Third Prize

Smart Bus Station Sign

Institution: **Oriental Institute of Technology**

Participants: **Jian Jinrong, Zhan Yilin, Lin Taida**

Instructor: **Xiao Ruxuan**

Design Introduction

In modern society, buses are a popular, necessary public vehicle. However, there are some problems and potential dangers with these vehicles. For example, if a person waves at the driver as the bus is approaching, the driver may be distracted and cause a dangerous situation. Or, if the bus driver stops the bus in a hurry to pick up a passenger, the vehicle following the bus may hit the bus. Alternatively, if a person does not pay attention while waiting for the bus, he/she could miss the bus and waste time. Research is needed to solve these issues. Our project proposes a smart bus station sign system that can be used on all roads, improving the convenience and safety of bus passengers.

Target Users

The smart bus station sign's main interface has a variety of buttons. When a user activates the bus calling function, the main interface presents the bus lines in real time, and a module on the bus receives a message from the host machine telling the bus that it has been called. Additionally, a voice function helps the elderly use the system.

Reasons to Use the Nios II Processor in the Design

We used the Nios® II processor for several reasons:

- The Nios II system core is flexible and convenient.
- With the Terasic Development and Education (DE1) board and the Altera® Quartus® II software, we could rapidly develop an embedded prototype system with the Nios II processor.

- Our instructor and seniors at our school have performed considerable research with the Nios II processor, and our school supports this system more than any other.

These advantages allowed us to create the system quickly.

Function Description

This design uses the DE1 board. The design is presented in a digital, modern method, and includes wave-free and bus-coming display functions. With these features, the new bus station sign lightens the burdens of passengers and the driver, leading to safer transportation. Highlights of the design are:

- Smart bus station sign (the module at the bus station)
  - Build a Nios II embedded system using SOPC Builder
  - Create a SD178A voice circuit and an nRF2401 RF module on the DE1 board
  - Design a SD178A controller using VHDL
  - Compile SD178A sub-drivers using the C language
  - Compile sub-drivers using the C language
  - Compile nRF2401 RF module sub-drivers using the C language
  - Compile the program for the main interfaces of the digital bus station sign using the C language
- Smart bus station sign (module on the bus)
  - Build a Nios embedded system using SOPC Builder
  - Create an nRF2401 RF module on the Taurus ACEX1K development board
  - Compile nRF2401 RF module sub-drivers using the C language
  - Compile the main program control settings of the bus sub-module using the C language
- Functions/objectives achieved
  - Build Nios and Nios II embedded systems
  - Display the main interface of smart bus station sign using a 320 x 240 LCD
  - Locate the bus correctly by touching the touch screen
  - Accurately send to the specified bus sub-module and support wave-free and bus-coming display functions
  - Support normal Chinese voice functions
  - Transmit the RF signal without obstacles

Performance Parameters

Table 1 shows the performance parameters for our design.

Table 1. System Performance

Parameter	Description
Flow status	Successful – Wed Sep 26 10:04:46 2007
Quartus II version	6.0 Build 178 04/27/2006 SJ Full Version
Revision name	Minimal_32_sram
Top-level entity name	Minimal_32_sram
Family	Cyclone® II FPGA Family
Device	EP2C20F484C7
Timing models	Final
Met timing requirements	No
Total logic elements	3,899/18,752 (21%)
Total pins	164/315 (52%)
Total memory bits	45,952/239,616 (70%)
Total phase-locked loops (PLLs)	0/4 (25%)

Design Architecture

The following sections describe the architecture of our design.

System Structure

The hardware has two parts: a module at the bus station and a module on the bus. Figure 1 shows the block diagram for the bus station module and Figure 2 shows the block diagram for the bus module.

