

# Paperless Receipts

Point of Sale Transactions using Electronic Receipts and Near Field Communication

Brad Davis  
Nelson Stoik  
Zeshan Nurani

Group 5

A point of sale board with barcode scanner that issues electronic receipts to an Android device using Near Field Communication.

## Abstract

The goal of this project was to create a solution to eliminate using paper receipts when purchasing goods. Items to be purchased have a barcode to be scanned by a barcode reader or typed in on a keyboard. The barcode information is sent to the system and matched against the systems database to convert the number to an item name and price. Now that there is a name and price, the item is added to the running list of items for the final receipt; the items price is also added to the subtotal cost. There is an LCD which displays the name of the last item scanned as well as the running subtotal of the transaction. Once all of the items have been entered, the system processes the transaction and generates an electronic receipt. This receipt is stored into external memory and also exported to an NFC breakout module. Once the receipt is in the NFC module, it is wirelessly transferred using Near Field Communication (NFC) to an application on the purchaser's Android device. To interface the hardware and software we used an Altera DE2 to generate, save, and send paperless receipts. Software components include parsing and converting barcode information, generating a receipt, storing the receipt in external memory, generating data for the NFC module, and lastly the Android application for receiving and viewing user's receipts. We were successful integrating all the software components, the device drivers, and the hardware.

# Table of Contents

[Abstract](#)

[Table of Contents](#)

[Functional Requirements](#)

[Overview](#)

[Barcode Scanner/Keyboard](#)

[Embedded System](#)

[SD Card](#)

[NFC Module](#)

[Android Application and Phone](#)

[Design and Description of Operation](#)

[Initialization](#)

[Program Flow](#)

[Hardware](#)

[Software](#)

[Bill of Materials](#)

[Reusable Design Units](#)

[Datasheet](#)

[Performance](#)

[User Block Diagram](#)

[Operating Conditions](#)

[I/O Signals](#)

[uCOS Synchronization Structures](#)

[Power Calculations](#)

[Background Reading](#)

[RFID and NFC on Mobile Phones](#)

[Asymmetric Encryption](#)

[Type 2 Tag](#)

[Software Design](#)

[System Software](#)

[Interfacing with the PS/2 Barcode Scanner & Keyboard](#)

[Interfacing with the SD Card](#)

[Scanning Items / Generating Receipt](#)

[Saving to SD Card](#)

[Interfacing with the NFC Module](#)

[Preparing Data for NFC Transfer](#)

[Android Application](#)

[Testing Plan](#)

[External Hardware and I/O](#)

[Barcode Scanner](#)

[NFC Board](#)

[System Software](#)

[Android Application](#)

[Results of Experiment and Characterization](#)

[Android Application](#)

[PS/2 Scanner](#)

[NFC Module](#)

[Safety](#)

[Environmental Impact](#)

[Sustainability](#)

[Android Application](#)

[DE2 Board](#)

[References](#)

[Appendix A - Quick Start Manual](#)

[Setup](#)

[Running](#)

[Appendix B - Hardware Components](#)

[Appendix C - Software](#)

[Appendix D - Future Work](#)

# Functional Requirements

## Overview

This project is to design a system to eliminate paper receipts. The four major components of the project are product input, receipt creation and storage, receipt transmission to the phone, and handling the receipt on the phone.

## Barcode Scanner/Keyboard

- The barcode scanner scans and transmits barcode information to the Altera DE2 board as a Human Interface Device.
- The barcode scanner interfaces with the DE2 board via the PS/2 port
- The keyboard interfaces with the DE2 board via the PS/2 splitter on the barcode scanner's cable

## Embedded System

- The system loads the latest item database from the SD card
- The system converts barcodes codes into item names and prices
- The system can scan multiple items
- The system consolidates all scanned items into an electronic receipt
- The system can remove the last item scanned if user makes an input error
- The system saves the stores receipt data onto the SD card
- The system displays the subtotal of the transaction and the last item scanned while the transaction is in progress
- The system displays the final price (including tax) when the transaction is complete
- The system prepares the receipt for the NFC module by storing NFC Data Exchange Format (NDEF) messages into NFC Type 2 tags

## SD Card

- The SD card can communicate with the Altera DE2 Board via the Serial Peripheral Interface
- The SD card stores the item database of barcodes, names, and prices
- The SD card stores copies of all the past receipts generated

