Project Title: BabyMac

Team Name: Microchicks

Team Members:
Name: Melissa Belleme
Email: melissabelleme@hotmail.com
Phone: 954-529-9193

Name: Katey Pickard
Email: lkp52187@ufl.edu
Phone: 256-651-9628

Project Abstract:
We will design a game for small children. This is an interactive game that will prompt the user with questions about images that display on a LCD touch screen. The user will have to identify colors and quantities. With the use of the LCD touch screen the child will have an unlimited number of opportunities to get the right answer. Once the correct image is chosen then the next question and answer key will be shown. The main components include an Atmel microprocessor, and LCD, a touch screen, a voice synthesizer chip, and a speaker.
Table of Contents:

I. Introduction ............................................................................................................. 3
II. Technical Objectives .......................................................................................... 3
III. Project Features ............................................................................................... 3
IV. Competitors .................................................................................................... 4
V. Concept Technology .......................................................................................... 4
VI. Project Architecture ......................................................................................... 8
VII. Design Procedure ........................................................................................... 12
IX. Flowchart ........................................................................................................ 13
X. Bill of Materials ................................................................................................ 15
XI. Gantt Chart ...................................................................................................... 16
XII. Division of Work ............................................................................................. 17
XIII. Appendices .................................................................................................... 17

How many Bumblebees do you see?
I. Introduction

BabyMac is an educational game but it can be modified to use the components and software for many other applications. The interactive interface can be applied to phone dialing, temperature control, and many more home automation applications.

The purpose of the project is to provide kids with a new game that is innovative and interactive. This game is more likely to get children interested in learning about their basics such as numbers and colors. For example, if we want to teach the child the ABC’s, using flashcards can bore the child very quickly and create little enthusiasm to try again. BabyMac on the other hand, will speak and ask the child to recognize the ABC’s presented in colorful Disney images and the machine will respond if it is correct or not. This interaction enhances the interest of the child towards learning.

II. Technical Objectives

The main objective of our project is to design an interactive game that has touch sensing capabilities and speech. Our technical objective is to achieve good communication between all our devices through our microprocessor as shown in Figure 14.

- The whole system should run with a maximum power supply of 12V.
- The LCD touch screen is about 6 inches diagonal.
- No additional memory should be necessary since the images are being provided by the PC.
- The game should turn itself off if it does not receive a response within two minutes.
- The processor used will be an Atmega324P of Atmel.
- The response time of the touch screen should be less than one second.

III. Project Features

- The system will have an LCD color display to show the answer options of the question asked.
- A touch screen is located on top of the LCD in order to send messages back to the microprocessor with the answer picked.
- An 8-Ohm speaker is connected to the text-to-speech to guide the user throughout the game.
- The game uses Visual Basic to organize and display the images synchronously.
IV. Competitors

There are similar products to BabyMac, but they are not directed to the same audience. In this project our audience is small children who are just starting to learn shapes, colors, etc. The games that are available in the market for this age group usually are not as sophisticated, they do not possess the same electronic components that we have included such as the touch screen, the text to speech processor, voice synthesizer, etc. Therefore this product is not comparable to any other kid’s games in the market.

Our cost objective is to keep the total price of materials to less than US$600.00

V. Concept Technology

In order for this product to become innovative, the minimum requirements are to include the following main components:

**Resistive touch screen**

A four wire resistive touch screen consists of two flexible layers coated with a transparent resistive material and separated by an air gap. The two layers are separated by invisible separator dots. When operating, an electrical current moves through the screen. When pressure is applied to the screen the layers are pressed together, causing a change in the electrical current and a touch event to be registered.

![Four-wire Resistive Touch screen composition](image)

**FIGURE 1.** Four-wire Resistive Touch screen composition

We decided to use a four-wire touch screen because they are generally the most affordable and durable so it will appropriate for our application. The pin out of the four wires is:

- Pin 1: Top
- Pin 2: Left
- Pin 3: Bottom
- Pin 4: Right
The BabyMac will feature a text-to-speech (TTS) synthesizer to allow vocal commands and interaction from the game. The TTS chip set of choice is the DoubleTalk RC8650. The RC8650 uses a serial input/output to communicate with the microprocessor and converts received ASCII text to spoken English. This chip has the option of a digital or analog output; the BabyMac will utilize the analog output through an 8-ohm speaker. The RC8650 also has internal memory and recording capabilities, as well as touchtone, musical, and sinusoidal tone generators. The RC8650 was selected because of the quality of the audio output, as well as the ease of using a serial interface. Figure 3 shows the basic hardware features of the chipset.
**Atmel Microcontroller:**
We chose to incorporate the Atmel microcontroller to interface with the voice synthesizer chip, and touch screen. This microprocessor possesses two serial ports that makes possible to communicate with the PC and the text-to-speech chip serially.

