Introduction

The goal of this project is to construct a portable AM receiver that will pick up the WWV radio signal broadcasted by the National Institute of Standards and Technology in Fort Collins, CO. The broadcast information includes time announcements, standard time intervals, standard frequencies, time corrections, a binary coded decimal (BCD) time code, geophysical alerts, marine storm warnings, and Global Positioning System status reports. The times are given in coordinated universal time (UTC). The receiver I will be constructing will not only output the audio transmissions, but it will also decode the BCD time code and display the information on a LCD screen.

The proposed project has numerous applications ranging from making sure that your alarm clock is set correctly to tuning a piano (the station broadcasts a standard 440 Hz tone every hour). The device would also be extremely valuable for devices that require high accuracy time synchronization such as seismograph equipment. When an earthquake occurs, various instruments in different locations record the magnitude of the quake. The clocks of these devices must be synchronized with each other in order to accurately compare the data collected from the different areas.

General Functional Description

A basic block diagram of the receiver is shown in Figure 1. The WWV station transmits on 2.5, 5, 10, 15, and 20 MHz. The receiver will be battery powered and tuned to the 10 MHz broadcast since it is the frequency that is least affected by the time of day. The project will be divided into two sections. The first section will consist of an AM receiver, audio amplifier, active low pass filter, and pulse generator. The second section will be the decoding circuitry needed to decode and display the BCD time code. For this
application, a microcontroller will be used to provide the decoding circuitry and display operation. The first output of the AM receiver will go to a speaker so that the user may listen to the vocal announcements and audio tones of the radio broadcast. To generate the second output, the receiver will send the broadcast signal through an active low pass filter so that the BCD time code can be extracted, which will then be fed to a pulse generating circuit. The pulse generator will be used to convert the signal into a digital waveform that the microcontroller can use. The generated pulses will then be sent to the microcontroller for decoding and display.

Figure 1: Basic System Block Diagram

The BCD time code is transmitted on a 100 Hz subcarrier. The time code provides information on the current minute, hour, day of year, and year. It also indicates if daylight savings time is currently in affect. Data is transferred in serial fashion at a rate of 1 bit per second. The entire data stream is comprised of 60 bits, which means a full minute of the time code is required to decode all of the information. A 170 ms 100 Hz pulse is used to
represent a 0 bit while a 470 ms pulse is used to represent a 1 bit. The LCD screen will
display the current coordinated universal time, day, and year. The software will also
display the user’s specified time zone and “DST” if daylight savings time is in affect. The
displayed coordinated universal time will be adjusted for daylight savings time and the
user’s time zone. A sketch of the final product is shown in Figure 2.

Figure 2: Sketch of Final Product w/ Estimated Max. Dimensions

**Detailed Functional Description**

*AM Receiver:* The receiver will either be a tuned radio frequency (TRF) or super
heterodyne AM receiver. The TRF receiver is much simpler in design, but it is known to
have stability problems since all of the amplification occurs at the carrier frequency. If stability cannot be achieved, a super heterodyne receiver may be used instead. In super heterodyne receivers, most of the amplification occurs at a frequency much lower than the carrier (455 kHz), so stability is typically not a problem. The receiver will also contain an audio amplifier, which will drive a small speaker, so that the user may listen to the WWV audio broadcast. The volume of the audio will be controlled by the user with the use of an audio tapered potentiometer.

*Active Low Pass Filter and Pulse Generator:* The audio transmission will also go to the input of the active low pass filter. The filter will extract and amplify the 100 Hz pulses from the audio signal. The amplified 100 Hz signal will then go to a pulse generator circuit which will convert the sinusoidal waveform into a digital waveform. The resultant pulse widths will be equal to the duration of time that the 100 Hz signal is present on the input of the generator. The pulse generation circuitry will most likely be made up of a tone decoder phase locked loop (PLL) or the combination of an envelope detector and a voltage comparator.

*Microcontroller:* Since my project will require portability and only a single timer channel, the Motorola MC68HC908AB32 (HC08) would probably be the most appropriate microcontroller unit (MCU) for my application since it is low power (max. operating current = 30 mA) and has the input capture resources I need. It also has 1k byte of RAM, 32k bytes of FLASH memory, and 512 bytes of EEPROM so it will most likely have more than enough memory to accommodate my software.

