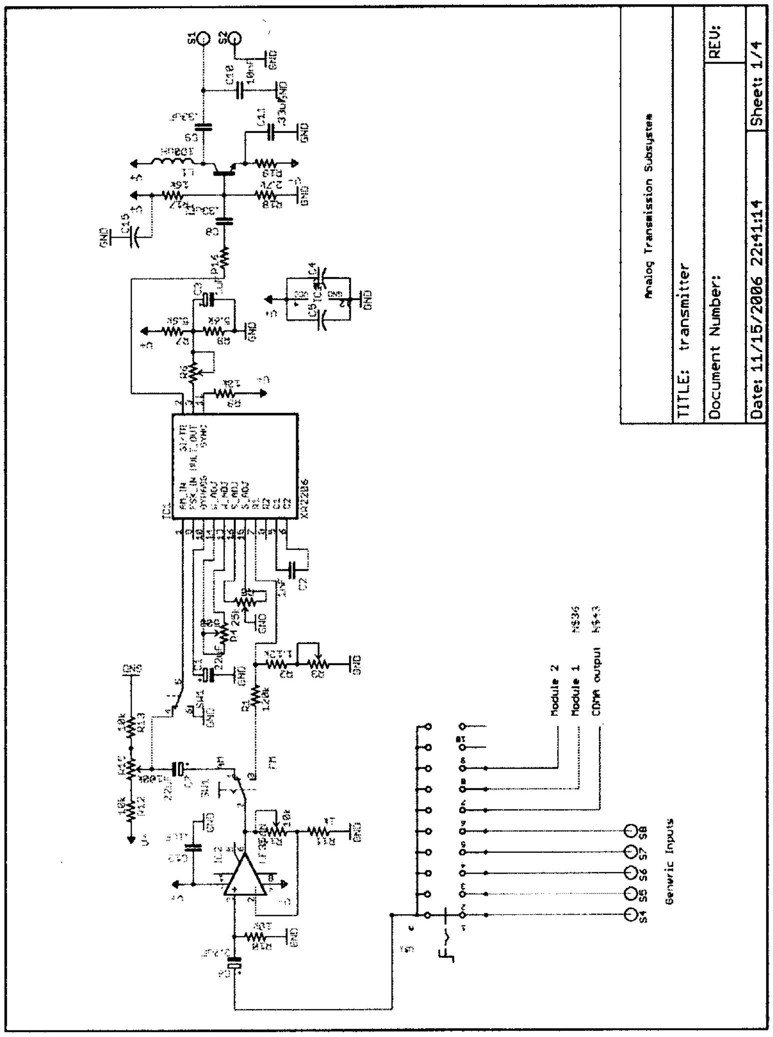
```
while (1) {
  packet=serial_get(); //get a serial packet
  index=(packet & 0x03); //break up the packet into CDMA data
  count=(packet >> 4); // and serial sync. count
  sersync=(j==count); //see if serial is synced up
  // if we are start despreading
  if (sersync){
    syncvalue=analog[index]*barker[j]; //despread Barker data
    sigvalue=analog[index]*code[j]; //despread user data
    syncsum+=syncvalue; //sum up despread Barker data
    sigsum+=sigvalue; // sum up despread user data
    if (j==10){
      // if we're all synched...
      if ((abs(syncsum))>=10){
        RA3=1;
        letter|=(unsigned char)((binary(sigsum))<<n);
        n++;
        if(n==8){
          n=0;
          bb_put(letter); //got a byte?, put it out via bit-bang
          letter=0;
        }
      }
      else {
        RA3=0;
        letter=0;
        n=0;
        rotate(); // not synched?  rotate Barker and try again
      }
      j=0;
      syncsum=0;
      sigsum=0;
    }
    else {
      j++;
    }
  }
  else {
    RA3=0;
```

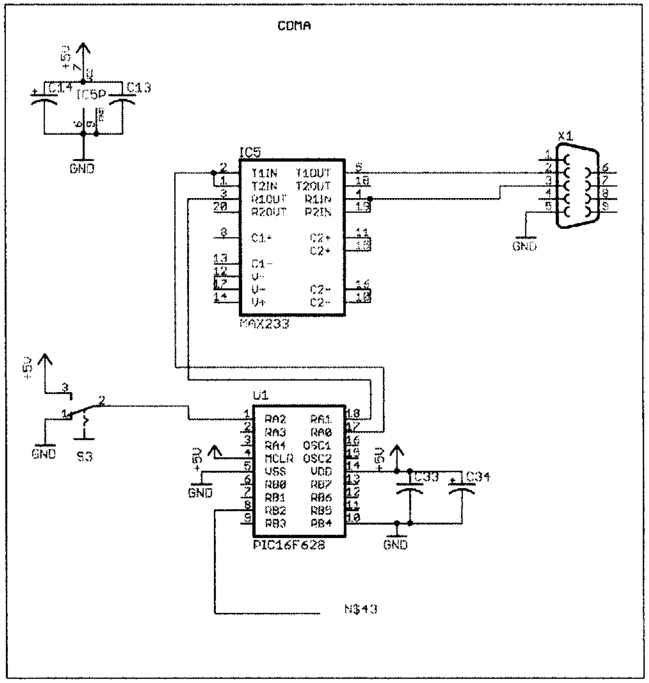
```
                letter=0;
                j=0;
                n=0;
            }
        }
}
```

Appendix B

Eagle CAD Schematics

B.1 Lab Kit Transmission Module Schematics



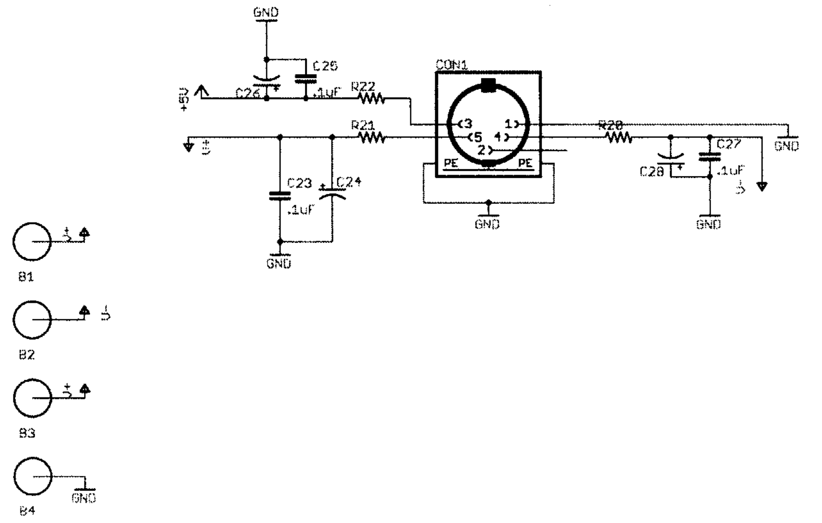


Built-In Communications Subsystems

TITLE: transmitter

Document Number: REV:

