Recipedia

The Digital Cookbook

Team Members
Kevin Quinn
Tim Spens
Koto Norose
Shawn Rhoades

Introduction

Our project consists of designing and implementing a digital cookbook for use in a kitchen/restaurant environment. This system will provide a convenient and easy to use interface that any person preparing a meal can follow. Through the use of a touch screen the user will be able to input commands directly by simply using an easy to follow graphical user interface (GUI). The system will also be capable of receiving simple voice commands such as Next, Previous, etc. to provide hands-free operation during meal preparation. As an added feature, the system will be capable of reading the recipe out loud to the user to provide added convenience. Time permitting the system will comprise of various features such as a calendar, time/date, timer, and USB expansion. Below is a summary all aspects of our project including our implementation strategies, timelines, schedule flow, interface specifications, and risks involved.

Motivation for Project

For our group this project intrigues our interests in areas such as voice recognition, touch screen applications, hardware interfaces, and software developments. We are all enthusiastic and willing to learn about areas that may be unfamiliar to us in order to complete this design. In the end, we would hope that our design will produce a convenient and marketable digital cookbook system that could be of use in all areas of food preparation.

System Overview

The system we plan to implement is built upon several hardware components and supported by software that controls various aspects of the design. The processing and control for our system is maintained by two embedded Motorola MC68HC11 processors. One controller will maintain all application software as well as hardware interfaces between most of our other hardware. The other will be used as an auxiliary controller to control and process voice signals through its analog-to-digital converter as an added feature of our system design. The user interface for our project will consist of a touch screen overlay installed into a 17" CRT monitor. The user will be able to input commands to the system by simply touching the buttons located on the screen that correspond to that command. We plan to use an FPGA programmable development board as a VGA controller between our monitor and our main processor that handles all inputs and outputs from and to the screen. The system will be capable of outputting recipes and other useful knowledge in the form of audio speech via a text-to-speech module. A more detailed description of our implementation strategies and hardware is located below.

Components & Interfaces

The sections below will discuss the hardware and software design, implementations and interface specifications for Recipedia. Each interface (hardware and software) will be explained in detail.

Micro-controller (2) / FPGA

At the heart of the system will be the MC68HC11E1CP2 microcontroller. This will be where the main software for Recipedia will be stored and running along with the control for the text-to-speech chip, Spartan3 FPGA, and it will also receive inputs from the touch screen overlay. The main software routine will send commands to the FPGA to update and display recipes on the CRT monitor. These commands will be simple numbers sent out from the microcontroller telling the FPGA which recipe to display from a set of recipes. The FPGA will update the CRT monitor with a recipe using a character ROM. The microcontroller will also be controlling the text-to-speech chip using TTL level SCI protocol with a configurable baud rate, 8 data bits, no parity, and 1 stop bit.

Another microcontroller will be used to capture voice commands from the user via a microphone. We will capture the data at 4-8 kHz. The data will be saved in 256-bytes of RAM temporarily for analyzing. By using frequency matching on the voice commands “home”, “next” and “back” we hope to match these commands to a known pattern. Once we have a match a one-hot encoded message will be sent directly from the command recognition microcontroller to the main microcontroller so the next or previous recipe line can be read or the CRT display can be updated.

Microcontroller Specifications

Requires +5V DC supply @ 200mA
512bytes EEPROM
512bytes RAM
1-3MHz (dependent upon external crystal)
Rich instruction set
SCI interface (baud rate and mode configurable)
8 bit ADC
Many interrupt sources
Two 8-bit registers
Two 16-bit index registers
11 output ports, 11 output ports and 16 configurable I/O ports

FPGA Specifications

Comes with a +5V DC adapter
200,000-gate Xilinx Spartan-3 XC3S200 FPGA (XC3S200FT256)
4,320 logic cell equivalents
Twelve 18K-bit block RAMs (216K bits)
2M-bit PROM
1M-byte SRAM

Text-to-Speech Interface

The SP03 module will receive ASCII characters from the main microcontroller serially. The chip will "speak" the strings of ASCII characters that it receives. The microcontroller will send ASCII characters one recipe line at a time until the user requests the next line or the previous line. The volume, pitch and speed can also be controlled from the main microcontroller via the serial interface. It is pre-programmable with up to thirty phrases. These phrases are stored in the Flash memory of the PIC16F872 processor located on the module.