As far as my prototype is concerned, I have decided to use a Motorola MC9S12DP256B (S12) since I will be thoroughly familiar with its resources and its
programmability after taking the ETEC 454 course during the winter quarter of 2004. My familiarity with the S12 MCU should cut down on the software development time when I begin coding for my prototype during the spring of 2004.

Figure 3 contains a detailed block diagram of the project hardware.

**Figure 3: Block Diagram of Project Hardware**

- **Antenna**
- **+6V**
- **TRF Receiver**
- **Audio Amp.**
- **Speaker**
- **Active Low Pass Filter**
- **Pulse Generator**
- **MC9S12DP256B**
- **Port K PK0-PK2**
- **Port A PA0-PA7**
- **Pulse Generator**
- **Port T PT0**
- **Push Button (Time Zone Selection)**
- **Reset Circuitry**
- **BDM Connector**
- **16 MHz Crystal**
- **+6V Supply**
- **4 AA Batteries**
- **Approx. Capacity = 2250 mAh**
- **LCD**
- **SPST Switch**
- **+6V**
- **12k Byte RAM**
- **256k Byte FLASH ROM**
- **4K Byte EEPROM**
- **BDM IN**
- **EXTAL XTAL**
- **Reset**
A single timer channel in Port T will be used to measure the pulse widths created by the pulse generator, which will be used for the decoding of the BCD time code. Ports A and K will control the LCD screen while Port A will be used to monitor button presses from the user to change the desired time zone. A 16 MHz crystal will clock the MCU and a single-pole-single-throw (SPST) switch will be used to connect the circuit to the power supply. The power supply will consist of 4 AA alkaline batteries.

**Software Requirements:**

The source code for the project will most likely be done using the C programming language since it will be the language I will be most familiar with when I begin coding. However, if the need for more efficient code arises, assembly language may be used. I will be using a µC/OS pre-emptive kernel for my task executions, unless I exceed timing constraints, in which case, a time slice scheduler may be more appropriate.

The software will consist of the following modules:

**KERNEL:** This will be a µC/OS pre-emptive kernel (or time slice scheduler if needed) that will control all the task executions in the software.

**DEBOUNCE:** This will provide the switch debouncing task for the push button switch used to change the desired time zone.

**LCD:** This will perform all the LCD screen initialization and communicate data to the LCD screen.

**MAIN:** This will be the most important part of the software. It will perform all the valid time code signal detection and decoding.
**User Interface:**

The user interface will be composed of a single LCD screen and two buttons. One button will be used to connect the device to power, while the other will allow the user to select the desired time zone to be displayed. Figure 4 contains a detailed sketch of the user interface.

![Figure 4: Sketch of User Interface](image)

The interface will have three main states. The first one will occur when the device is initially powered up. It will indicate to the user that reception of the signal has not yet been acquired. The next state will occur when the signal is successfully acquired. At this point, the user will be instructed to wait while the signal is decoded (1 complete minute of uninterrupted time code will be needed to decode and display the current time). If at any point the data stream is interrupted through the loss of reception, the display will return to the first state notifying the user that reception was lost. The final state will occur
when the signal is successfully decoded and displayed on the LCD. At this point, the user may select the desired time zone with the use of the Time Zone button. The displayed time will be adjusted according to the selected time zone and will be updated every minute. Figure 5 illustrates the displays of these various states.

**Figure 5: LCD Displays for Each State**

![Diagram showing LCD displays for each state: No Reception, Signal Acquired, Signal Lost, Please Wait, Signal Decoded, T: 1200 PST DST, DAY 365 of 2003.]

**Development Plan:**

I expect that the design and testing of the AM receiver will be the most difficult portion of this project, so I will be setting a large portion of my development time towards the design and stability testing of this circuitry. Since TRF receivers require amplifiers with large amounts of gain at high frequencies, they are very prone to oscillations. Just as much time, if not more, will need to be spent on proper circuit layout.
as on the actual design of the receiver to ensure stability. Since most of the TRF receiver will be operating at a high frequency (10MHz), breadboarding will probably not be a suitable format for circuit testing due to the stray capacitances associated with breadboards at high frequencies. Instead, I will need to use a combination of point-to-point soldering and wire wrapping to achieve stable test circuits. Unfortunately, this will also substantially slow down my development time. As mentioned before, if a stable TRF receiver is not attainable, then a super heterodyne may be constructed instead.