Date: 11/15/2006 22:41:14 Sheet: 2/4

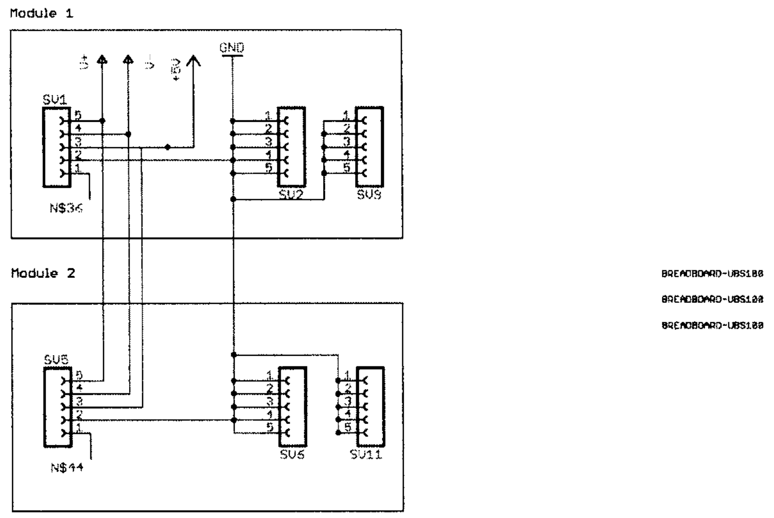


Power Subsystem

TITLE: transmitter

Document Number: REV:

Date: 11/15/2006 22:41:14 Sheet: 3/4



Extension Modules

TITLE: transmitter

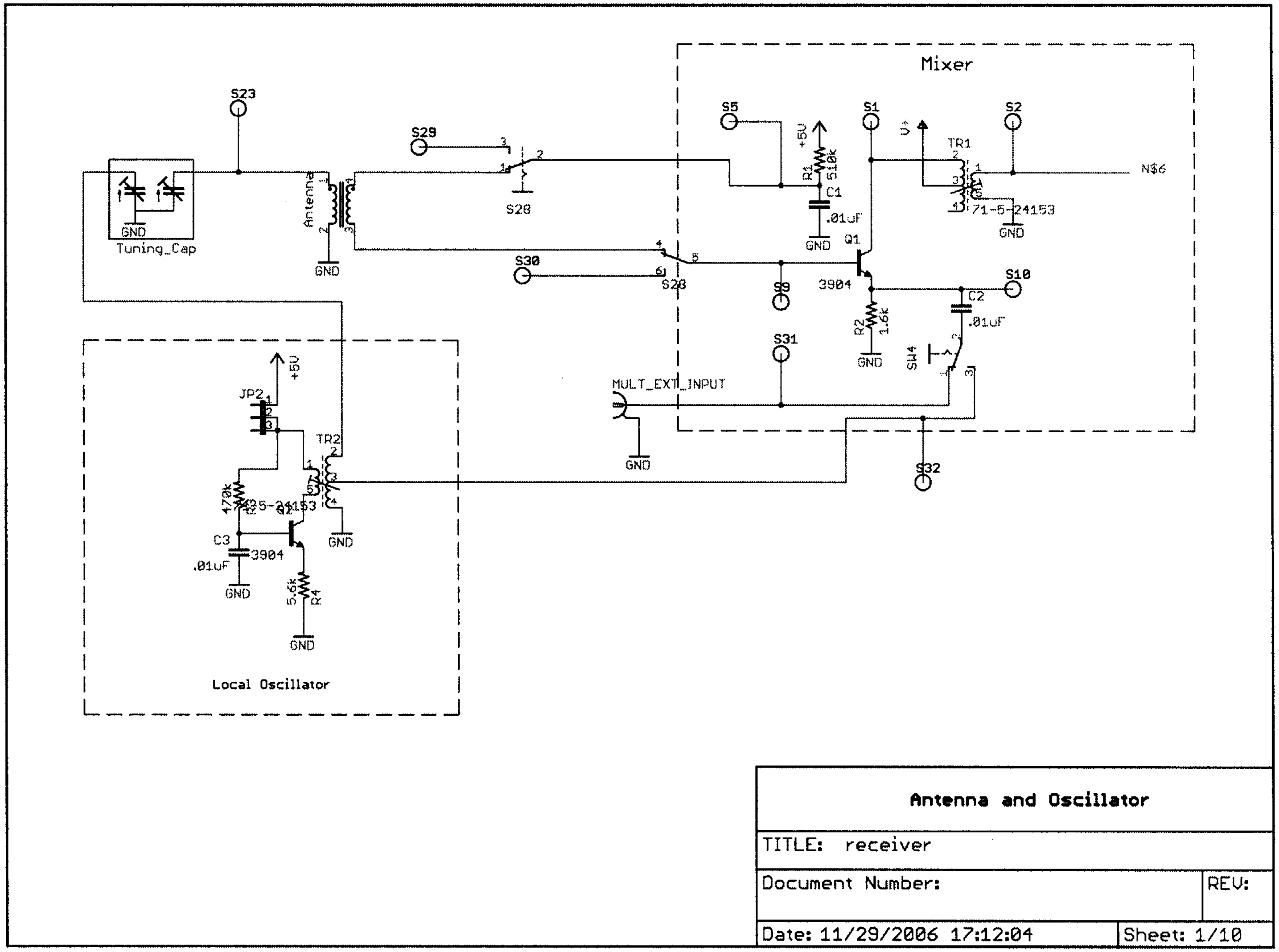
Document Number:

REV:

Date: 11/15/2006 22:41:14

Sheet: 4/4

B.2 Lab Kit Receiver Module Schematics



Antenna and Oscillator

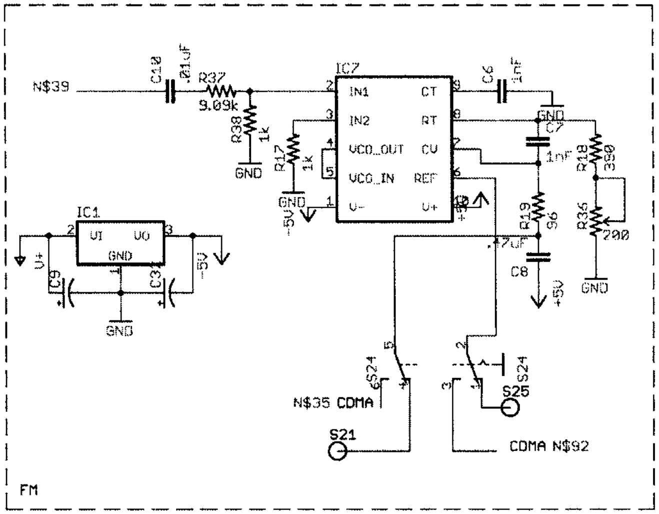
TITLE: receiver

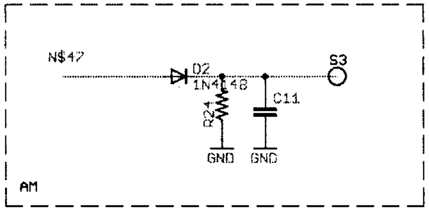
Document Number:

REV:

Date: 11/29/2006 17:12:04

Sheet: 1/10



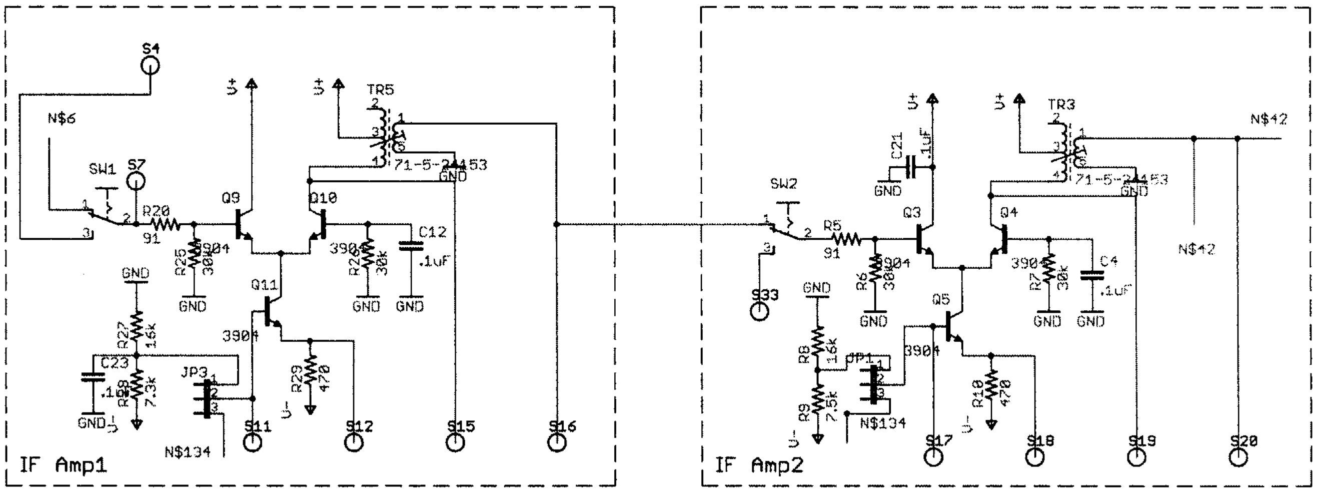


Built-in Demodulators

TITLE: receiver

Document Number: REV:

Date: 11/29/2006 17:12:04 Sheet: 2/10

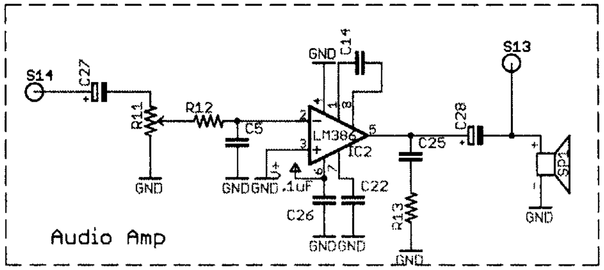


Superhet. IF Amps

TITLE: receiver

Document Number: REV:

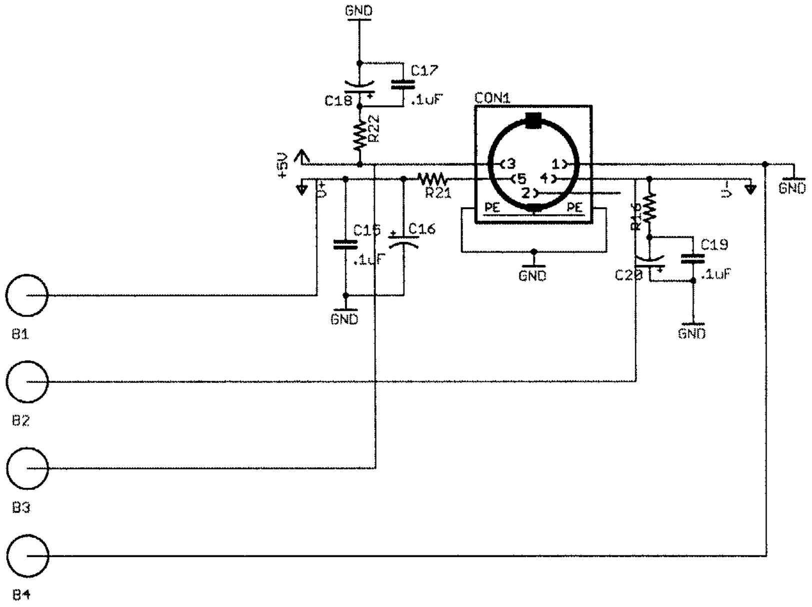
Date: 11/29/2006 17:12:04 Sheet: 3/10



BREADBOARD-UBS100

BREADBOARD-UBS100

Miscellaneous	
TITLE: receiver	
Document Number:	REV:
Date: 11/29/2006 17:12:04	Sheet: 4/10

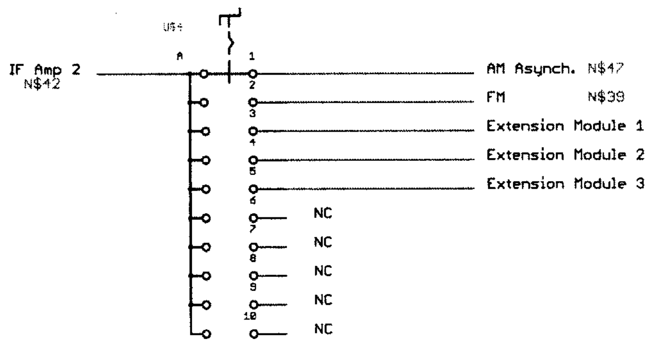


Power Supply

TITLE: receiver

Document Number: REV:

Date: 11/29/2006 17:12:04 Sheet: 5/10



Demodulator Switch

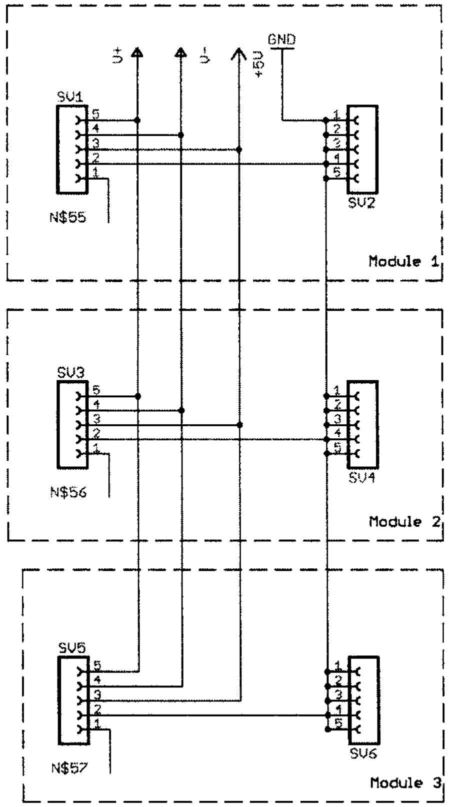
TITLE: receiver

Document Number:

REV:

Date: 11/29/2006 17:12:04

Sheet: 6/10

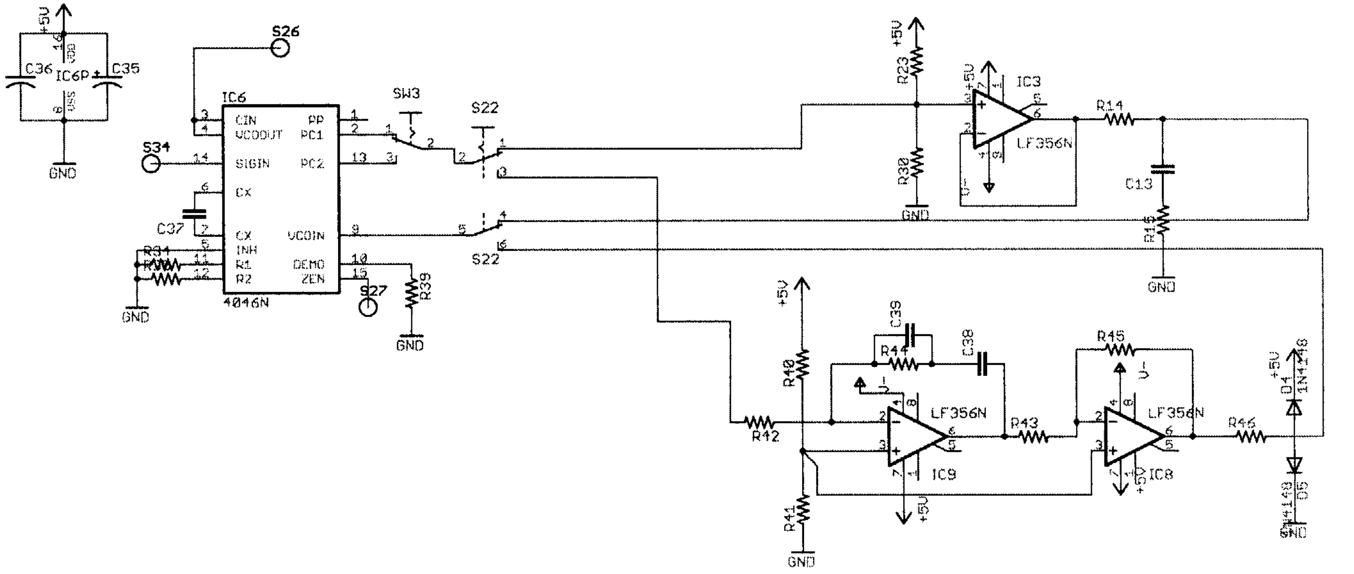


Extension Modules

TITLE: receiver

Document Number: REV:

