EC450 Microprocessors: Final Project

Professor Giles

Kevin Mannix & Adam Duda

12/11/2013

Project Goals

The idea we had when we began our project was to make text based adventure game that focused mainly on the battle sequence of the adventure. With this idea we devised a state machine that would start the player off in one of three different places and, eventually, all lead to the same end destination, back in the adventurers’ home town. We wanted to give the user a way to select which way they wanted to progress through the story by using a potentiometer. Along the way the player would encounter obstacles that would require external inputs to progress in the adventure. In the beginning we wanted to have a few different obstacles such as needed to use a light source to light a cave or having to answer on screen questions to light up LEDs on a seven segment display to advance to the next state. Another hardware aspect we wanted to implement was having a motor or speaker output either a rumble or sound, respectively during state changes to simulate movement or to signify that the adventurer had taken damage. The adventurer would also encounter at least one random battle on their way home as well. The battle sequence had originally included four different options to choose between, attack, defend, power-up, and heal. The monster also originally had multiple choices that could happen in battle that exactly like the adventurer minus the healing option. Our last idea was to display all of this on a LCD screen attached to the MSP430. Before dealing with a lot of the UART code, we thought this was a very doable project.

When we began working on the project by first wanting to get the states story progression up and running. Here was the biggest snag in our plan. We worked for close to twelve hours to get UART to print multiple lines of text and then transition to the next state. Only after talking to professor Giles did we realize the way we had originally programmed the project would not work. With some alterations we were able to get the states down but this set us back quite a bit of time. Realizing we did not have enough time to bring this project to where we originally wanted to, we scrapped a lot of the ideas we had at the beginning and went for just a text based adventure game with external inputs to progress. We were able to get the potentiometer to select the direction the user wanted to go as well as a phototransistor to tell if user was supplying enough light to illuminate a dark part of the story to progress. The motor was implemented but only worked through debugging. The seven segment display was scrapped entirely, as well as the speaker, LCD screen, and the battle sequence leaving our project relying heavily on the software side of things. After some thought and additional ideas, we went on to add more hardware to the project.

A major change we made after our first demo was replacing the potentiometer with two photoresistors that the user could cover with a finger to select which way they wanted to go. We removed the motor from the project completely in favor of a speaker that plays a short 8-bit type song for background music. A random chance at entering a battle with an animal at each stage was also added with LEDs to signal how much heal you have left, 5 in total signifying 20 health each for a total of 100 health, and a another LED to blink twice when you attack the animal. We also used the TimerA module and a speaker to generate some background music. By initializing the WDT, we decreased a value for how long the note should play. By advancing through an array of frequency values that we would set as the period for each note, we generated a simple pattern of notes to make background music to play while the game was being played.

Description

Our project uses the MSP430G2553 Microcontroller connected to a breadboard that uses a phototransistor and photoresistors to be used to detect the presence of intense light and aid the user in selecting which way, left or right, they want to pick respectively. The breadboard also has a speaker connected to it as well as 6 LEDs to signify health and produce sound. The overall program utilizes the UART capabilities of the MSP430 displaying the entire game on screen. In the program we start off by printing one of three random starting points and prompt the user to either choose left or right. A speaker was used with the Watchdog Timer and the Timer A to generate some background music for the user. This provided a nice ambient sound and another dimension to the game. The way the user chooses an option is they first click the button on board of the MSP430 to get inside the button handler were is loops until a selection is made. This is done to stop the program from reading a single choice as multiple and skipping states. Once a selection is made, the program breaks out of the button handler and prints the next state that the user selected by their choice. With each state there is a 50% chance you will encounter a battle with a wild animal that you must kill to progress. Once in the battle sequence you can either attack or defend. Attacking the animal reduces its health while defending reduces the amount of damage you take from the animals attack. When you attack, a green LED will blink to signal you have dealt damage to the animal. Alternatively, every time you take increments of 20 damage, one of the LEDs representing your health will turn off. If you die against the animal you will be prompted to press the button if the user wishes to play again, but if you kill the animal you can then choose which direction you would like to continue. In one of the states you will enter a cave that is too dark to continue. The user will then need to use a light on the phototransistor to illuminate the cave to continue. Eventually after passing through the correct path the user will come to the end of the game.

Project Assessment

At first, the project had a small error that set us back quite a bit. Because we were unable to get anything to print after the initial state we could not continue to add more and more components we wanted to add. Eventually we fixed the error but by that time, the demo date was less than 24 hours away and implementing everything we wanted to seemed like an impossible task. We were able to get everything together for the first demo but we both felt that that was not at all what we wanted out of this project and decided to put more work into it. Over the next day we spent a lot of time in the lab and added a lot of things to get the project where we wanted it to be. After our final demo we both felt great about the project and would say it met our expectations.

Next Steps

After working on the project more and getting it to where it is now, there is not much we could do to improve the project. One thing we could do is make sure the phototransistor is able to calibrate itself instead of needed to be adjusted from day to day. It would be possible to replace the two photoresistors with a touch screen but that would be a monumental task which would take quite a while to learn to use let alone implement in the project. We could change to a different song when in a battle sequence to make the battle feel more intense as if your character is actually in danger. We could also increase the number of states the user could go through thus increasing the length of the game but this would possibly require us to use external memory. We would also aim to add an LCD screen rather than just using LEDs to indicate the health of both the monster you are fighting and the character. We would also like to add more complex background music to give our game some memorable music (like Mario or Zelda games, in which their music is iconic).

Final Comments

Between the two of us the work was split pretty evenly. For the most part, if one of us was in the lab working on the project, the other was right there with him. Kevin was the driving force behind a lot of the base work, ie. Text and state map. Both of use came up with ideas how to implement hardware with the project such as using a speaker for music, LEDs to signify life, and phototransistor to detect light. The idea to use photoresistors to select which way the user wanted to go was given to use from Prof. Giles and this later replaced the potentiometer idea we originally had.