SP03 Specifications

Requires +5V DC supply @ 70mA
Serial TTL interface (2-wire, configurable baud rate)
On-board 8 Ω, 300mW (SIP) or 23.5mW (OEM) speaker driver
Easy-to-use ASCII or hexadecimal command sequences
Bi-color LED for visual indication of activity
0.100. Pin spacing for easy prototyping and integration

Touch Screen Overlay

Because of cost constraints we are using a touch screen overlay on a CRT monitor instead of a LCD touch screen. We will be using the ELO E274 touch screen with the AccuTouch 2210 controller. This has a serial interface for our main microcontroller to capture the touch coordinates.

The controller communicates by sending 8-byte packets. The packets are transmitted and received through eight consecutive read/write I/O ports. The first byte of each packet is the command byte, and the seven remaining bytes are the data bytes. The command byte is an ASCII character, from 'A' to 'T'. A command byte in upper-case indicates a set command to the controller. The data bytes then alter an internal setting of the controller. A command byte in lower-case indicates a query command to the controller.

A typical query/response/set interaction flows as follows:

1. Host sends query command packet
2. Controller sends a response packet
3. Controller sends an Acknowledge response packet
4. Host sends a set command packet
5. Controller sends an Acknowledge response packet

The "T" in byte 0 indicates the packets are Touch packets. Byte 1 contains the Status bits, the X coordinate is the Intel (byte swapped) integer formed by bytes 2 & 3, Y is in bytes 4 & 5, and Z follows in bytes 6 & 7. As you move your finger, bytes 2-5 will change. Byte 1 indicates your initial touch, stream touches, and un-touch with 1, 2, and 4 respectively. These values correspond to the bit positions defined for the Touch packet. Bytes 6 & 7 are constant on AccuTouch controllers as they do not support a Z-axis.

We will be writing a low level device driver for this controller for the main microprocessor. AccuTouch provides C code for a low level driver. This code will be used to write our driver in assembly, due to the fact that a compiler will generate code too large and will not fit in our microcontrollers' limited memory.

AccuTouch Specifications

Requires +5V DC supply @ 60ma
Operates at standard RS232C levels
Configurable baud rate

Testing and Integration Strategies

For our project testing strategies we plan to test extensively. At each stage of our design, we plan to test the component, software, or interface that we complete. Both standalone testing as well as integrated tests will be conducted. As a result of this we can safely assume that our design is correctly operating with each new component or software routine that we implement. Each team member will be responsible for testing each component they are committed to. In some cases where more than one team member is working on an aspect of the project the project lead will be responsible to ensure that all members are correctly testing and interfacing their current work. As each milestone is completed, testing once again will be conducted. This is to ensure that our current stage of the process is compatible with all previously completed processes. As a result of this extensive testing, we alleviate some of the stress of testing all components at once after and to affirm that our project is producing desired results.

Group Communication Plan

We will communicate via meetings, email, and the phone. Meetings will be crucial when we work with hardware aspects of our project. It will be necessary to discuss troubleshooting strategies and also share experiences. In other words, working on hardware with a teammate will be necessary to multi-task. Email will be used for sending application code, updates, etc. to each other. It is useful when our schedule does not coincide. Also we will decide when we will meet via email. Phone will be used as a means of emergency contact. For example when we have some trouble in the lab and we need to consult with another team member we will use the phone.

We are considering that we will meet during the summer often to work on hardware aspects of our project. We need to meet in the lab to use equipment such as oscilloscopes, voltage supplies, and frequency generators. If a team member is unavailable to meet they will work on software simulation or other tasks such as gathering the information of parts from a datasheet, making documentation, etc. During the fall we will meet more often as we attend the regular class. On the weekdays we plan on working together in the laboratory to test hardware and on the weekends we will work on mostly software and occasionally meet.

We will use the communication skills gained from other classes, such as technical writing, etc. We will try to solve problems that arise within the group by communicating with each other. Sometimes we may ask the communication students who are working in the engineering department for help with our communication skills such as how to make our presentations more efficient, etc.

We may also communicate with professors for advice. We do not have any specified technical advisors but will not hesitate to ask advice to faculty members that may be able to assist us.

Schedule Flow

Our schedule and milestones for our design process is located below. The Gantt chart below illustrates milestones and a timeline for our project design. Each of our project leaders for the different sections will be in charge of meeting deadlines associated with each milestone. The following table illustrates our prospective milestones we will accomplish. These are the milestones we feel are necessary for us to be able to finish our project in a timely manner.