- Operating voltage: 1.8 – 5.5V
- Two programmable serial USART
- 10-bit analog-to-digital converter
- 32 programmable I/O lines

**LCD Display**
In this application a LCD Display that receives LVDS signaling was used. It is possible to use other LCDs that might take VGA signal directly. For cost effectiveness the LVDS display was chosen and a VGA to LVDS converter was required (See Figure 5). This signal converter is called ALR-1400 from DigitalView.

The LCD chosen is 6.5” diagonal and it is large enough to include all eight answer options of every question. Below, the block diagram and pin out of the two connectors that had to be built.
Visual Basic Programming

We chose to use Visual Basic to design our PC interface because it provided the simplest way to accomplish our picture viewer. We were able to control the pictures being displayed on the PC by having our VB program read a serial input through the USB cable. Visual Basic also allowed us to easily control the location of the form windows. The form displayed on the touch screen is sent through the VGA-LVDS as an extension of the PC desktop. The code used can be obtained from the appendices section.
VI. Project Architecture

Figure 7 and 8 show the schematic and layout of the Power/USB/Atmega 324P Board. This Board uses the 12V power supply and routes 12V to the VGA to LVDS converter shown in Figure 5. It also steps down the voltage to 5V used by the microprocessor, the touch screen, and the USB chip. Finally, it steps down the voltage to 3.3V with will be used to power the backlight in the LCD Display. This board has connectors that tie this board with the main power supply, the touch screen, the programmer, the USB connector from the PC, the power connector of the VGA to LVDS board, the LCD backlight “ON” signal, and finally it ties to the TTS board.
FIGURE 8. Layout of Power/USB/Atmega324P Board

FIGURE 9. Text-to-Speech Board Connections
The Power/USB/Atmega342P board contains the Atmega324P, the FT232RL USB conversion chip, as well as two voltage regulators (one 3.3V and one 5V). The Atmel has two USARTs; one is used for output to the Text-to-speech board, and the other is used for output to the PC through the USB chip. The Atmel also has four inputs on its analog port from the analog touch screen. The on-board A/D converter is used to convert the signals from the touch screen. The FT232RL converts the TX/RX signals into appropriate format for the USB connection to the PC.

The Text-to-speech board uses the RC8650 Text-to-Speech chip from RCSystems to convert ASCII text to speech. The only inputs to this board are TX, RX, power, and ground. The connections shown in the figure above were made between the two components of the RC8650. Also, the amplifier circuit in the figure shows a 3kHz lowpass filter and an LM4876 amplifier as recommended by the manufacturers. The speaker is an 8 Ω speaker that has a power rating of 2 Watts. The LM4876 is a 1.1 W Audio Amplifier.

Finally, all these main components are put together inside a plastic enclosure:

- Power/USB/Atmega board
- VGA to LVDS signal converter
- Text-to-speech board with amplifier
- 8-Ohm speaker
- LCD Display
- Touch screen
FIGURE 11. Integrating components

FIGURE 12. Finished product
**VIII. Design Procedure**

When deciding about the purpose of the project, we began by deciding the level of complexity we wanted to include on this experimental project. We started with the following block diagram:

![Initial Block Diagram](image1)

**FIGURE 13. Initial block diagram**

After faculty reviewed this design it was decided not to include the FPGA component since both of the teammates that are part of this group are inexperienced with digital design. The final block diagram for the system is shown in Figure 14.

![Final Block Diagram](image2)

**FIGURE 14. Final Block Diagram**
IX. Flowchart

The algorithm used to write the game is shown in Figure 15.

FIGURE 15. Algorithm for the game
How the game works

1. Start the game
2. A cartoon image will be displayed in the laptop
3. A voice will ask a question related to the picture
4. The touch screen LCD display will give the user eight possible answers
5. The user will use his or her finger to respond to the question
6. The voice will respond if the selection was right or wrong
7. Once the right answer was chosen the next question will be asked.