The tuned amplifier stages of the receiver will first be tested individually with the use of a function generator with a 10 MHz (and 455 kHz if a super heterodyne is used) output signal to ensure that each stage is providing the necessary amplification for the desired frequency. The AM detector of the receiver will be tested with the use of an amplitude modulated test signal also provided by a function generator. The audio amplifier, low pass filter, and pulse generator sections will be tested with a similar method as the tuned amplifier stages. The only exception is that the low pass filter section will be designed so that the frequency of interest is 100 Hz. The testing of the filter and pulse generator will be conducted by using an audio recording of the WWV broadcast as the input. The outputs of the filter and pulse generator will be examined to ensure that the amplitude of the 100 Hz pulses are adequate for use with the microcontroller and that there are no undesired frequencies present on the output waveforms. Once the operation of each stage of the circuit is verified, they will be connected together in their corresponding places of the overall circuit. An antenna will then be attached to the circuit and the receiver will be taken outdoors for reception testing. By attaching the antenna, the center frequency of the first tuned amplifier stage will most likely change since the
antenna has a small amount of capacitance associated with it. Therefore certain component values will need to be changed in this stage to compensate for this shift in frequency.

Since WWV is a shortwave radio signal, its reception is dependent on several factors including atmospheric conditions, solar flares, sunspot activity, and the construction of the building that the user is trying to receive the signal in. Since it is nearly impossible for me to receive the signal in the laboratory (due to the construction of the building), the development of the decoding software will be done through the use of an audio recording of the broadcast. This will ensure that I always have a stable signal to work with when I begin writing and debugging my software.

Most of the development and testing will take place in the ET340 laboratory. Reception testing will need to take place outdoors since I cannot receive the signal within the laboratory. For the software development, I will be using the CodeWright system and the Noral debugging POD. Other hardware and software testing will be performed with the use of various oscilloscopes, function generators, and multi-meters located in ET340. A weekly project development schedule for the remainder of the academic year can be found in Figure 6.

**Figure 6: Weekly Project Development Schedule**

**Winter Break of 2003:**

*Week 1:* Receiver design and testing

*Week 2:* Receiver design and testing

*Week 3:* Receiver design and testing

*Week 4:* Receiver design and testing
**Winter Quarter of 2004:**

*Week 1:* Reception testing  
*Week 2:* Reception testing  
*Week 3:* Active low pass filter design and testing  
*Week 4:* Active low pass filter design and testing  
*Week 5:* Pulse generator design and testing  
*Week 6:* Pulse generator design and testing  
*Week 7:* Pulse generator design and testing  
*Week 8:* Initial software design  
*Week 9:* Initial software design  
*Week 10:* Dead Week

**Spring Quarter of 2004:**

*Week 1:* Modify pre-emptive kernel  
*Week 2:* Modify pre-emptive kernel  
*Week 3:* Write DEBOUNCE module/Hardware review  
*Week 4:* Modify LCD module (if necessary)/Begin writing MAIN module  
*Week 5:* Write MAIN module/Software system presentation  
*Week 6:* Write MAIN module  
*Week 7:* Write MAIN module  
*Week 8:* System Testing  
*Week 9:* Code review  
*Week 10:* Final presentation
Prototype Demonstration:

The demonstrated prototype will consist of a soldered PC board and a development board which will all be contained within the enclosure shown in Figure 2. The receiver, audio amplifier, low pass filter, and pulse generator will all be located on the PC board. All the connections on this board will made with a combination of point to point soldering and wire wrapping. All of the remaining hardware (including the MCU, memory, crystal, etc.) will be located either on the development board or within the MCU itself.

For the prototype demonstration, I will use an audio recording of the WWV broadcast to demonstrate the decoding and display of the device since I cannot receive the signal within the laboratory. I may also use a video recording of the prototype operating outdoors to prove the functionality of the entire system.