Completion Date	***Milestone***
May 15th, 2006	• ***All parts acquired and tested***
June 15th, 2006	• ***Schematics of complete system compiled*** • ***Pseudo-code written for software***
July 15th, 2006	• ***5V 2A power supply complete*** • ***Software & interface for text-to-speech***
October 10th, 2006	• ***VGA and character ROM implemented via Spartan3 FPGA w/interface to µController***
October 20th, 2006	• ***Analog components complete: amplifier, filter, etc.*** • ***System enclosure complete***
October 30th, 2006	• ***Software & interface for touch screen & application software complete***
November 30th, 2006	• ***Extra features complete: command recognition, timer, calendar, USB interface, etc.***
December 7th, 2006	• ***System fully operational*** • ***Ready for presentation & demonstration***

Table 1. Schedule and Milestones

ID	Task Name	Start	Finish	Duration
1	Acquire All Parts	5/5/2006	5/15/2006	7d
2	Schematics of Complete System	5/5/2006	6/15/2006	30d
3	Pseudocode Written	5/5/2006	6/15/2006	30d
4	Power Supply	5/5/2006	7/14/2006	51d
5	Software & Interface Text-to-Speech	5/5/2006	7/14/2006	51d
6	VGA & Character ROM	7/17/2006	10/10/2006	62d
7	Analog Components	7/17/2006	10/20/2006	70d
8	System Enclosure	7/17/2006	10/20/2006	70d
9	Touch Screen Software & Interface	7/17/2006	10/30/2006	76d
10	Extra Features	9/1/2006	11/30/2006	65d
11	System Testing	7/14/2006	12/7/2006	105d

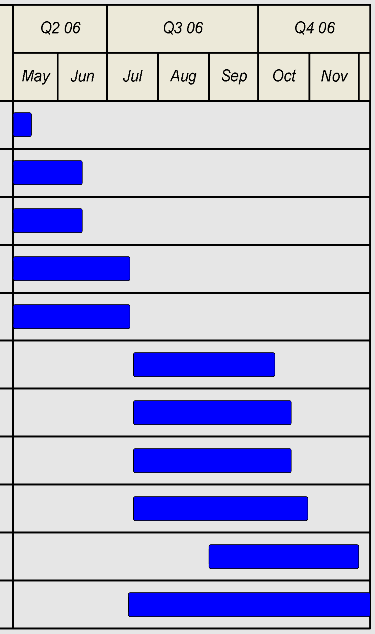


Table 2. Gantt chart

As illustrated by the chart above, we plan to work on many aspects of our project in parallel to ensure adequate time to complete them. Also testing will begin as soon as our power supply is built and will continue until completion of our system.

Tasks Delegation

For the tasking of our project we decided we will break up major sections into time slots. We will work together in groups of 1-4 on each section. For each section we will assign a lead who will be in charge of setting times for us to meet, and keep us on schedule for that specific component design. We have created a colored diagram to show the different tasks which each person will be responsible for. Each of us will be involved in most of the areas of the project, but some key areas will be assigned to individuals and smaller groups for a more efficient process.

As noted in the chart below, the biggest areas such as schematics, writing pseudo code, testing, and development of extras (which includes voice recognition) will be done with all team members meeting and working together. We have assigned project leads for each section who will take charge of the scheduling for that section and keeping members on track.

Our other sections will be done in smaller groups of 1 or 2 people. Kevin will be in charge of our power supply and getting the power working for our project. He will also be in charge of developing and learning how to use our touch screen including the software development pertaining to it. Koto and Kevin will be handling the analog components and devices. This includes devices that filter or amplify incoming or outgoing signals. Koto will be lead of that section. Tim and Shawn will be working together to develop the text-to-speech software and interfaces as well as incorporating it into our design. Shawn will be leading this section of the project. Finally, Tim will be handling the VGA and Character ROM for our FPGA and will also be in charge of the system enclosure as he has connections with equipment that can produce what we need. The table below illustrates our tasks and leads for each design aspect.

Task	***Kevin***	***Koto***	***Tim***	***Shawn***
Schematics	Lead			
Pseudo code		Lead		
Power Supply	Lead			
Text_to_Speech (software & interface)				Lead
VGA & Character ROM (FPGA Spartan 3)			Lead	
Analog Components		Lead		
System Enclosure			Lead	
Touch Screen Software & Interface	Lead			
Extra Features				Lead
System Testing			Lead	

Table 3. Tasks

We plan to follow our time schedule closely. The different leaders for each section of the project focus on the deadlines and ensure that the rest of the group stays on track and on time.