FIGURE 16. How the game works
### X. Bill of Materials

<table>
<thead>
<tr>
<th>Description</th>
<th>Part Number</th>
<th>Quantity</th>
<th>Price Each</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.4&quot; Touch screen - Bergquist</td>
<td>4004270000%</td>
<td>1</td>
<td>$58.90</td>
<td>$58.90</td>
</tr>
<tr>
<td>6.5&quot; LCD Display - Toshiba</td>
<td>LTA065B0D2F</td>
<td>1</td>
<td>$283.00</td>
<td>$283.00</td>
</tr>
<tr>
<td>ALR-1400 - DigitalView</td>
<td>41710003X-3</td>
<td>1</td>
<td>$131.67</td>
<td>$131.67</td>
</tr>
<tr>
<td>ATMEGA, ATMEG324P</td>
<td>ATMEGA324P-20PU-ND</td>
<td>1</td>
<td>$6.02</td>
<td>$6.02</td>
</tr>
<tr>
<td>CAPACITOR, 1µF</td>
<td>GRM21BR71C105MA01L</td>
<td>3</td>
<td>$0.11</td>
<td>$0.33</td>
</tr>
<tr>
<td>CAPACITOR, 10µF</td>
<td>GRM21BR72A103MA01L</td>
<td>3</td>
<td>$0.12</td>
<td>$0.36</td>
</tr>
<tr>
<td>CAPACITOR, 22pF</td>
<td>D220J20C0GH63L6R</td>
<td>3</td>
<td>$0.11</td>
<td>$0.33</td>
</tr>
<tr>
<td>CAPACITOR, 150pF</td>
<td>K151J15C0GF5TH5</td>
<td>1</td>
<td>$0.09</td>
<td>$0.09</td>
</tr>
<tr>
<td>LM4876 Amplifier</td>
<td>LM4876M/NOPB</td>
<td>1</td>
<td>$1.50</td>
<td>$1.50</td>
</tr>
<tr>
<td>Double Talk RC8650</td>
<td>RC8650</td>
<td>1</td>
<td>$38.00</td>
<td>$38.00</td>
</tr>
<tr>
<td>OSCILLATOR, 7.3728 MHz</td>
<td>ECS-8FM-073-TR</td>
<td>1</td>
<td>$1.50</td>
<td>$1.50</td>
</tr>
<tr>
<td>SPEAKER, 8Ω</td>
<td>AS05008MR-4-R</td>
<td>1</td>
<td>$2.19</td>
<td>$2.19</td>
</tr>
<tr>
<td>CAPACITOR, 8.2nF</td>
<td>UP050B822K-B-BZ</td>
<td>1</td>
<td>$0.03</td>
<td>$0.03</td>
</tr>
<tr>
<td>CAPACITOR, 22nF</td>
<td>B37987M1223K054</td>
<td>1</td>
<td>$0.09</td>
<td>$0.09</td>
</tr>
<tr>
<td>CAPACITOR, 100nF</td>
<td>GRM21AR72E102KW01D</td>
<td>1</td>
<td>$0.14</td>
<td>$0.14</td>
</tr>
<tr>
<td>CAPACITOR, 10nF</td>
<td>GRM21BR72A103MA01L</td>
<td>2</td>
<td>$0.12</td>
<td>$0.24</td>
</tr>
<tr>
<td>CONNECTORS, HDR1X2</td>
<td>1-480698-9</td>
<td>5</td>
<td>$0.38</td>
<td>$1.90</td>
</tr>
<tr>
<td>CONNECTORS, HDR1X4</td>
<td>1-1747276-4</td>
<td>2</td>
<td>$1.19</td>
<td>$2.38</td>
</tr>
<tr>
<td>DIODE, DIODE</td>
<td>1N4148W-TP</td>
<td>4</td>
<td>$0.44</td>
<td>$1.76</td>
</tr>
<tr>
<td>DR731ND, INDUCTOR_10µH</td>
<td>DR73-100-R</td>
<td>1</td>
<td>$1.67</td>
<td>$1.67</td>
</tr>
<tr>
<td>DR731ND, INDUCTOR-15uH</td>
<td>DR73-150-R</td>
<td>1</td>
<td>$1.67</td>
<td>$1.67</td>
</tr>
<tr>
<td>FT232</td>
<td>768-1007-2-ND</td>
<td>1</td>
<td>$2.65</td>
<td>$2.65</td>
</tr>
<tr>
<td>FUSE</td>
<td>MFU0805FF01500P100</td>
<td>3</td>
<td>$0.47</td>
<td>$1.41</td>
</tr>
<tr>
<td>LT1616</td>
<td>LT1616ES6#TRMPBF</td>
<td>2</td>
<td>$2.10</td>
<td>$4.20</td>
</tr>
<tr>
<td>RESISTOR, 1.5kΩ</td>
<td>CRCW08051K50JNEA</td>
<td>3</td>
<td>$0.08</td>
<td>$0.24</td>
</tr>
<tr>
<td>RESISTOR, 10kΩ</td>
<td>CRCW080510K0JNEA</td>
<td>3</td>
<td>$0.08</td>
<td>$0.24</td>
</tr>
<tr>
<td>RESISTOR, 16.5kΩ</td>
<td>CRCW080516K5FKEA</td>
<td>1</td>
<td>$0.09</td>
<td>$0.09</td>
</tr>
<tr>
<td>RESISTOR, 1kΩ</td>
<td>CRCW08051K00JNEA</td>
<td>1</td>
<td>$0.08</td>
<td>$0.08</td>
</tr>
<tr>
<td>RESISTOR, 30.1kΩ</td>
<td>CRCW080530K1FKEA</td>
<td>1</td>
<td>$0.09</td>
<td>$0.09</td>
</tr>
<tr>
<td>RESISTOR, 100kΩ</td>
<td>CFR-50JB-100K</td>
<td>5</td>
<td>$0.06</td>
<td>$0.30</td>
</tr>
<tr>
<td>RESISTOR, 47kΩ</td>
<td>CFR-50JB-47K</td>
<td>1</td>
<td>$0.06</td>
<td>$0.06</td>
</tr>
<tr>
<td>RESISTOR, 33kΩ</td>
<td>CFR-50JB-33K</td>
<td>1</td>
<td>$0.06</td>
<td>$0.06</td>
</tr>
<tr>
<td>Other (header, crimp, solder)</td>
<td></td>
<td></td>
<td></td>
<td>$20.00</td>
</tr>
</tbody>
</table>