Figure 1. Smart Bus Station Sign (at Bus Station) Design on DE1 Board

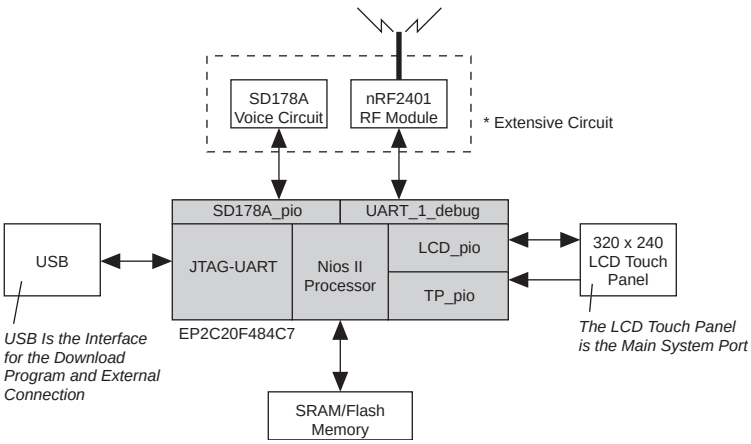
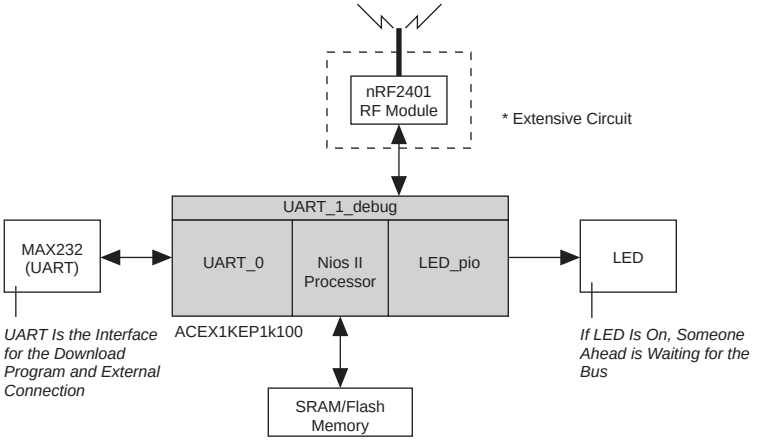


Figure 2. Smart Bus Station Sign (on Bus) Design on ACEX1K Board



Nios II Embedded System Design Diagram

Figure 3 shows the design schematic in the Quartus II software, including:

- Nios II CPU main system design
- Touch screen TOUCHCNTR design (compiled from VHDL code)
- SD178A voice circuit and program SD178A CNTR (compiled from VHDL code)