Risks and Mitigation Plans

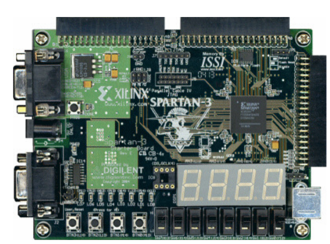
Some of the risks involved in our project consist of a lack of experience with certain aspects of the project components, pricing of parts, understanding datasheet specifications for components, and complexity of software we will need to develop.

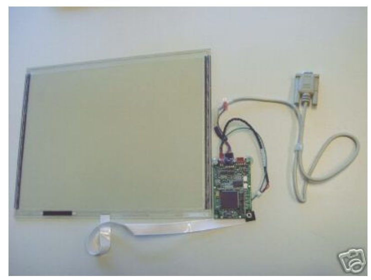
First it will be necessary to learn how the hardware interfaces to other hardware. For example how the touch screen interfaces produce usable outputs, conversion of voice signals to digital outputs, and hardware drivers when necessary for hardware components. We are willing to learn new things and overcome this risk.

Second, there is a considerable complexity of the software needed for our project. Developing software will be endless. Debugging will possibly take most of our time. Also the programmable memory could be too small for our project and we will have to write very efficient code. To build the database and to gather its data (recipes) will take time since a certain amount of data will be necessary. Consequently, we need to write an efficient database due to the limited memory space also. Our group members are relatively knowledgeable programmers but a little hesitant with the timeframe in order to complete a large-scale program and database implementations. We will spend our summer to prepare for this risk.

Finalized Bill of Materials (BOM)

Our project will consist of various hardware and software components. Some of the components will be available for purchase while others we will design ourselves. We plan on purchasing the touch screen, either mountable to an existing display or a stand-alone monitor, a microcontroller capable of handling software applications, A/D converter, text-to-speech chip, small hard drive, microphone, and speaker. The design aspects include application software, hardware driver software, power supply, analog-to-digital conversion and text-to-speech circuitry. Below is a list of purchased components that we plan to use in the design of our project.

Part	Picture	Total Costs:
Embedded Processors Primary Vendor: University of Utah Model #: 68HC11 Part #: MC68HC11E1CP2 Unit Cost: $13.50 (Own 2) Quantity: 2		$0.00
FPGA Primary Vendor: www.digilentinc.com Model #Spatan3 starter kit Part #: XCS3S200 Unit Cost: $99.00 (Own 2) Quantity: 2		$99.0 + shipping
Text to Speech Chip Primary Vendor: hobbyengineering.com Model #: SPO3 Part #: R184SP03 Lead Time: None (In-stock) Unit Cost: $99.00 Quantity: 1		$99.00 + shipping

Secondary Vendor: acroname.com Model #: SPO3 Part #: R184SP03 Lead Time: None (In-stock) Unit Cost: $102.00 Quantity: 1		
Touch Screen Overlay Primary Vendor: ebay.com Model #: SCN-AT Part #: E274 Lead Time: None (Purchased) Unit Cost: $56.00 Quantity: 1 Secondary Vendor: elo.com Model #: SCN-AT Part #: E274 Lead Time: None (Purchased) Unit Cost: $199.00 Quantity: 1		$56.00 + shipping
CRT 15" Monitor *ViewSonic OptiQuest Q71 CRT Monitor* Primary Vendor: Deseret Industries (DI) Model #: VCDTS21348-2M Lead Time: None (Purchased) Unit Cost: $10.00 Quantity: 1 Total Cost:		$10.00
Miscellaneous Analog/Digital Parts Primary Vendor: Raelco Model #: ANY Part #: ANY Lead Time: None (Purchased) Unit Cost: N/A		N/A

Quantity: N/A Secondary Vendor: UofU Model #: ANY Part #: ANY Lead Time: None (Purchased) Unit Cost: N/A Quantity: N/A		
FINAL COSTS INVOLVED		$279.51

Table 4. Bill of Materials & Vendors

System Enclosure

For our system's enclosure we will construct a large box shaped structure assembled from aluminum, Lexan, or Plexiglas material adequate enough to enclose the large CRT monitor used in our project, circuit boards, etc. A simple diagram of our enclosure is located in appendix A of this report.