**TOTAL** $563.19
XI. Gantt Chart

Available Weeks in Spring 2009

(K) Preliminary Project Proposal
(M+K) Research/Formal Proposal
(K) TTS Development Board Testing
(K) TTS Breadboard & Code Design
(K) PC Interface Research
(M) TScreen w/ Development Board
(K) TScreen w/ Hardware &...
(M) TScreen w/ Breadboard &...
(K) Kately @ Interview (Off)
(M+K) Integration of All Parts
(M) PCB Design-TScreen-M
(K) PCB Design-TTS & PC Interface
(M+K) Populate PCB
(M+K) Testing & Troubleshooting
(M+K) Spring Break
(M+K) Final Presentation (April 14)
(M+K) Final Demo (April 16)

Planned □ Extension □ Downtime
XII. Division of Work

<table>
<thead>
<tr>
<th>Task Name</th>
<th>Melissa Belleme</th>
<th>Katey Pickard</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATMEL TTS Code Design</td>
<td>0%</td>
<td>100%</td>
</tr>
<tr>
<td>TTS PCB Design/Construction</td>
<td>0%</td>
<td>100%</td>
</tr>
<tr>
<td>Visual Basic Code Design</td>
<td>0%</td>
<td>100%</td>
</tr>
<tr>
<td>Poster</td>
<td>0%</td>
<td>100%</td>
</tr>
<tr>
<td>Touch Screen Code Design</td>
<td>100%</td>
<td>0%</td>
</tr>
<tr>
<td>Enclosure and packaging</td>
<td>100%</td>
<td>0%</td>
</tr>
<tr>
<td>Power Regulator PCB Design/Construction</td>
<td>100%</td>
<td>0%</td>
</tr>
<tr>
<td>ATMEL/FTDI PCB Design/Construction</td>
<td>90%</td>
<td>10%</td>
</tr>
<tr>
<td>Final Report</td>
<td>90%</td>
<td>10%</td>
</tr>
</tbody>
</table>

XIII. Appendices

Atmega324P Code:

// Project Title: BabyMac
// Authors: Katey Pickard & Melissa Belleme
// Project description: BabyMac is an interactive children's game that verbally questions the child about colors, numbers, and shapes seen in images displayed on a PC monitor. The child interacts with the game through a touch screen.

Code Description: This program is written for the Atmega324P microprocessor. The code sets up both UARTs for use; One UART will control the Text-to-Speech component and the other will be giving input to the PC to decide which image and answer key to display in the VB application. The program is set up using structures so that many images, image numbers, corresponding questions and answers can be added and accessed easily for serial communication with the TTS and PC. The program also receives input from the touch screen, and decides which of the 8 squares the user has touched.

#include <avr/io.h>
#include <util/delay.h>
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include "AVR32_USART.h"
#include "LCD.h"
#include <avr/pgmspace.h>

//****functions*****
unsigned int ReadTScreenX (void);
unsigned int ReadTScreenY (void);
void delay(int del);

//****global variables*****
//Note: Global values can be displayed on the watch window (debug mode) unsigned int
unsigned int MSB=0,LSB=0,XValue=0,YValue=0;
unsigned int XAxis=0,YAxis=0;
unsigned int xValue=0,yValue=0;
char x;
int count=0;
int set=0;  // Set chooses which of the images will be used
int verify=0;  // 0 = false and 1 = true
int answer=0;
int picture=0;

struct image { // Set up structure to keep track of data for each image/screen
    int image_num; // Image number: to send to PC to get correct picture
    int correct; // Tells which of the eight squares contains the correct answer
    char *text;   // Greeting Message
    char *question; // First Question
    char *ans1;   // Answer Choice #1
    char *ans2;
    char *ans3;
    char *ans4;
    char *ans5;
    char *ans6;
    char *ans7;   // Answer Choice #8
    char *ans8;
    char *oops;   // Try again message
    char *good;  // Good job message
};

const struct image data[] =
{
    {1,2,"Welcome to Baby Mac! Let's play a game. Are you ready to have fun?","Here is your first question: What color is Belle's dress?","Green","Pink","Blue","Red","Orange","Black","Brown","Yellow","No, that's wrong. Why don't you try it again?","Perfect! You are off to a great start."},

