Digital Notebook

Chirag Sangani, Sumeet Kumar and Nitin Pal
Under the guidance of Palash Soni
Electronics Club, IIT Kanpur

Abstract: This project's aim is to develop a device capable of basic text editor functions. Such a device would be the forerunner of a Personal Digital Assistant, which are of great value in the corporate world.

Introduction

The main objective was to create a user-friendly and intuitive interface where a user can easily type in a text note and store his data. The final product, however, had capabilities vastly superior to this original objective and implemented its own "Operating System" (OS) which allowed for different software applications to be installed into the device without affecting the proper working of the other applications already present. The OS also implemented an Application Programming Interface (API) which allowed for the development of applications with worrying about hardware considerations.

Project Design

We executed the project with a two-pronged approach – to develop the hardware and software separately and integrate them together.

A) Hardware

Physically, the device had a structure not unlike a commercial laptop. Hardware user interfaces consisted of a monochrome LCD screen capable of displaying text as well as graphics (128*64 pixels based on KS0108 controller) and a full sized QWERTY keypad using a 'Combinatorial Keypad' concept for text input. Other hardware features include a connector for supplying power via a battery pack or wall outlet and a proprietary port for interfacing with modules made specifically for this device. This proprietary port uses the standard USART interface for communication – the baud rate of which can be configured as per the requirements of the application. Due to the standardised interface and the unique nature of the device software, this port allows the device to communicate with a wide variety of other electronic objects, such as other microcontrollers, computers (via a separate RS232 serial port module which can be attached to the main device), robots, etc. The device can issue commands and receive inputs through this port.

The central unit was an ATmega32 microcontroller from Atmel's AVR series clocked at 12 MHz via an external crystal oscillator. External storage was provided in the form of an Atmel EEPROM chip (AT24C64) interfaced via Two-Wire-Interface. This additional storage device was included due to the insufficient amount of storage capacity built into the microcontroller.

B) Software

The software for this project was developed on Atmel's AVR Studio and used the AVR-GCC compiler. The development of the software was further divided into two stages – the OS and the applications.

i) The Operating System

The operating system works on the basis of pre-processor directives. Since the entire software has to be compiled as a whole and run on the device, the code looks for installed applications at compile time using these directives and generates the appropriate final code.

The OS maintains the application list. This list contains all information about the installed applications and is generated automatically at compile-time. At the main screen, the user is presented with this list and is given an option to select any one of the applications to use. After the application ends, the user is returned to this screen.

The OS maintains the Real Time System Clock. This clock maintains the current time and date, and also updates itself as long as the device is powered. To address power consumption concerns when the device was operated on a battery pack, the OS allows for a sleep mode – a state in which all the data would be retained, however, the code execution will pause and the LCD screen will turn off. One can return to normal functioning from this mode by pressing any key on the keypad.

The OS maintains the hardware drivers and automatically includes them at compile-time and loads them at boot-time. The drivers for the graphical LCD and the EEPROM chip were obtained from a free open source GPL library called AVRLIB. The drivers for the keypad were written ourselves, since the keypad uses a unique implementation.

ii) Applications

Applications developed for this project are unique in the fact that they are designed with very little hardware considerations in mind, since the OS handles most of the hardware interfacing and provides an API. The applications have to be designed according to a specific template, however, the overhead in making these applications compatible with the OS is minimal and the code structure itself remains unchanged.

These applications are made available in the initial build of the device:

a) A Digital Notepad, capable of text-editing functions and storing text notes
b) A Terminal, capable for sending and receiving data through the proprietary port
c) An Attendance Register, capable of maintain an attendance sheet.

Conclusion

The project can be said to be a success, since the completed device vastly exceeded its vision at the start of the project. Further work on the device can focus on improving the GUI implementation of the home screen and adding more features, such as calendar, address book, etc, and adding more hardware features such as the ability to play audio files and interfacing with the computer via a USB interface.

Acknowledgement

We would like to thank our mentor, Palash Soni, for his guidance and the Electronics Club Coordinators, Ankit Gupta, Anubhav Singla and Siddharth Garg, for their valuable inputs and guidance which they provided throughout this project. We would also like to thank Mr. Rajeev Shankar Sinha, who provided us with the concept of 'Combinatorial Keypad' which allowed us to implement a full QWERTY keypad.

Notes and References

The AVRLIB library used for the graphical LCD screen and the EEPROM chip can be found at: http://www.mil.ufl.edu/~chrisarnold/components/microcontrollerBoard/AVR/avrlib/