## NFC Module

- The NFC module is able to communicate with the Altera DE2 via the GPIO
- The NFC module can send the NFC Type 2 tags to the Android device

## Android Application and Phone

- The Android phone communicates with the NFC module via the NFC antenna
- The Android app can extract data from the NFC antenna built into the phone
- The Android app and phone can load and store the receipt information into local memory
- The Android app allows users to view all past receipt information

# Design and Description of Operation

## Initialization

On system startup:

- The NFC module is initialized and linked to the Altera DE2
- The PS/2 port is initialized and linked to the barcode scanner
- The SD card is initialized, linked, and the files directory is scanned
- The system loads the database of scannable items into volatile memory from the SD card

## Program Flow

- The user starts the process by pressing the start button on the board to start a new transaction
- Product input is done with a barcode scanner. We scan the items barcode and transfer the data to the board using an PS/2 connection. The data can also be entered using the PS/2 keyboard
- Our system software then searches the database for the item price and name
- The item is added to the ongoing transaction information that stores all previously scanned items. If the item is a duplicate on the receipt, the quantity is incremented. The subtotal price is incremented by the item's price
- While items are being scanned, the onboard LCD displays the name of the last item scanned as well as the running subtotal
- Once all of the items have been scanned and the customer is ready to end the transaction, the end button is pressed. The tax is calculated and added to the receipt
- The final price is displayed on the onboard LCD
- The receipt is stored onto the external SD card
- The receipt data is broken into NDEF messages and each message is loaded into an NFC Type 2 Tag
- Once all the tags are ready, the user is notified (using the onboard LCD) to tap their phone to the NFC module
- When the phone is linked to the NFC module, the tags begin to transmit to the phone
- The Android app reads each NDEF message and formats the receipt for viewing within the Android application
- The user is able to review all of his or her receipts using the Android application.

See the User Block Diagram in the Datasheet for simple control flow.

## Hardware

See Appendix B

## Software

See Software Design

## Bill of Materials

Part	Description	Link	Datasheet	Price
NFC transmitter / receiver [1]	NFC IC and breakout board for communication with the DE2	<a href="http://www.adafruit.com/product/364">http://www.adafruit.com/product/364</a>	<a href="https://learn.adafruit.com/system/assets/assets/000/009/730/original/rfid_nfc_PN532_v1.3.png?1396896145">https://learn.adafruit.com/system/assets/assets/000/009/730/original/rfid_nfc_PN532_v1.3.png?1396896145</a>	\$39.95
Barcode Scanner [2]	Scanner for 1D barcodes	<a href="https://www.adafruit.com/products/1202">https://www.adafruit.com/products/1202</a>	<a href="http://www.adafruit.com/datasheets/MCR12_specsheet.pdf">http://www.adafruit.com/datasheets/MCR12_specsheet.pdf</a>	\$69.95
Android Device - Nexus 5	Android device with NFC abilities	<a href="https://play.google.com/store/devices/details/Nexus_5_16GB_Black?id=nexus_5_black_16gb&amp;hl=en">https://play.google.com/store/devices/details/Nexus_5_16GB_Black?id=nexus_5_black_16gb&amp;hl=en</a>	<a href="https://play.google.com/store/devices/details/Nexus_5_16GB_Black?id=nexus_5_black_16gb&amp;hl=en">https://play.google.com/store/devices/details/Nexus_5_16GB_Black?id=nexus_5_black_16gb&amp;hl=en</a>	\$349.00*
SD Memory Card	SD Card for storing item database off board	<a href="http://www.staples.ca/en/Kingston-16GB-SDHC-Card-Class-10/product_935625_2-CA_1_20001?externalize=certona">http://www.staples.ca/en/Kingston-16GB-SDHC-Card-Class-10/product_935625_2-CA_1_20001?externalize=certona</a>	<a href="http://www.staples.ca/en/Kingston-16GB-SDHC-Card-Class-10/product_935625_2-CA_1_20001?externalize=certona">http://www.staples.ca/en/Kingston-16GB-SDHC-Card-Class-10/product_935625_2-CA_1_20001?externalize=certona</a>	\$11.95
Altera/Terasic DE2 development board	Development board with a rich set of features	<a href="http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&amp;CategoryNo=165&amp;No=30&amp;PartNo=4">http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&amp;CategoryNo=165&amp;No=30&amp;PartNo=4</a>	<a href="http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&amp;CategoryNo=165&amp;No=30&amp;PartNo=4">http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&amp;CategoryNo=165&amp;No=30&amp;PartNo=4</a>	\$517.72
Perf Board	50 X 80MM SINGLE-HOLE BREADBOARD	<a href="http://www.allelectronics.com/make-a-store/item/sb-24/50-x-80mm-single-hole-breadboard-double-sided/1.html">http://www.allelectronics.com/make-a-store/item/sb-24/50-x-80mm-single-hole-breadboard-double-sided/1.html</a>		\$4.00
40-pin ribbon cable	GPIO Ribbon Cable	<a href="https://www.adafruit.com/products/1988">https://www.adafruit.com/products/1988</a>		\$2.95