Figure 3. Design Schematic

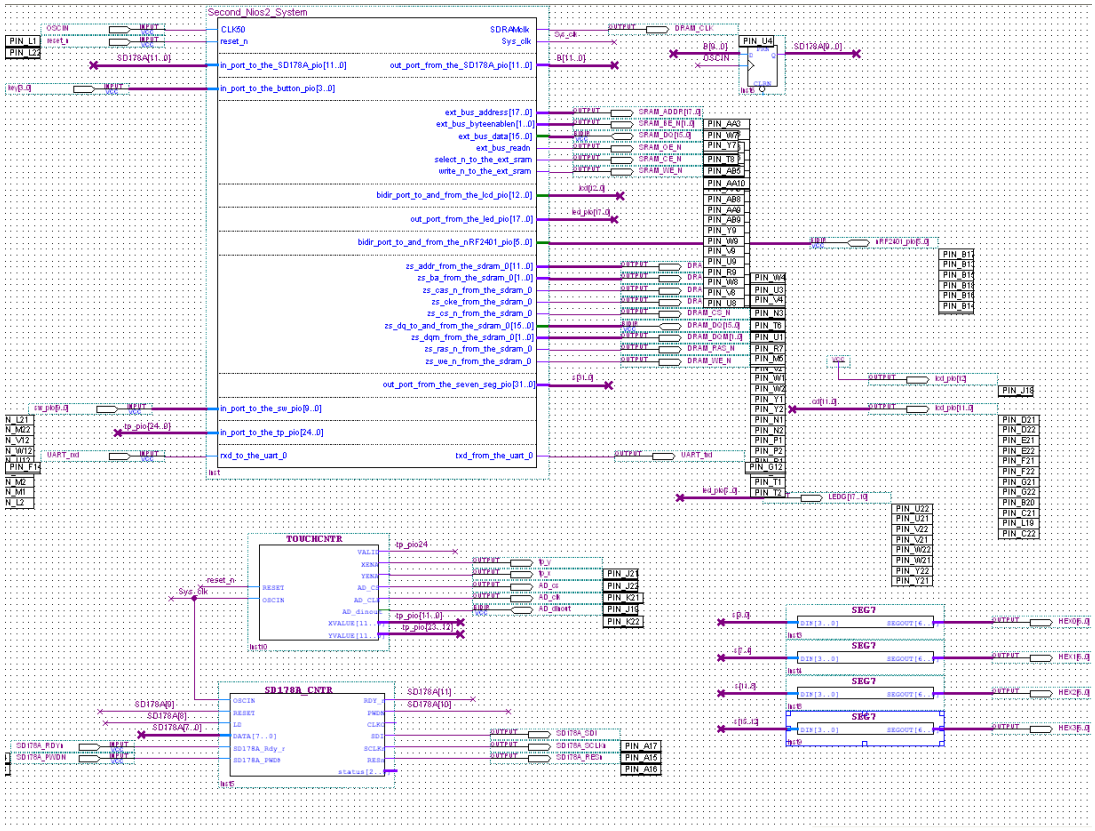
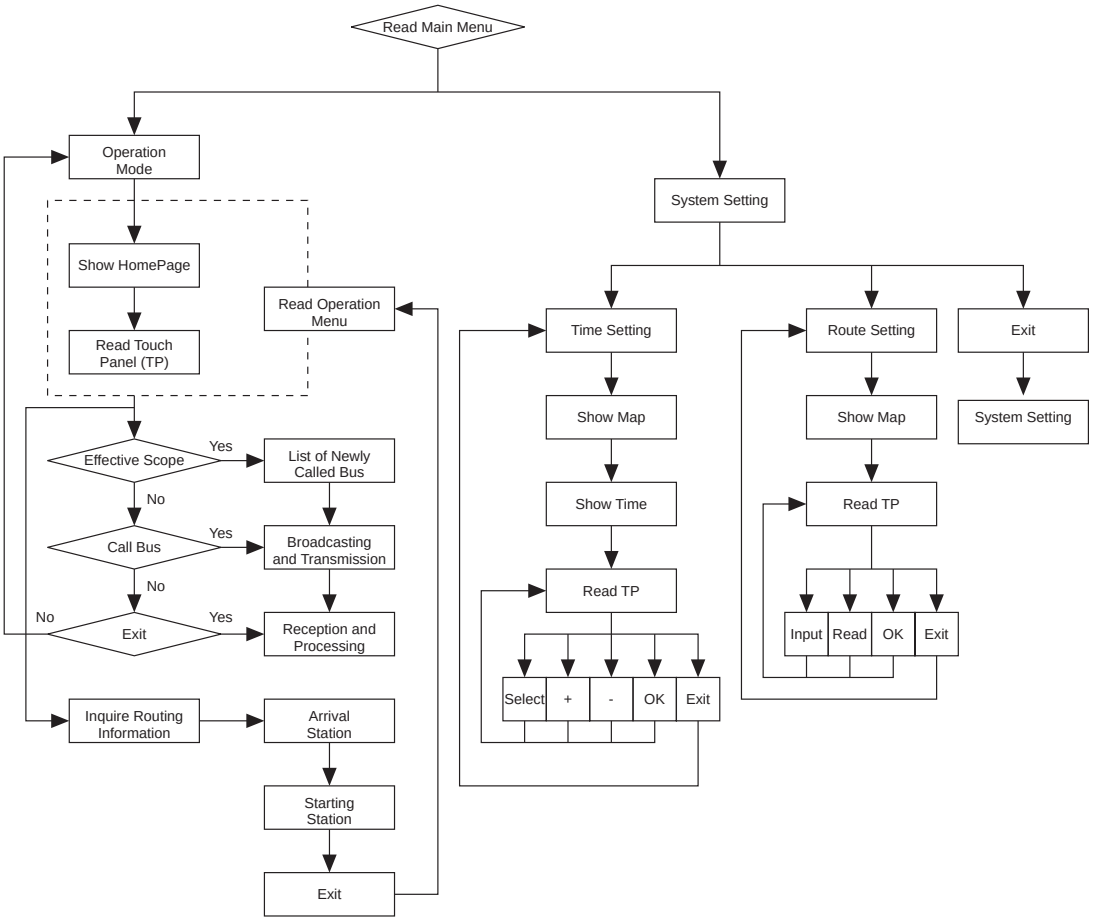