    {2,3,"Welcome to Baby Mac! Let's play a game. Are you ready to have fun?","Now let's look at this picture again. How many blue birds do you see?","One","Two","Three","Four","Five","Six","Seven","Eight","That's not quite right. Let's think again.","Wonderful! You're doing great!"},

    {3,6,"Welcome to Baby Mac! Let's play a game.","Now it's time to have some fun. How many puppies can you count?","One","Two","Three","Four","Five","Six","Seven","Eight","That's incorrect. Why don't you try again?","Good job! That's right! Let's keep going."},

    {4,2,"Welcome to Baby Mac! Let's play a game. Are you ready to have fun?","Now look a little closer. How many puppies are wearing hats?","One","Two","Three","Four","Five","Six","Seven","Eight","No. Count again.","You're right."},

    {5,4,"Welcome to Baby Mac! Let's play a game.","What color are the puppies hats?","Green","Pink","Blue","Red","Orange","Black","Brown","Yellow","Why don't you try again?","Fantastic."},

    {6,8,"Welcome to Baby Mac! Let's play a game. Are you ready to have fun?","What color is Ariel's friend Flounder?","Green","Pink","Blue","Red","Orange","Black","Brown","Yellow","Maybe you looked at the wrong fish. Try again.","That's outstanding! You sure know your colors."},

    {7,4,"Welcome to Baby Mac! Let's play a game. Are you ready to have fun?","Now, what color is Ariel's hair?","Green","Pink","Blue","Red","Orange","Black","Brown","Yellow","I'll give you a hint: Ariel is leaning on a rock.","That's right! Good job!"},

    {8,3,"Welcome to Baby Mac! Let's play a game. Are you ready to have fun?"},
};
"Now it's time to get ready for the ball. What color is Cinderella's dress?", "Green", "Pink", "Blue", "Red", "Orange", "Black", "Brown", "Yellow", "Maybe you should try again.", "Excellent. You're doing so well.").

//Image 9: Cinderella--How many pumpkins?
{9,3,"Welcome to Baby Mac! Let's play a game. Are you ready to have fun?", "Now look around. How many pumpkins are there?", "One", "Two", "Three", "Four", "Five", "Six", "Seven", "Eight", "Oops, think again.", "That's just terrific.").

//Image 10: Cinderella--What color are the pumpkins?
{10,5,"Welcome to Baby Mac! Let's play a game. Are you ready to have fun?", "Let's look again. What color are the pumpkins?", "Green", "Pink", "Blue", "Red", "Orange", "Black", "Brown", "Yellow", "Pumpkins aren't that color! Try again.", "You're doing just fine. Thanks for helping Cinderella get ready for the ball.").

//Image 11: Cinderella--How many ugly stepsisters does Cinderella have?
{11,2,"Welcome to Baby Mac! Let's play a game. Are you ready to have fun?", "Cinderella's stepsisters don't want her to try on the slipper. How many ugly stepsisters does she have?", "One", "Two", "Three", "Four", "Five", "Six", "Seven", "Eight", "No. That's wrong. Try again.", "Good job!").

//Image 12: Cinderella--How many mice friends does Cinderella have?
{12,4,"Welcome to Baby Mac! Let's play a game. Are you ready to have fun?", "Thank goodness Cinderella has some mice friends there to help her. How many mice can you count?", "One", "Two", "Three", "Four. You're right! The slipper fits!", "Five", "Six", "Seven", "Eight", "No. That's wrong. Try again.", "Now Cinderella can live happily ever after. That's all for today. Thanks for playing!").

};

void SendImageNumber(int a)
{// Send Image Number to PC through USART0
 x = data[a].image_num;
 while ( !( UCSR0A & (1<<UDRE0)) ); // Get next character from text string
 UDR0 = x;
 return;
}

void AskQuestion(int a)
{// Send text to TTS Board through USART1
 for (count=0; data[a].question[count] != '\0'; count++)
 { x = data[a].question[count];
 while ( !( UCSR1A & (1<<UDRE1)) );
 UDR1 = x;
 }
 while ( !( UCSR1A & (1<<UDRE1)) );
 UDR1=0x0D;
 return;
}