\* Device owned by group members. No purchase necessary.

## Reusable Design Units

- Tiny Encryption Algorithm
  - Use this application note as a starting point to learn about encryption and the DE2 board; from here we can pursue AES encryption, a proposed enhancement
  - How to use the Tiny Encryption Algorithm to Encrypt and Decrypt [3]
    - Source Size: TBD, Compiled Sized: TBD, Performance: To be tested, Resource Requirements: TBD
- GPIO
  - Use the GPIO on the Altera DE2 board to interface with our hardware
  - GPIO Application Note for Altera DE2 Development and Education Board [4]
  - I2C Device Integration [5] [6]
  - GPIO Internal Pull-Up Resistor [7]
  - Source Size: 11.9 KB, Compiled Sized: TBD, Performance: To be tested, Resource Requirements: TBD
- SD Card Interfacing
  - Use the SD card slot on the Altera board to enable the proposed enhancement of storing the database and receipts on an external SD card
  - SD Card Interfacing [10]
  - Source Size: TBD, Compiled Sized: TBD, Performance: To be tested, Resource Requirements: TBD
- NFC
  - Use an NFC board to communicate between the system and the user's mobile device
  - NFC Reader / Writer Library [12]
  - Source Size: 9.6KB, Compiled Sized: TBD, Performance: To be tested, Resource Requirements: TBD