Appendix A – Diagrams

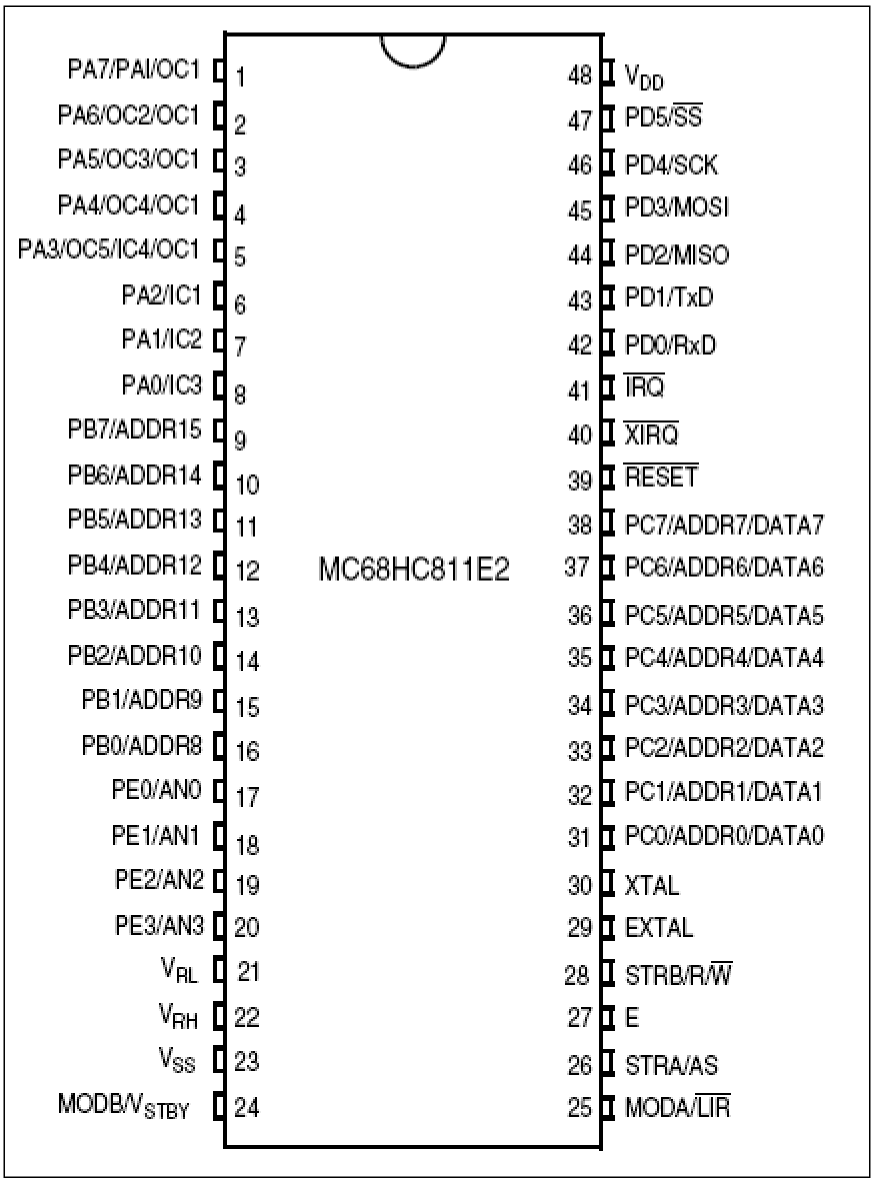


Diagram 1. MC68HC11E2 pin assignments

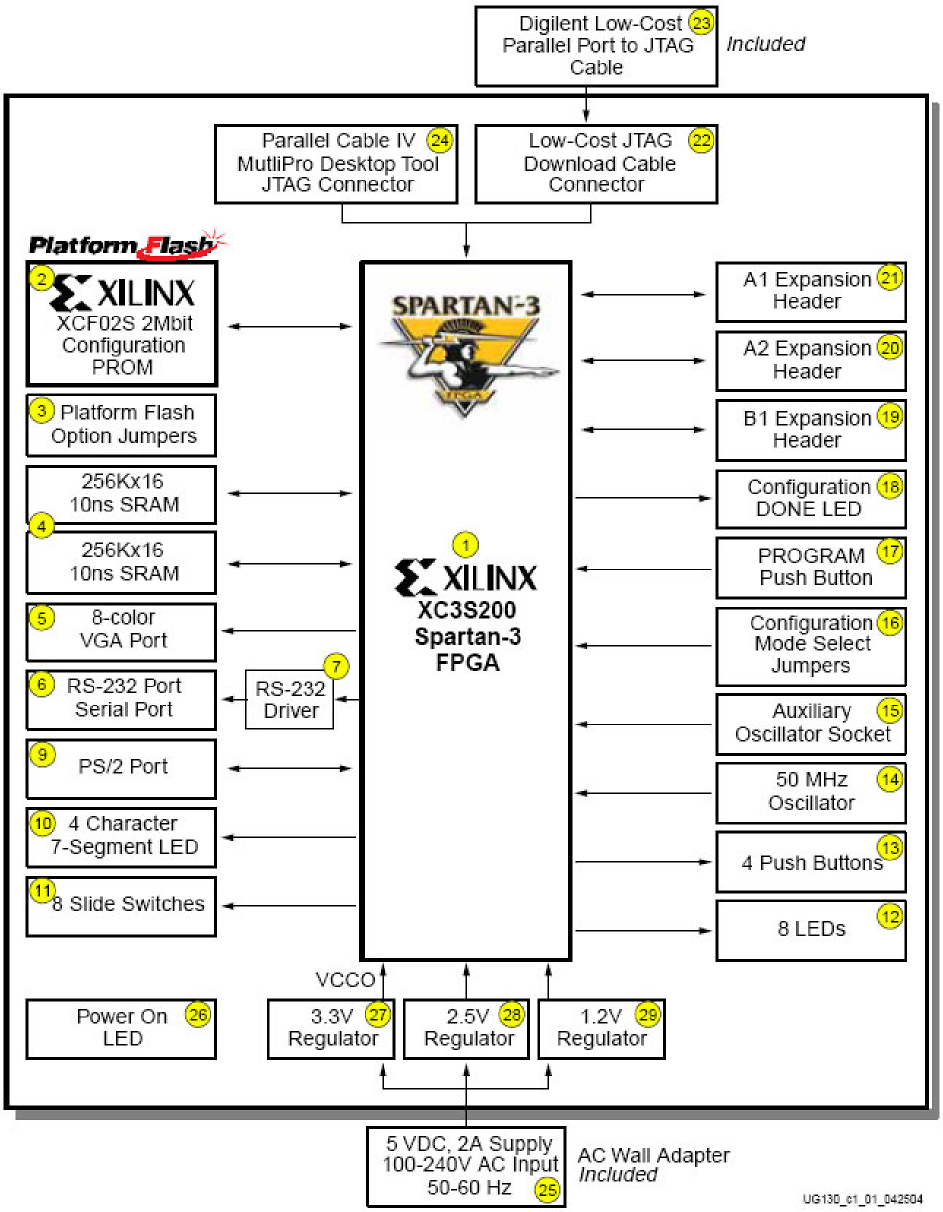