Figure 4 shows the system flow chart for the software on the module at the bus station.

Figure 4. Smart Bus Station Sign Flow Chart (Software at Bus Station)



Software Function User Interface and Flow Charts

Figure 5 shows the main program process and user interface.

Figure 5. Main Program Process and User Interface

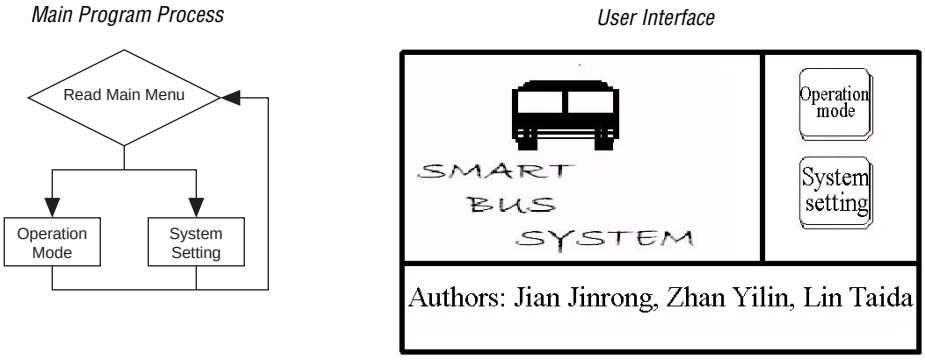


Figure 6 shows the operation mode process and user interface.