Date: 11/29/2006 17:12:04 Sheet: 7/10

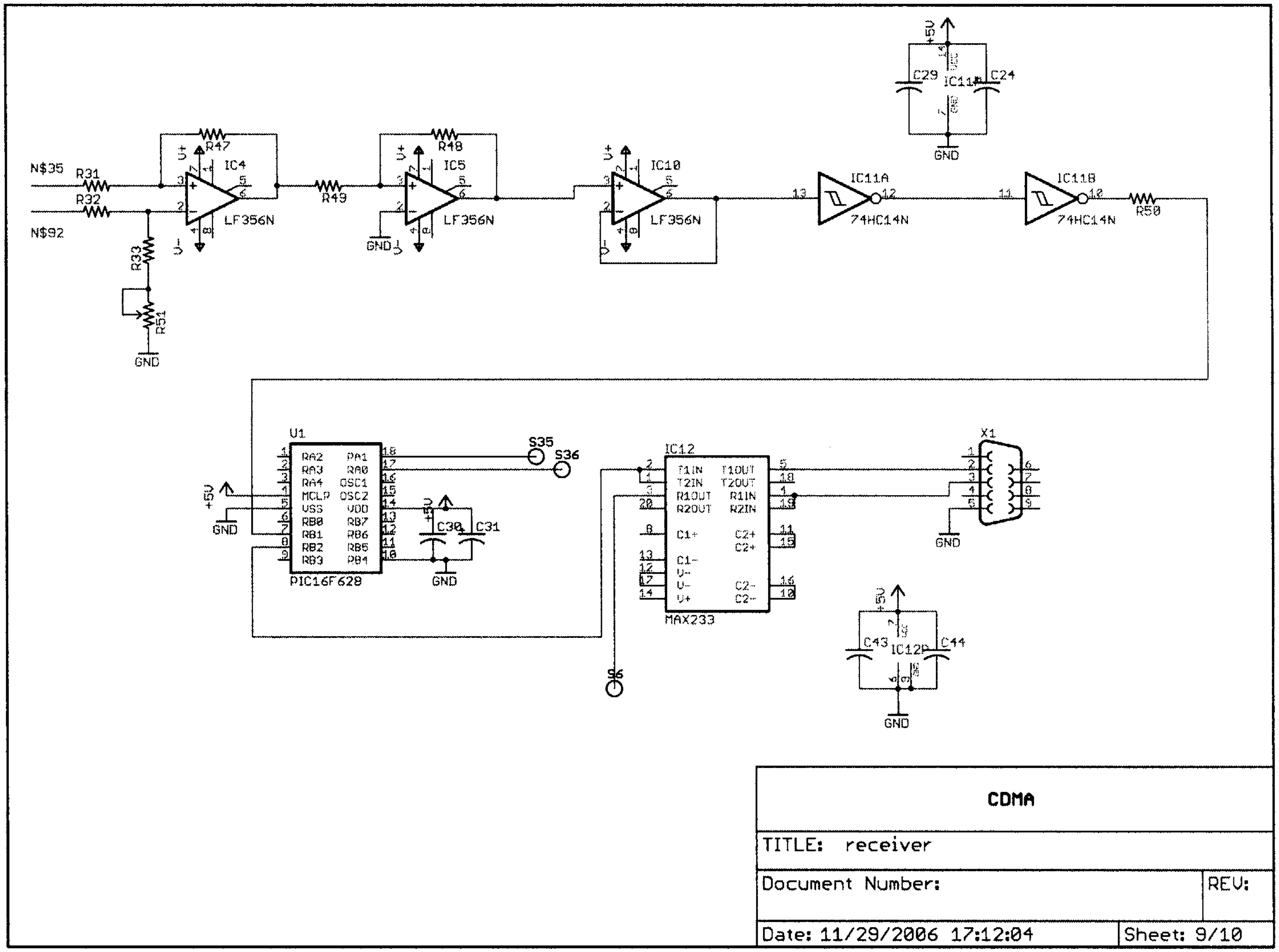


CD4046

TITLE: receiver

Document Number: REV:

Date: 11/29/2006 17:12:04 Sheet: 8/10

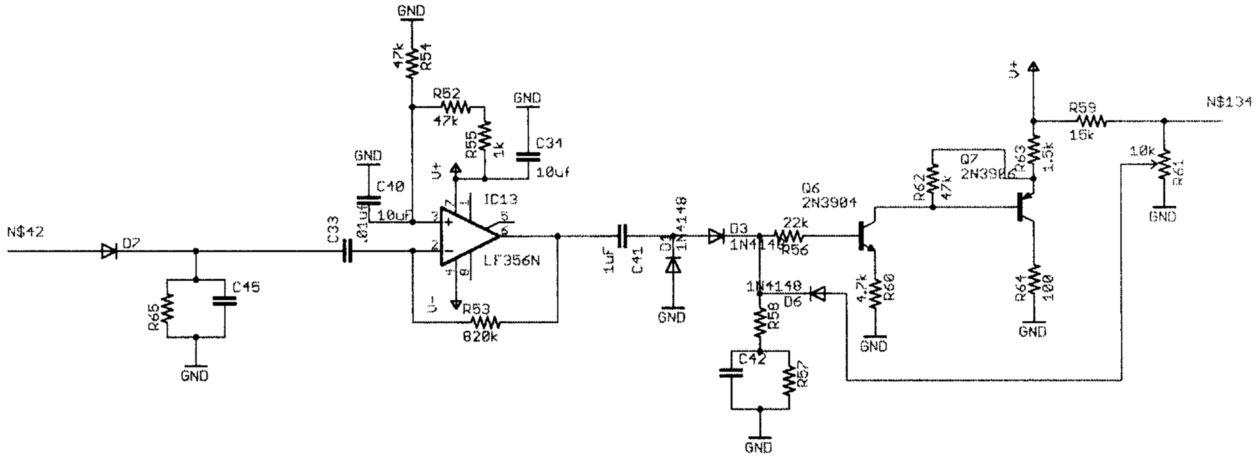


CDMA

TITLE: receiver

Document Number: REV:

Date: 11/29/2006 17:12:04 Sheet: 9/10



Automatic Gain Control

TITLE: receiver

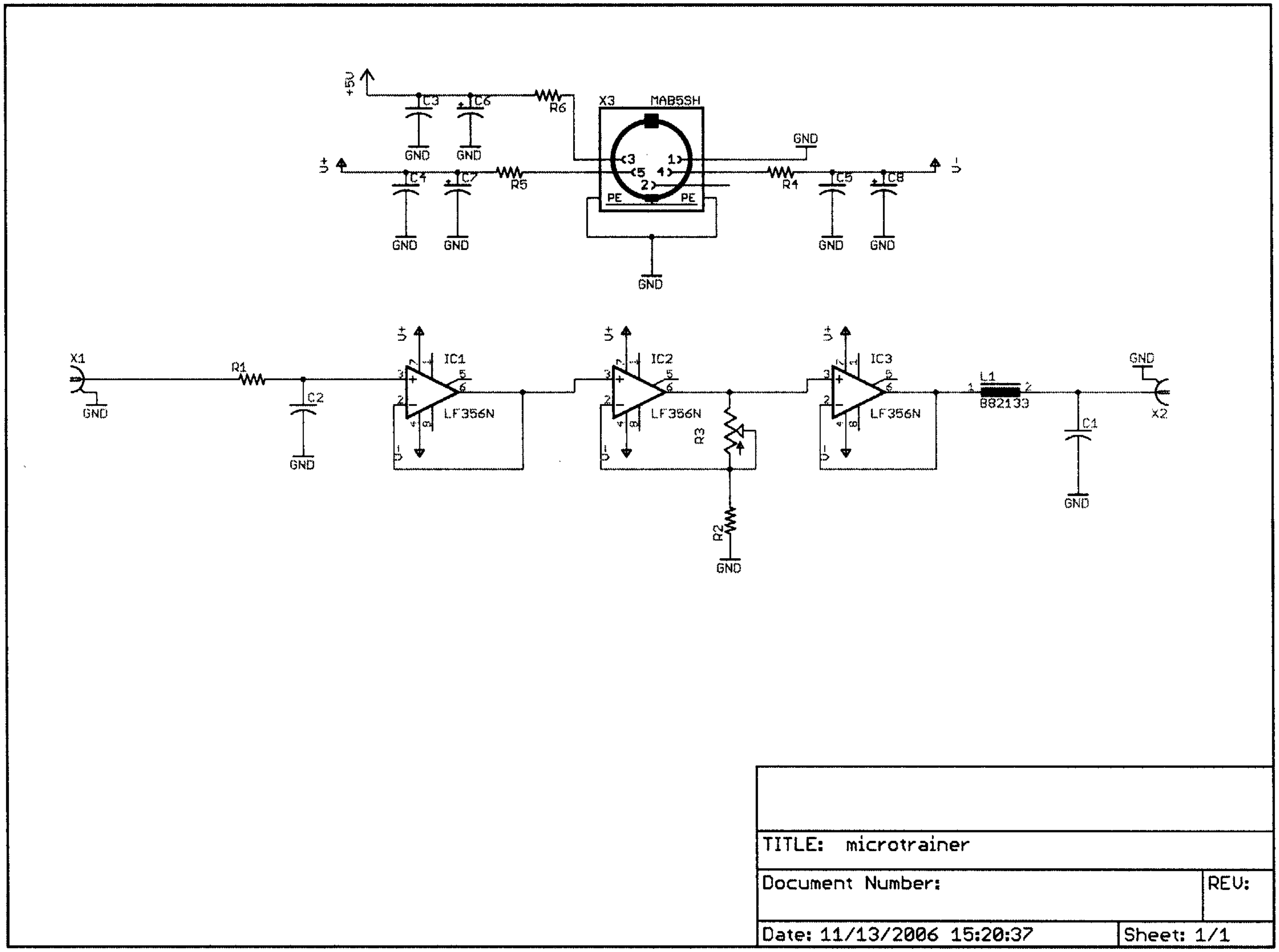
Document Number:

REV:

Date: 11/29/2006 17:12:04

Sheet: 10/10

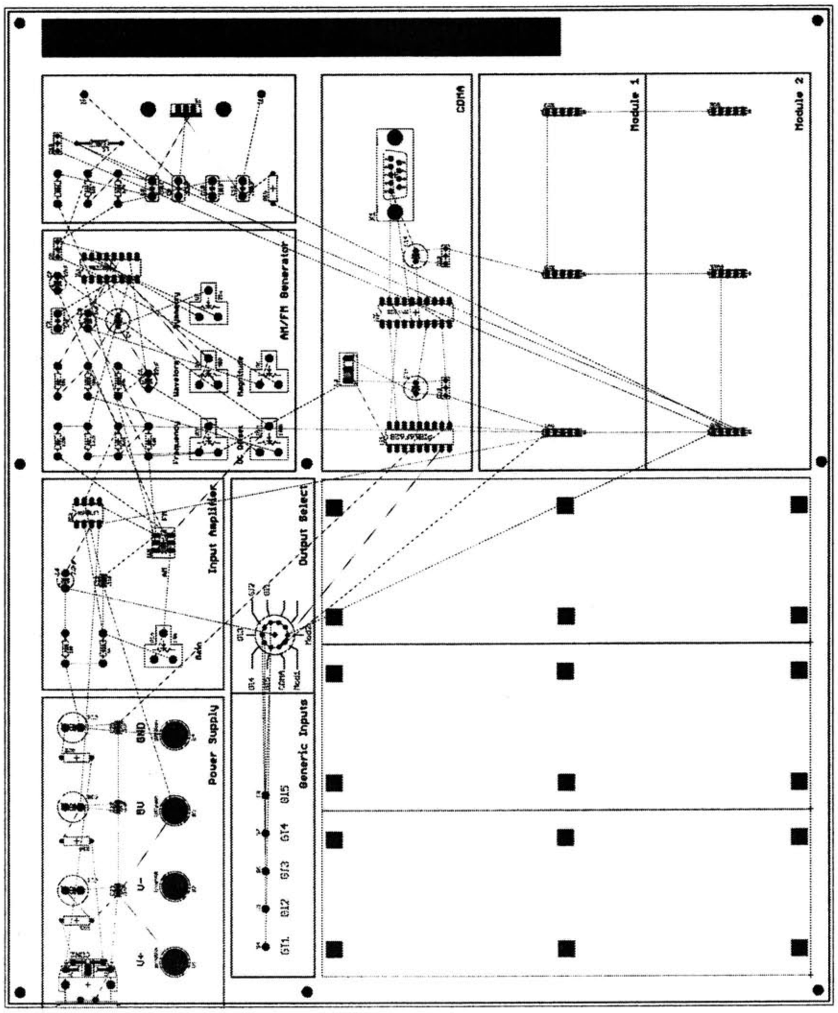
B.3 Pickup Circuit Schematic



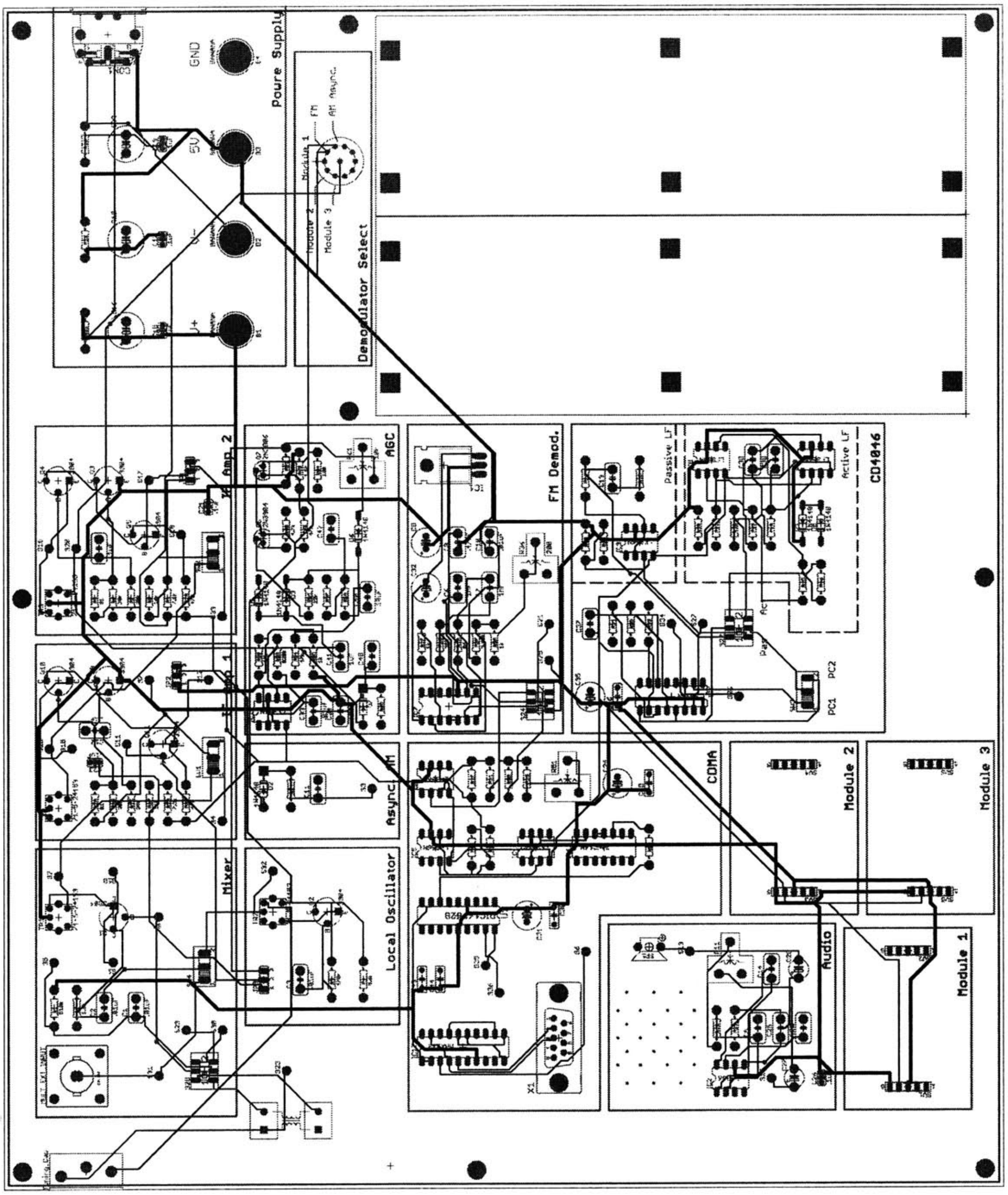
Appendix C

Eagle CAD Board Layouts

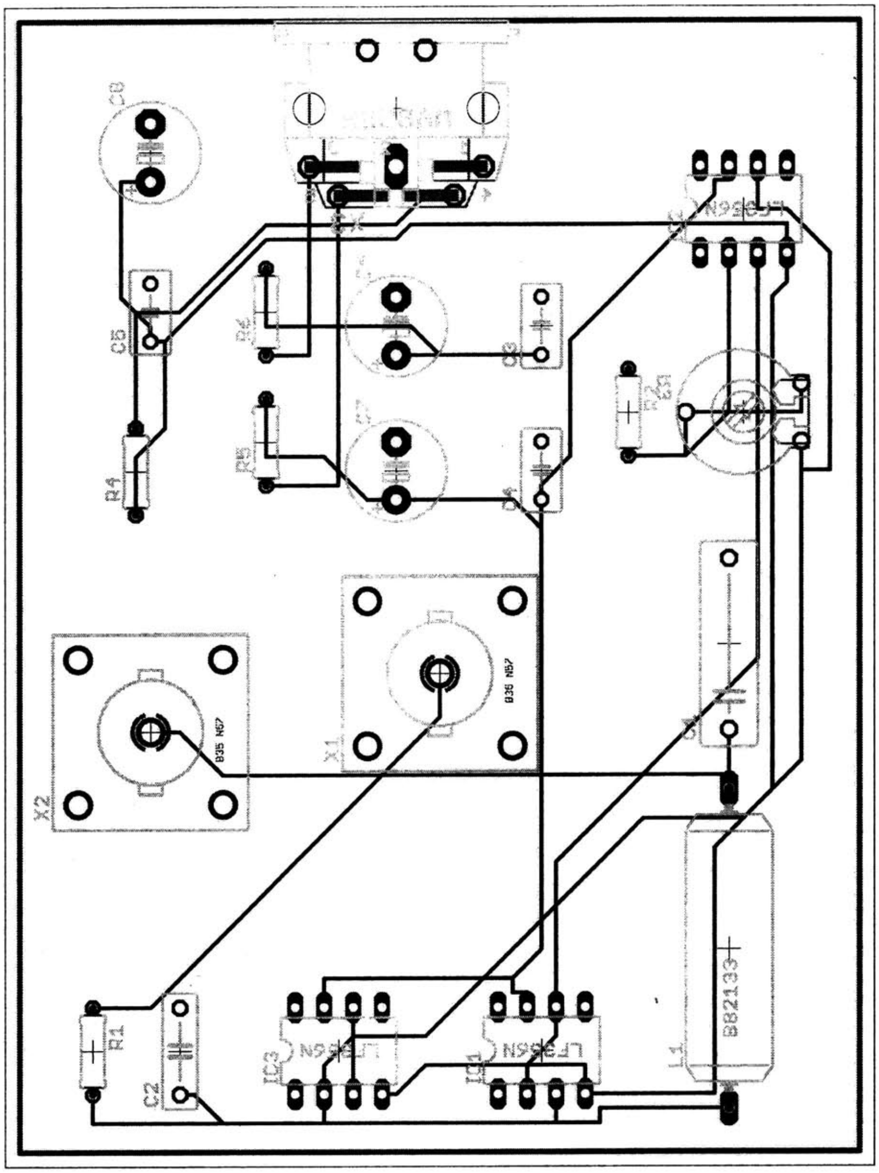
C.1 Laboratory Kit Transmission Module Layout


[redacted]

C.2 Laboratory Kit Receiver Module Layout



C.3 Pickup Circuit Layout



Appendix D

Laboratory Kit Parts List

D.1 Transmission Module Parts

Part Number	**Vendor**	**Description**	**Quantity**
ED5008-ND	Digikey	Component Sockets	65
ED3108-ND	Digikey	8-pin DIP Socket	1
ED3116-ND	Digikey	16-pin DIP Socket	1
ED3118-ND	Digikey	18-pin DIP Socket	1
ED3120-ND	Digikey	20-pin DIP Socket	1
CP-2350-ND	Digikey	Power Supply DIN Connector	1
M7837-ND	Digikey	100μH Inductor	1
3352E-501-ND	Digikey	500Ω Potentiometer	1
3352E-103-ND	Digikey	10kΩ Potentiometer	2
3352E-253-ND	Digikey	25kΩ Potentiometer	1
3352E-104-ND	Digikey	100kΩ Potentiometer	1
3352E-105-ND	Digikey	1MΩ Potentiometer	1
6209FE-ND	Digikey	DB9 to DIP serial connector	1
EG1916-ND	Digikey	SPST Sliding Switch	1
497-2539-5-ND	Digikey	TIP120 Darlington Pair	1
LF356N-ND	Digikey	LF356 Operational Amplifier	1
PIC16F628-04/P-ND	Digikey	PIC16F628 Microcontroller	1
MAX233CPP+G36-ND	Digikey	RS232 Line Transceiver	1

7006K-ND	Digikey	Binding Post, Red	3
7007K-ND	Digikey	Binding Post, Black	3
2N3904D26ZCT-ND	Digikey	2N3904 NPN Transistor	1
34972	Jameco	XR2206 Monolithic Function Generator	1
587948	Jameco	10-position Rotary Switch	1
17C6982	Newark	UBS100 Breadboard Unit	3