Diagram 2. Spartan 3 FPGA block diagram

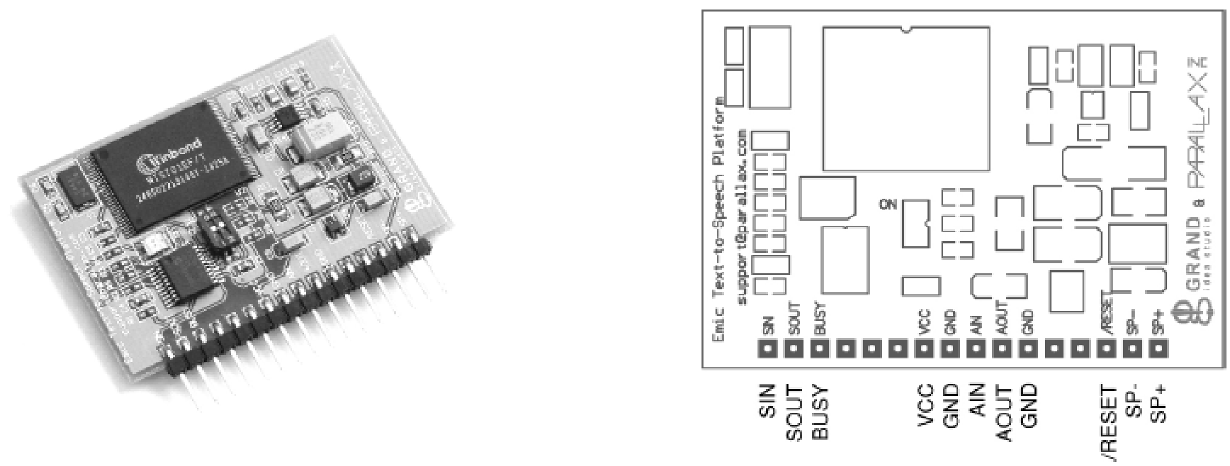


Diagram 3. SPO3 Text-to-Speech module