Figure 6. Operation Mode Process and User Interface

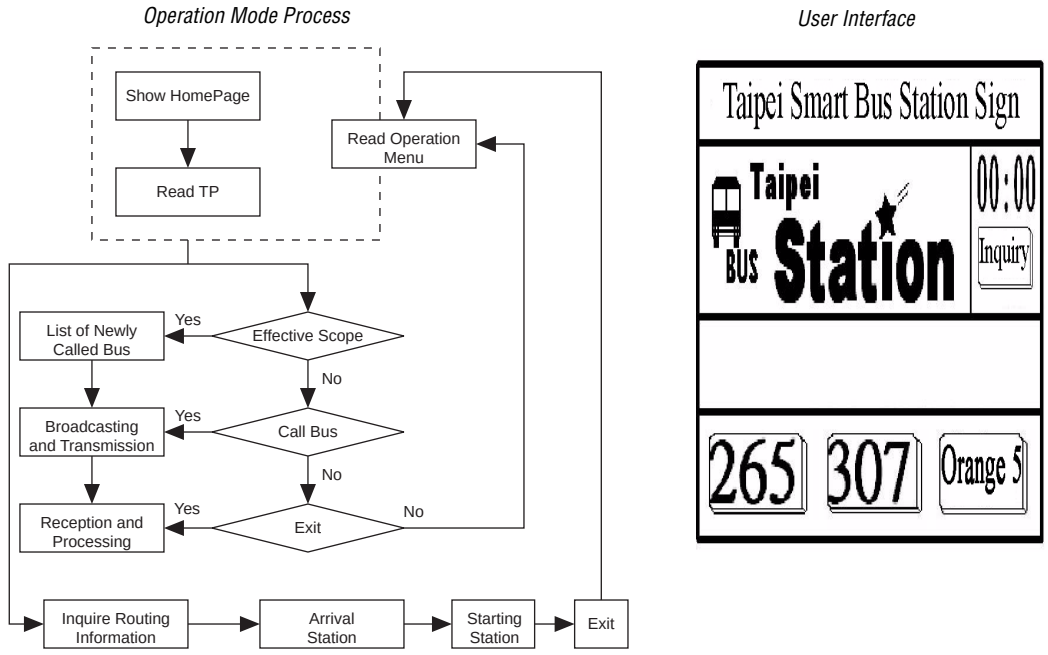


Figure 7 shows the system setting process and user interface.

Figure 7. System Setting Process and User Interface

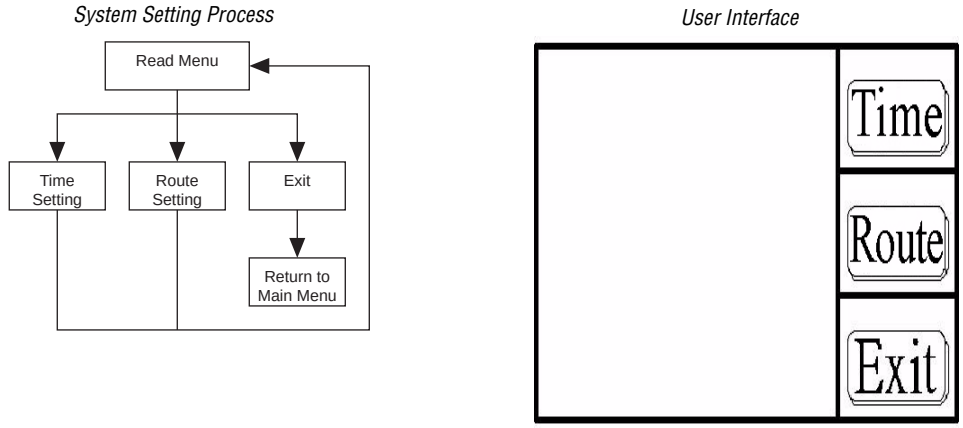


Figure 8 shows the time setting process and user interface.