void ReadAnswer(int a, int b)
{
 if (b==data[a].correct)
 verify=1;
 else verify=0;
 if (b==0)
 return;
 if (b==1) {
 for (count=0; data[a].ans1[count] != '\0'; count++)
 { x = data[a].ans1[count];
 while ( !( UCSR1A & (1<<UDRE1)) );
 UDR1 = x;
 }
 while ( !( UCSR1A & (1<<UDRE1)) );
 }
UDR1=0x0D;
return;
}
else if (b==2) {
    for (count=0; data[a].ans2[count] != '0'; count++)
    {
        x = data[a].ans2[count];
        while ( !( UCSR1A & (1<<UDRE1)) ) ;
        UDR1 = x;
    }
    while ( !( UCSR1A & (1<<UDRE1)) ) ;
    UDR1=0x0D;
    return;
}
else if (b==3) {
    for (count=0; data[a].ans3[count] != '0'; count++)
    {
        x = data[a].ans3[count];
        while ( !( UCSR1A & (1<<UDRE1)) ) ;
        UDR1 = x;
    }
    while ( !( UCSR1A & (1<<UDRE1)) ) ;
    UDR1=0x0D;
    return;
}
else if (b==4) {
    for (count=0; data[a].ans4[count] != '0'; count++)
    {
        x = data[a].ans4[count];
        while ( !( UCSR1A & (1<<UDRE1)) ) ;
        UDR1 = x;
    }
    while ( !( UCSR1A & (1<<UDRE1)) ) ;
    UDR1=0x0D;
    return;
}
else if (b==5) {
    for (count=0; data[a].ans5[count] != '0'; count++)
    {
        x = data[a].ans5[count];
        while ( !( UCSR1A & (1<<UDRE1)) ) ;
        UDR1 = x;
    }
    while ( !( UCSR1A & (1<<UDRE1)) ) ;
    UDR1=0x0D;
    return;
}
else if (b==6) {
    for (count=0; data[a].ans6[count] != '0'; count++)
    {
        x = data[a].ans6[count];
        while ( !( UCSR1A & (1<<UDRE1)) ) ;
        UDR1 = x;
    }
    while ( !( UCSR1A & (1<<UDRE1)) ) ;
    UDR1=0x0D;
    return;
```c
}  
else if (b==7) {
    for (count=0; data[a].ans7[count] != '0'; count++) {
        x = data[a].ans7[count];
        while (! (UCSR1A & (1<<UDRE1))) ;
        UDR1 = x;
        while ( ! (UCSR1A & (1<<UDRE1))) ;
        UDR1=0x0D;
        return;
    }
}  
else if (b==8) {
    for (count=0; data[a].ans8[count] != '0'; count++) {
        x = data[a].ans8[count];
        while (! (UCSR1A & (1<<UDRE1))) ;
        while ( ! (UCSR1A & (1<<UDRE1))) ;
        UDR1=0x0D;
        Return;
    }
}  
else return;

void SendText(int a){ // Send text to TTS Board through USART1
    for (count=0; data[a].text[count] != '0'; count++) {
        x = data[a].text[count];
        while (! (UCSR1A & (1<<UDRE1))) ; // Wait for empty transmit buffer
        UDR1 = x;
        while ( ! (UCSR1A & (1<<UDRE1))) ;
        UDR1=0x0D;
        return;
    }
}

void TryAgain(int a){
    for (count=0; data[a].oops[count] != '0'; count++) //Loop through struct array until all characters in string have been sent
    {
        x = data[a].oops[count];
        // Get next character from text string
        while ( !(UCSR1A & (1<<UDRE1))) ; // Wait for empty transmit buffer
        UDR1 = x;  // Send character
        while ( !(UCSR1A & (1<<UDRE1))) ; // Wait for empty transmit buffer
        UDR1=0x0D;     // Send carriage return to signify end of line
        return;
    }
}

void GoodJob(int a){
    for (count=0; data[a].good[count] != '0'; count++) //Loop through struct array until all characters in string have been sent
    {
        x = data[a].good[count];
        // Get next character from text string
        while ( !(UCSR1A & (1<<UDRE1))) ;  // Wait for empty transmit buffer
        UDR1 = x;
        while ( !(UCSR1A & (1<<UDRE1))) ; // Wait for empty transmit buffer
        UDR1=0x0D;
        return;
    }
}  
```
x = data[a].good[count];
   // Get next character from text string
   while ( !( UCSR1A & (1<<UDRE1)) )                   // Wait for empty transmit buffer
     UDR1 = x;  // Send character
   while ( !( UCSR1A & (1<<UDRE1)) )                   // Wait for empty transmit buffer
     UDR1 = 0x0D; // Send carriage return to signify end of line
   return;

unsigned int ReadTScreenY(void)
{
  DDRA=0b00000101;  //bit0:X+ bit1:Y- bit2:X- bit3:Y+
  PORTA|=0b00000001;  //X+ 5V
  PORTA&=~0b00000100;  //X- 0V
  DIDR0&=~0b00001010;  //analog input for Y- and Y+
  ADMUX=0b01000001;   // channel 1 (ADC1)
  ADCSRA = 0b10000110; // ADC clock prescalar of 64
  ADCSRA |= (1 << ADSC);  // Start conversion
  while(ADCSRA & (1 << ADSC)); // wait for conversion to complete
  return ADC;//XValue;
}