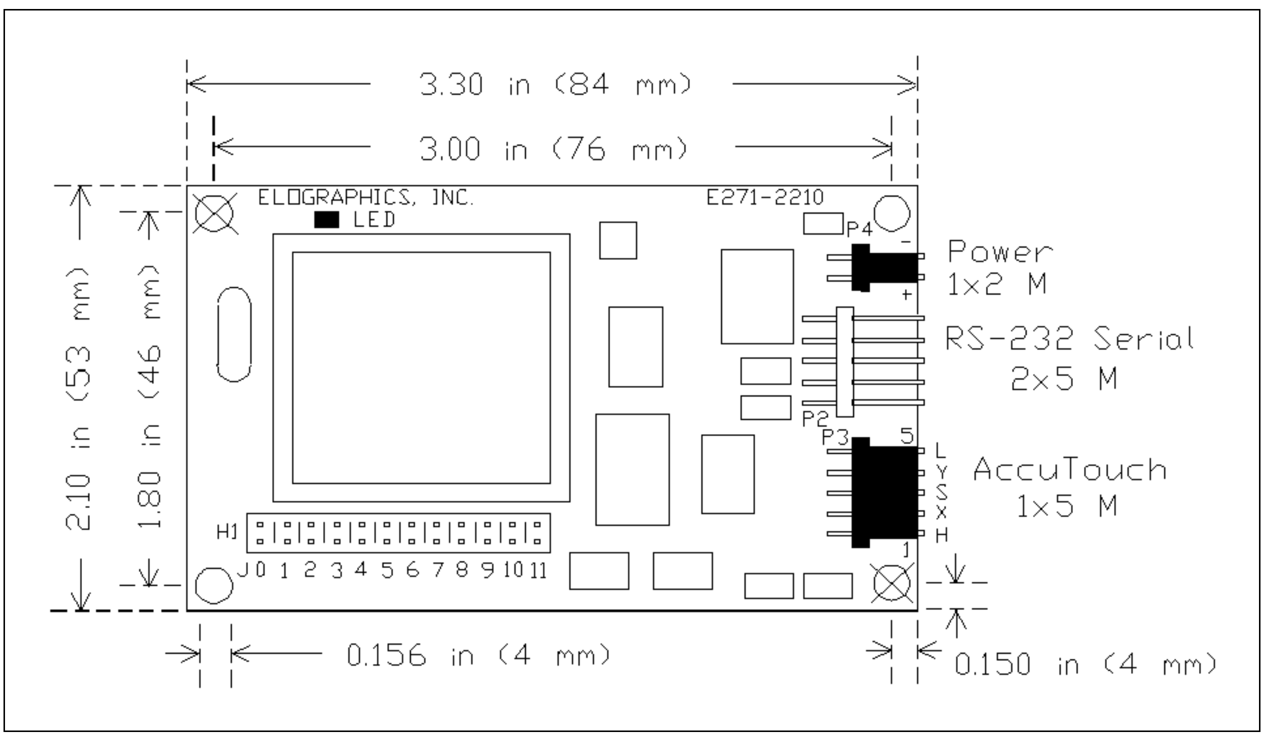


Diagram 4. AccuTouch 2210 touch controller

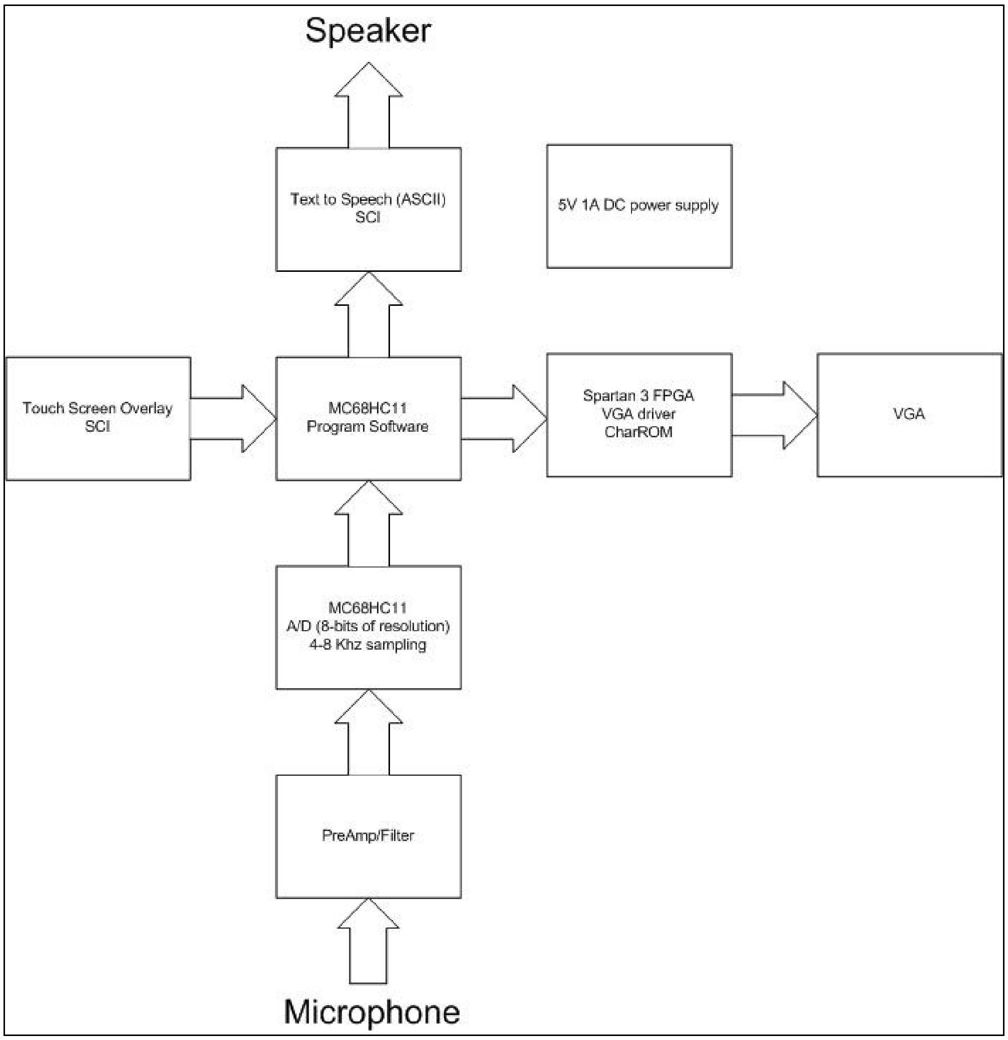