D.2 Reception Module Parts

Part Number	Vendor	Description	Quantity
LA-540	OS Electronics	Ferrite Loopstick Antenna	1
BC-540	OS Electronics	Poly-film Variable Tuning Capacitor	1
ED3108-ND	Digikey	8-pin DIP Socket	8
ED3116-ND	Digikey	16-pin DIP Socket	2
ED3118-ND	Digikey	18-pin DIP Socket	1
ED3120-ND	Digikey	20-pin DIP Socket	1
A24512=ND	Digikey	BNC Vertical Post, PCB Mount	1
LM386N-1-ND	Digikey	LM386 Audio Amplifier	1
LM565CN=ND	Digikey	LM565 Phase-Locked Loop	1
LM7905-ND	Digikey	7905 -5V Regulator	1
LF356N-ND	Digikey	LF356 Operational Amplifier	5
PIC16F628-04/P-ND	Digikey	PIC16F628 Microcontroller	1
MAX233CPP+G36-ND	Digikey	MAX233 RS232 Line Transceiver	1
7006K-ND	Digikey	Binding Post, Red	3
7007K-ND	Digikey	Binding Post, Black	3
2N3904FS-ND	Digikey	2N3904 NPN Transistor	9
2N3906FS-ND	Digikey	2N3906 PNP Transistor	1
3352E-201-ND	Digikey	200Ω Potentiometer	1
3352E-103-ND	Digikey	10kΩ Potentiometer	3
3352E-104-ND	Digikey	100kΩ Potentiometer	2
1N4148FS-ND	Digikey	1N4148 Diode	7
CP-2350-ND	Digikey	Power Supply DIN Connector	1
296-2052-5-ND	Digikey	CD4046BE IC Phase-Locked Loop	1
296-1577-5-ND	Digikey	74HC14 Schmitt Trigger Inverter	1
GC0251K-ND	Digikey	Compact 8Ω Speaker	1
42IF300	Mouser	IF300 Var. Transformer without Cap.	1
42IF301	Mouser	IF301 Var. Transformer with Cap.	1
42IF302	Mouser	IF302 Var. Transformer with Cap.	1

STR-92	All Electronics	Transistor Socket	8
587948	Jameco	10-position Rotary Switch	1
17C6982	Newark	UBS100 Solderless Breadboard Unit	3

D.3 Lab 5 Parts

Part Number	Vendor	Description	Quantity
CP-2350-ND	Digikey	Power Supply DIN Connector	1
LF356N-ND	Digikey	LF356 Operational Amplifier	3
A24512=ND	Digikey	BNC Vertical Post, PCB Mount	2
1N23C	Boca Semiconductor	Microwave Crystal Diode	1

Appendix E

Typical Component Values

E.1 Figure 3-2

Component	Value
R_1	8 kΩ
R_2	4 kΩ
R_3	1.2 kΩ
R_4	51 Ω
R_5	110 kΩ
R_6	200 Ω
R_E	20 Ω
$L_{ant.}$	6.8 μH

E.2 Figure 3-5

Component	Value
R_1	5.6 kΩ
R_2	5.6 kΩ
R_3	1.12 kΩ
R_4	100 kΩ
R_5	10 kΩ
R_6	100 kΩ

R_7	10 kΩ
R_{sym}	25 kΩ
R_{wav}	500 Ω
R_{sync}	10 kΩ
R_m	10 kΩ
C_1	1 nF
C_2	1 μF

E.3 Figure 3-8

Component	**Value**
R_1	5.6 kΩ
R_2	5.6 kΩ
R_3	120 kΩ
R_4	1.12 kΩ
R_5	100 kΩ
R_{sym}	25 kΩ
R_{wav}	500 Ω
R_{sync}	10 kΩ
R_m	10 kΩ
C_1	1 nF
C_2	1 μF

E.4 Figure 5-6

Component	**Value**
R_1	582 kΩ
R_2	66 kΩ
R_3	100kΩ
C_1	0.1 μF

C_2	1 nF

E.5 Figure 5-8

Component	Value
R_1	42 kΩ
R_2	130 kΩ
R_3	2.8 MΩ
R_4	100 kΩ
C_1	0.01 μF
C_2	.002 μF

E.6 Figure 5-11

Component	Value
R_0	390 kΩ
R_1	96 Ω
C	0.47 μF

E.7 Figure 6-18

Component	Value
R_f	10 kΩ
R_g	1 kΩ
R_p	100 kΩ
R_1	10 Ω
R_2	20 Ω
C	0.47 μF

E.8 Figure 7-5

Component	**Value**
R_C	10 kΩ
C_C	1o μF

Bibliography

[1] Fairchild Semiconductor Corporation, *NPN General Purpose Amplifier*, 2001.

[2] Fairchild Semiconductor Corporation, *NPN Epitaxial Darlington Transistor*, rev. a1 ed., June 2001.

[3] P. Horowitz and W. Hill, *The Art of Electronics.* Cambridge University Press, 1989.

[4] Thomas H. Lee, *The Design of CMOS Radio-Frequency Integrated Circuits.* Cambridge University Press.

[5] Allan V. Oppenheim, Alan S. Willsky, and S. Hamid Nawab, *Signals & Systems.* Prentice Hall, 1996.

[6] EXAR Corporation, 48720 Kato Road Fremont, CA 94538, *XR-2206 Monolithic Function Genrator*, June 1997.

[7] Robert W. Cox Master's thesis, Massachusetts Institute of Technology, June 2002.

[8] K. Robert Schetgen, ed., *The ARRL Handbook for Radio Amateurs.* American Radio Relay League, 1994.

[9] B. D. W. Stephen D. Senturia, *Electronic Circuits and Applications.* Krieger Publishing Co., 1993.

[10] National Semiconductor Coporation, *Low Voltage Audio Power Amplifier*, Aug. 2000.

[11] National Semiconductor Corporation, *LM565 Phase Locked Loop*, May 1999.

[12] Fairchild Semiconductor, *CD4046BC Micropower Phase-Locked Loop*, Jan. 1999.

[13] Floyd M. Gardner, *Phaselock Techniques.* John Wiley & Sons, Inc., 2005.

[14] Thomas B. Mills, *The Phase Locked Loop IC as a Communication System Building Block.* National Semiconductor Corporation, application note 46 ed., June 1971.

[15] K. J. Madhav, "A DSSS Implementation Of A CDMA System for A Developing EECS Lab Course." May 2004.

[16] Microchip Technology Incorporated, *PIC16F627A/628A/648A Datasheet*, 2004.

[17] Jin Au Kong, *Electromagnetic Wave Theory.* EMW Publishing, 2006.

[18] David H. Staelin, "Electromagnetics and Applications." 2005.