unsigned int ReadTScreenX(void)
{
  DDRA=0b00001010;  //bit0:X+ bit1:Y- bit2:X- bit3:Y+
  PORTA|=0b00001000;  //Y+ 5V
  PORTA&=~0b00000010;  //Y- 0V
  DIDR0&=~0b00000101;  //analog input for Y- and Y+
  ADMUX=0b01000000;   // channel 0 (ADC0)
  ADCSRA = 0b10000110; // ADC clock prescalar of 64
  ADCSRA |= (1 << ADSC);  // Start conversion
  while(ADCSRA & (1 << ADSC)); // wait for conversion to complete
  return ADC;
}

int GetResponse(void)
{
  xValue=ReadTScreenX();
  _delay_ms(10);
  yValue=ReadTScreenY();
  _delay_ms(10);        // Check input
  if ((xValue < 120) & (yValue < 120)){
    PORTB = 0x00;
    return 0;}
  else if (((xValue > 0) & (xValue < 256)) & ((yValue > 513) & (yValue < 1024))){
    PORTB = 0x01;
    return 1;}
  else if (((xValue > 257) & (xValue < 512)) & ((yValue > 513) & (yValue < 1024))){
    PORTB = 0x02;
    return 2;}
  else if (((xValue > 513) & (xValue < 768)) & ((yValue > 513) & (yValue < 1024))){
    PORTB = 0x03;
return 3;
}
else if (((xValue > 769) & (xValue < 1024)) & ((yValue > 513) & (yValue < 1024)))
{  
  PORTB = 0X04;
  return 4;
}
else if (((xValue > 100) & (xValue < 256)) & ((yValue > 100) & (yValue < 512)))
{  
  PORTB = 0X05;
  return 5;
}
else if (((xValue > 257) & (xValue < 512)) & ((yValue > 0) & (yValue < 512)))
{  
  PORTB = 0X06;
  return 6;
}
else if (((xValue > 513) & (xValue < 768)) & ((yValue > 0) & (yValue < 512)))
{  
  PORTB = 0X07;
  return 7;
}
else if((xValue > 769) & (xValue < 1024)) & ((yValue > 0) & (yValue < 512))
{  
  PORTB = 0X08;
  return 8;
}
else return 0;

void delay_3s(void)
{
  _delay_ms(500);
  _delay_ms(500);
  _delay_ms(500);
  _delay_ms(500);
  _delay_ms(500);
  _delay_ms(500);
}

void WaitPC(void)
{
  while ( !(UCSR0A & (1<<RXC0)) )  // Wait for data to be received  
  {
    x=UDR0;
    return;
  }
}

int main(void)
{
  LCD_INIT();  // Initialize LCD for debugging
  USART_INIT(9600); //Sets up both UARTs for 9600 baud, no parity, 1 stop bit
  //*****PORT USED FOR LEDS*****
  DDRB=0x0F; //bit 0:1 output
  PORTB=0x00; //All off
  DDRD=0xFF;
  PORTD=0;
  while (1)
  {
    set=0;
    //delay_3s();
    WaitPC();
    SendText(set); // Welcome the User to the game
    _delay_ms(500);
    _delay_ms(500);
    for (set = 0; set <= 7; set++)
    {
      SendImageNumber(set); //picture= set + 1;
      delay_3s();
      AskQuestion(set); // Ask the question
      delay_3s();
      while (verify == 0){
        // Wait for input from user
      }
    }
  }
}
GetResponse();
answer=GetResponse(); // Read in response from touch screen
ReadAnswer(set,answer); // TTS reads response aloud and "verify" is set
if((answer != 0) & (verify == 0)){
TryAgain(set);
delay_3s();
answer=0;
_delay_ms(250);
}
// SendImageNumber(picture);
_delay_ms(500);
GoodJob(set);
verify=0;
}
while( !( UCSR0A & (1<<UDRE0)) ) // Wait for empty transmit buffer
{
UDR0 = 1; // Send character
return 1;
}

**Visual Basic Code:**

```vb
Public Class MDIParentForm
  Public portin As Integer
  Public var1 As String
  Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    Image0Form.MdiParent = Me
    ImageFormKey.MdiParent = Me
    Image0Form.Show()
    ImageFormKey.Show()
    var1 = "G"
    SerialPort1.Open() 'Open Serial port on COM12, 9600 baud, 8 data, no parity, 1 stop bit
    SerialPort1.Write(var1)
  End Sub
  Private Sub SerialPort1_DataReceived(ByVal sender As System.Object, ByVal e As System.IO.Ports.SerialDataReceivedEventArgs) Handles SerialPort1.DataReceived
    Try
      ImageFormKey.Close()
      Image0Form.Close()
    Catch ex As Exception
      MsgBox("closing error")
    End Try
    portin = SerialPort1.ReadByte
    If portin = 1 Then
      Image0Form.number = 15
      ImageFormKey.number = 15
      ImageFormKey.Show()
      Image0Form.Show()
      MsgBox("You win! Game Over. Please Close BabyMac now.")
      Image0Form.Close()
      ImageFormKey.Close()
    Else
      Image0Form.number = portin
      ImageFormKey.number = portin
    End If
  End Sub
```

getresponse();

answer = getresponse(); // read in response from touch screen

readanswer(set, answer); // TTS reads response aloud and "verify" is set

if (answer != 0 & verify == 0)
{
    tryagain(set);
delay_3s();
    answer = 0;
    delay_ms(250);
}