Diagram 5. Recipedia Block Diagram

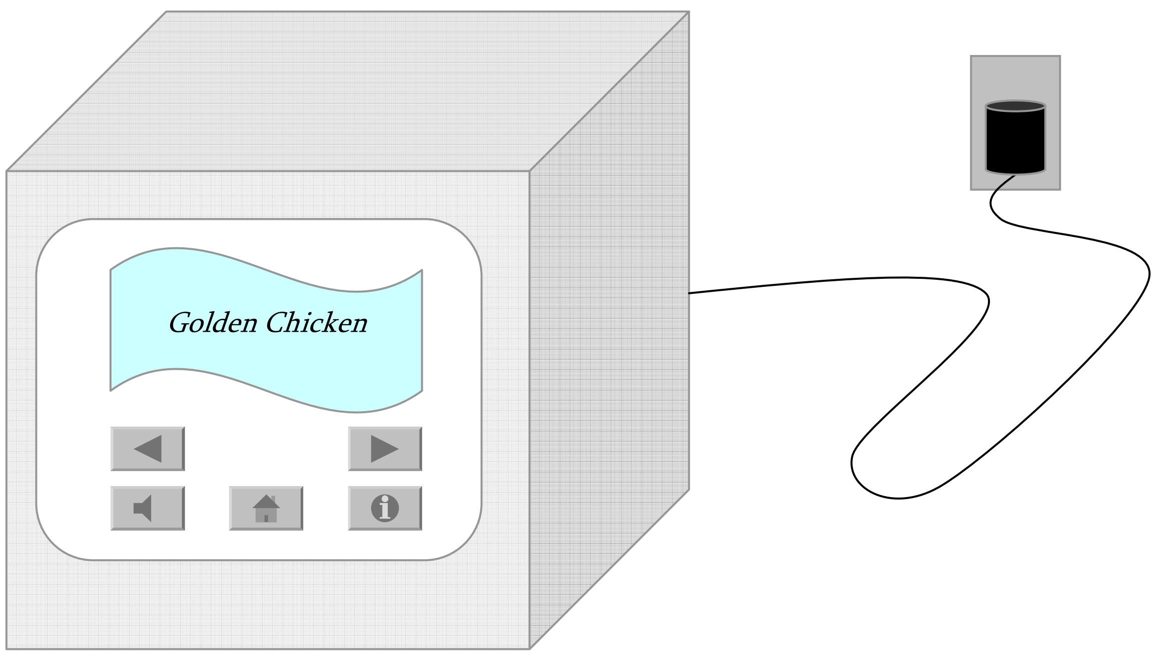


Diagram 6. System Enclosure