Electrical Specifications:

Project Specifications

Designed Receiver Sensitivity: 50µV

Displayed Clock Format: 24 hour

Displayed Clock Resolution: 1 minute

Standard Carrier Frequency of Broadcast: 10 MHz

Standard Audio Tones of Broadcast: 100, 440, 500, and 600 Hz

Standard Displayed Time: Coordinated Universal Time

Time Synchronization Uncertainty: 1 ms

Max. Audio Output Level: 12 dB
### Power Requirements

**Source:** 4 AA Alkaline Batteries

**Approx. Capacity:** 2250 mAh

**Estimated Battery Life:** 5.6 hours

**Worst Case Power Dissipation:** 2.4W

### Environmental Requirements

**Operating Temperature Range:** -40º to 85º C

### PCB Size Limits

**Max. PCB Dimensions:** 8” x 4” x 2”

### Preliminary Part List:

<table>
<thead>
<tr>
<th>Qty</th>
<th>Description</th>
<th>Part #</th>
<th>Source</th>
<th>Price</th>
<th>Lead Time</th>
<th>Worst Case Current</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Microcontroller</td>
<td>MC9S12DP256B</td>
<td>Arrow Elect.</td>
<td>$20.30</td>
<td>4 weeks</td>
<td>65 mA</td>
</tr>
<tr>
<td>1</td>
<td>Tone Decoder/PLL</td>
<td>LM567</td>
<td>National Semi.</td>
<td>$0.21</td>
<td>4-6 weeks</td>
<td>10 mA</td>
</tr>
<tr>
<td>1</td>
<td>JFET RF Transistor</td>
<td>2N5484</td>
<td>Mouser</td>
<td>$0.15</td>
<td>Have</td>
<td>1 mA</td>
</tr>
<tr>
<td>2</td>
<td>NPN RF Transistor</td>
<td>MPS5179</td>
<td>Mouser</td>
<td>$0.65*2</td>
<td>Have</td>
<td>1mA*2</td>
</tr>
<tr>
<td>1</td>
<td>10.7 MHz IF Xfmr</td>
<td>42IF123</td>
<td>Mouser</td>
<td>$0.79</td>
<td>Have</td>
<td>1 mA</td>
</tr>
<tr>
<td>3</td>
<td>10.7 MHz IF Xfmr</td>
<td>42IF122</td>
<td>Mouser</td>
<td>$0.79*3</td>
<td>Have</td>
<td>1mA*3</td>
</tr>
<tr>
<td>2</td>
<td>35.3k ohm Resistor</td>
<td>Resistor</td>
<td>Digikey</td>
<td>$0.06*2</td>
<td>Have</td>
<td>170 µA*2</td>
</tr>
<tr>
<td>2</td>
<td>16.2k ohm Resistor</td>
<td>Resistor</td>
<td>Digikey</td>
<td>$0.06*2</td>
<td>Have</td>
<td>370 µA*2</td>
</tr>
<tr>
<td>2</td>
<td>1k ohm Resistor</td>
<td>Resistor</td>
<td>Digikey</td>
<td>$0.06*2</td>
<td>Have</td>
<td>6 mA*2</td>
</tr>
<tr>
<td>2</td>
<td>75 ohm Resistor</td>
<td>Resistor</td>
<td>Digikey</td>
<td>$0.06*2</td>
<td>Have</td>
<td>80 mA*2</td>
</tr>
<tr>
<td>1</td>
<td>1.18k ohm Resistor</td>
<td>Resistor</td>
<td>Digikey</td>
<td>$0.06</td>
<td>Have</td>
<td>5.1 mA</td>
</tr>
<tr>
<td>1</td>
<td>1M ohm Resistor</td>
<td>Resistor</td>
<td>Digikey</td>
<td>$0.06</td>
<td>Have</td>
<td>6 µA</td>
</tr>
<tr>
<td>10</td>
<td>0.01 µF capacitor</td>
<td>Capacitor</td>
<td>All Electronics</td>
<td>$0.05*10</td>
<td>Have</td>
<td>N/A</td>
</tr>
<tr>
<td>3</td>
<td>100 pF capacitor</td>
<td>Capacitor</td>
<td>All Electronics</td>