// sendlevelname(picture);

delay_ms(500);
goodjob(set);
verify = 0;
}

while (!(UCSR0A & (1<<UDRE0))) // wait for empty transmit buffer
{
    UDR0 = 1; // send character
    return 1;
}

**Visual Basic Code:**

```vb
Public Class MDIParentForm
  Public portin As Integer
  Public var1 As String
  Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    Image0Form.MdiParent = Me
    ImageFormKey.MdiParent = Me
    Image0Form.Show()
    ImageFormKey.Show()
    var1 = "G"
    SerialPort1.Open() 'Open Serial port on COM12, 9600 baud, 8 data, no parity, 1 stop bit
    SerialPort1.Write(var1)
  End Sub
  Private Sub SerialPort1_DataReceived(ByVal sender As System.Object, ByVal e As System.IO.Ports.SerialDataReceivedEventArgs) Handles SerialPort1.DataReceived
    Try
      ImageFormKey.Close()
      Image0Form.Close()
    Catch ex As Exception
      MsgBox("closing error")
    End Try
    portin = SerialPort1.ReadByte
    If portin = 1 Then
      Image0Form.number = 15
      ImageFormKey.number = 15
      ImageFormKey.Show()
      Image0Form.Show()
      MsgBox("You win! Game Over. Please Close BabyMac now.")
      Image0Form.Close()
      ImageFormKey.Close()
    Else
      Image0Form.number = portin
      ImageFormKey.number = portin
    End If
  End Sub
```
ImageFormKey.Show()
Image0Form.Show()
MsgBox("Click OK")
End If
End Sub
End Class

Public Class Image0Form
  Public number As Integer
  Dim portin As Integer
  Private Sub Image0Form_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    If number = 1 Then
      picShowPicture.Image = My.Resources.belle
    ElseIf number = 2 Then
      picShowPicture.Image = My.Resources.belle
    ElseIf number = 3 Then
      picShowPicture.Image = My.Resources.dalmatians
    ElseIf number = 4 Then
      picShowPicture.Image = My.Resources.dalmatians
    ElseIf number = 5 Then
      picShowPicture.Image = My.Resources.dalmatians
    ElseIf number = 6 Then
      picShowPicture.Image = My.Resources.ariel
    ElseIf number = 7 Then
      picShowPicture.Image = My.Resources.ariel
    ElseIf number = 8 Then
      picShowPicture.Image = My.Resources.cinderella2
    ElseIf number = 9 Then
      picShowPicture.Image = My.Resources.cinderella2
    ElseIf number = 10 Then
      picShowPicture.Image = My.Resources.cinderella2
    ElseIf number = 11 Then
      picShowPicture.Image = My.Resources.cinderella3
    ElseIf number = 12 Then
      picShowPicture.Image = My.Resources.cinderella3
    Else
      picShowPicture.Image = My.Resources.main_default
    End If
    'MsgBox(portin)
  End Sub
End Class

Public Class ImageFormKey
  Public number As Integer
  Dim portin As Integer
  Private Sub ImageFormKey_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    If number = 1 Then
      AnswerKey.Image = My.Resources.colors
    ElseIf number = 2 Then
      AnswerKey.Image = My.Resources.numbers
    ElseIf number = 3 Then
      AnswerKey.Image = My.Resources.numbers
    ElseIf number = 4 Then
      AnswerKey.Image = My.Resources.numbers
    ElseIf number = 5 Then
      AnswerKey.Image = My.Resources.colors
    ElseIf number = 6 Then
      AnswerKey.Image = My.Resources.colors
    End If
  End Sub
End Class
AnswerKey.Image = My.Resources.colors
ElseIf number = 7 Then
    AnswerKey.Image = My.Resources.colors
ElseIf number = 8 Then
    AnswerKey.Image = My.Resources.colors
ElseIf number = 9 Then
    AnswerKey.Image = My.Resources.numbers
ElseIf number = 10 Then
    AnswerKey.Image = My.Resources.colors
ElseIf number = 11 Then
    AnswerKey.Image = My.Resources.numbers
ElseIf number = 12 Then
    AnswerKey.Image = My.Resources.numbers
Else
    AnswerKey.Image = My.Resources.default_ts
End If
'MsgBox(portin)
End Sub
End Class