Figure 8. Time Setting Process and User Interface

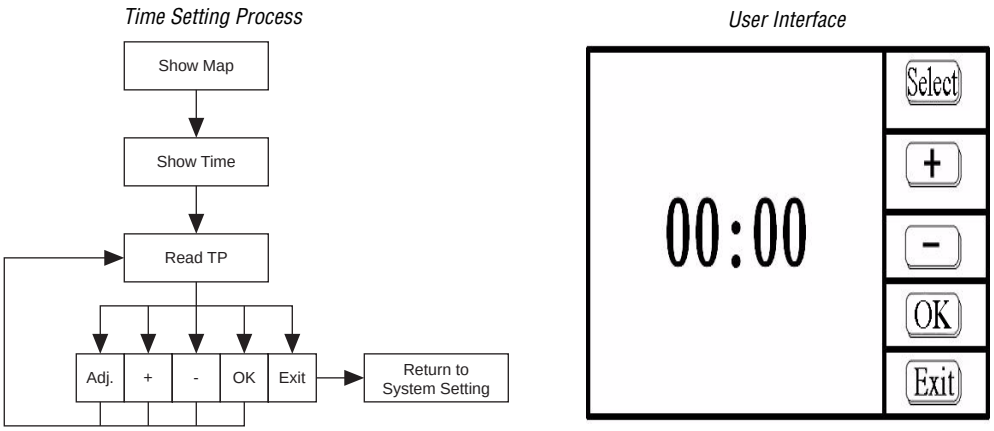
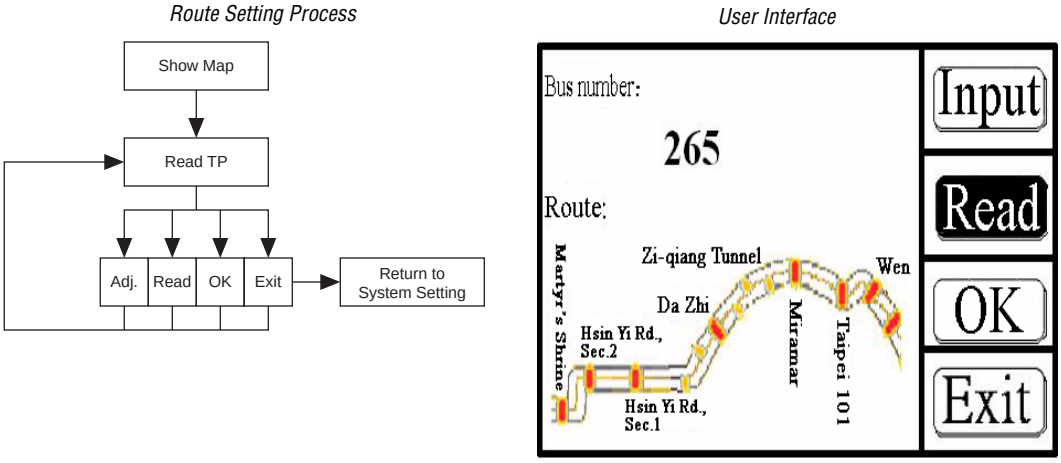


Figure 9 shows the route setting process and user interface.

Figure 9. Route Setting Process and User Interface



Design Methodology

Starting with the DE1 development board, we added a 320 x 240 touch-screen LCD, an RF radio transmission interface, and an SD178 Chinese voice circuit. We performed the following tasks:

- Designed the SD178A controller using VHDL
- Compiled the SD178A sub-drivers using the C language
- Compiled the sub-drivers using the C language.
- Compiled the nRF2401 RF module sub-drivers using the C language

We then used SOPC Builder in the Quartus II software to build the Nios II embedded system and we designed the embedded software in the Nios II Integrated Development Environment (IDE). We compiled the main program of smart bus station sign (at the bus station) using the C language.

Design Features

Our design has the following features:

- Smart bus station sign (the module at the bus station)
  - Build a Nios II embedded system using SOPC Builder
  - Create a SD178A voice circuit and an nRF2401 RF module on the DE1 board
  - Design a SD178A controller using VHDL
  - Compile SD178A sub-drivers using the C language
  - Compile sub-drivers using the C language
  - Compile nRF2401 RF module sub-drivers using the C language
  - Compile the program for the main interfaces of the digital bus station sign using the C language
- Smart bus station sign (module on the bus)
  - Build a Nios embedded system using SOPC Builder
  - Create an nRF2401 RF module on the ACEX1K development board
  - Compile nRF2401 RF module sub-drivers using the C language
  - Compile the main program control settings of the bus sub-module using the C language

Adaptability

We used the DE1 development board, touch-screen LCD, and Nios II processor to compile the smart bus station sign program with the C language in the Nios II IDE. Then, we downloaded the application to the development board and used the touch-screen LCD to input or output the images, thereby digitizing the bus station signs.

Creation

Our project revolutionizes traditional bus station signs and presents them in a modern, digital manner. The system helps users operate the interface, and differentiates between the Chinese, English, Taiwanese, etc. languages, better serving the passengers.

Conclusion

We used software and hardware integration to design the prototype system. Using the DE1 and ACEX1K development boards, we built Nios and Nios II embedded systems and compiled the main programs of host machine for the station sign and in-bus module as well as the SD178A voice circuit. The smart bus station sign prototype features convenience, safety, an excellent user interface, and Chinese voice controls. We want to incorporate these functions into real life, so we will continue to research more in this area.