<td>$0.05*3</td>
<td>Have</td>
<td>N/A</td>
</tr>
<tr>
<td>3</td>
<td>2-18 pF trimmer capacitor</td>
<td>Trimmer Capacitor</td>
<td>All Electronics</td>
<td>$0.5*3</td>
<td>Have</td>
<td>N/A</td>
</tr>
<tr>
<td>1</td>
<td>Dual Audio Op-Amp</td>
<td>LM833</td>
<td>National Semi.</td>
<td>$0.32</td>
<td>Have</td>
<td>5 mA</td>
</tr>
<tr>
<td>1</td>
<td>Low Voltage Audio Op-Amp</td>
<td>LM386</td>
<td>National Semi.</td>
<td>$0.15</td>
<td>4-6 weeks</td>
<td>4 mA</td>
</tr>
<tr>
<td>1</td>
<td>10k ohm Audio Taper Pot.</td>
<td>Potentiometer</td>
<td>All Electronics</td>
<td>$0.50</td>
<td>1 week</td>
<td>600 µA</td>
</tr>
<tr>
<td>1</td>
<td>SPST Slide Switch</td>
<td>SSW-222</td>
<td>All Electronics</td>
<td>$0.33</td>
<td>Have</td>
<td>N/A</td>
</tr>
<tr>
<td>1</td>
<td>Push Button Switch</td>
<td>103-1112</td>
<td>Mouser</td>
<td>$0.83</td>
<td>1 week</td>
<td>N/A</td>
</tr>
<tr>
<td>1</td>
<td>32 Ohm Speaker</td>
<td>SK-232</td>
<td>All Electronics</td>
<td>$0.75</td>
<td>1 week</td>
<td>110 mA</td>
</tr>
<tr>
<td>1</td>
<td>LCD</td>
<td>LM1180</td>
<td>Pencom Design</td>
<td>$8.00</td>
<td>1 week</td>
<td>3 mA</td>
</tr>
<tr>
<td>1</td>
<td>16 MHz Crystal</td>
<td>XC271-ND</td>
<td>Digikey</td>
<td>$2.63</td>
<td>1 week</td>
<td>17 mA</td>
</tr>
<tr>
<td>1</td>
<td>4 AA Battery Holder</td>
<td>BH-342</td>
<td>All Electronics</td>
<td>$0.80</td>
<td>Have</td>
<td>N/A</td>
</tr>
<tr>
<td>Item</td>
<td>Quantity</td>
<td>Part Number</td>
<td>Supplier</td>
<td>Price</td>
<td>In Stock</td>
<td>Lead Time</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>----------</td>
<td>-------------</td>
<td>-------------</td>
<td>---------</td>
<td>----------</td>
<td>-----------</td>
</tr>
<tr>
<td>1/8” Mono Phone Plug</td>
<td>1</td>
<td>274-286</td>
<td>Radio Shack</td>
<td>$2.99</td>
<td>Have</td>
<td>N/A</td>
</tr>
<tr>
<td>1/8” Mono Phone Jack</td>
<td>1</td>
<td>274-251</td>
<td>Radio Shack</td>
<td>$2.99</td>
<td>Have</td>
<td>N/A</td>
</tr>
<tr>
<td>Perf Board</td>
<td>1</td>
<td>PC-4</td>
<td>All Electronics</td>
<td>$2.00</td>
<td>Have</td>
<td>N/A</td>
</tr>
<tr>
<td>Energizer e^22 Batteries</td>
<td>4</td>
<td>X91</td>
<td>Best Buy</td>
<td>$4.99</td>
<td>Have</td>
<td>N/A</td>
</tr>
<tr>
<td>47 µF Electrolytic Capacitor</td>
<td>3</td>
<td>Capacitor</td>
<td>All Electronics</td>
<td>$0.15*3</td>
<td>1 week</td>
<td>N/A</td>
</tr>
<tr>
<td>Reset Circuit</td>
<td>1</td>
<td>MAX6314</td>
<td>Digikey</td>
<td>$1.04</td>
<td>1 week</td>
<td>12 µA</td>
</tr>
<tr>
<td>6 pin BDM Connector</td>
<td>1</td>
<td>115035</td>
<td>Jameco</td>
<td>$0.18</td>
<td>1 week</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Total: $56.82

Total Worst Case Power Dissipation = 2.4 W