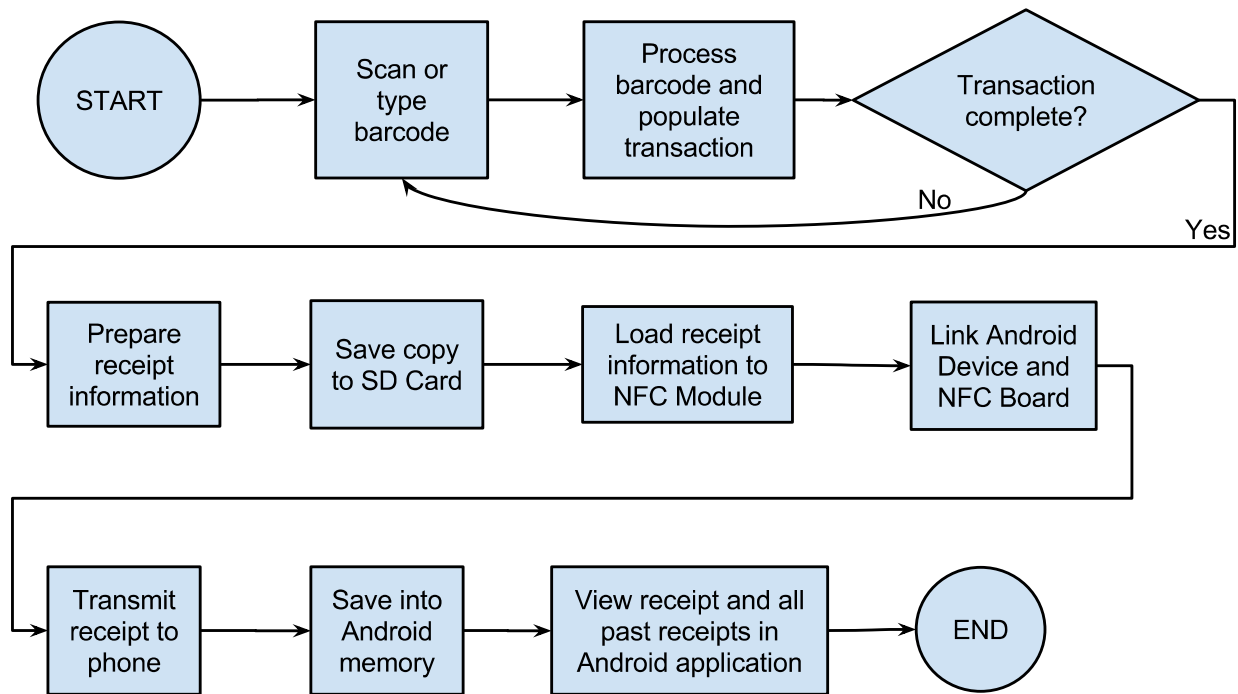
## Datasheet

### Performance

- Our project is capable of scanning one barcode every second.
- All one dimensional UPC barcodes can be scanned, provided the item has been entered into the item database on the SD card.
- The receipt can have as many items as there are in the item database.
- The total storage size of the system, which includes the item database and each generated receipt, is 2 GB.
- The Android application works on all Android phones that have an NFC antenna and Android Lollipop (5.0) operating system or greater.
- The NFC board can transfer the receipt to the Android phone at a rate of up to 424 kbits/second at a distance of up to 10 centimeters.



## User Block Diagram



## Operating Conditions

The project is designed to operate in an indoor and protected environment. It needs to be protected from moisture, and dust. The operating temperature should be +/- 10 °C of room temperature.

## I/O Signals

Device	Signal Name	Interface	Description
LED	LED_X	Onboard	X is a number between 1 and 8 for the 8 LED onboard lights
KEY3	start_button_isr	Onboard	button pressed to start a transaction
KEY2	end_button_isr	Onboard	button pressed to end a transaction
KEY1	delete_button_isr	Onboard	button pressed to delete last item scanned
PS/2 Barcode Scanner	KB_Data	PS2	Data from PS2 device
	KB_Clock	PS2	Clock for PS2 device

	GND	PS2	ground
	Vcc	PS2	+5 V
NFC	SCK	I2C	slave clock
	MISO	I2C	master input
	MOSI	I2C	master output
	SSEL	I2C	slave select
	Vcc	I2C	+3.3 V
	GND	I2C	ground

## uCOS Synchronization Structures

Device	Signal Name	File	Description
Semaphore's	mainSem	receipt_func.h	used to signal that a transaction has started
	lastItemSem	receipt_func.h	used to signal the item removal task to run
	ps2_input_ready_sem	ps2_func.h	signals that there is data from ps2 device
	wait_for_data_to_be_read	ps2_func.c	pause ps2 input until last data has been read
	wait_for_input_enabled	ps2_func.c	start and stop ps2 input
Mutex's	scanMutex	receipt_func.h	protects the scanning integer
	dbMutex	receipt_func.h	protects the variable that holds the database size
	lastItemMutex	receipt_func.h	protects the variables used to track the last item scanned
	r1Mutex	receipt_func.h	protects the global receipt struct
	last_scanned_mutex	ps2_func.c	protects last scanned data from ps2

	input_mutex	ps2_func.c	protects ps2 flow control variable
Interrupt's	start_button_isr	KEY3	button pressed to start a transaction
	end_button_isr	KEY2	button pressed to end a transaction
	delete_button_isr	KEY1	button pressed to delete the last item scanned
	NFC_interrupt	Software	Signal to the controller that the NFC has data available

## Power Calculations

There will be three different power modes to calculate. The modes are standby mode, receipt creation mode, and receipt transfer mode.

Power in Standby Mode =  $0.41 \text{ A} * 9.03 \text{ V} = 3.70 \text{ W}$

Power in Receipt Creation Mode =  $0.50 \text{ A} * 9.00 \text{ V} = 4.50 \text{ W}$

Power in Receipt Transfer Mode =  $0.43 \text{ A} * 9.02 \text{ V} = 3.88 \text{ W}$

There needs to be some assumptions on the usage of the system. First assume that the system will be powered on for 8 hours a day, every day of the year. Next assume that during every hour of operation there will be 15 transactions taking 2 minute each in Receipt Creation Mode and 15 seconds in Receipt Transfer Mode. The remaining time will be in Standby Mode. Therefore:

Time in Receipt Creation Mode =  $15 \text{ transactions} * 2 \text{ min} * (1 / 60) * 8 \text{ hrs} = 4 \text{ hours} / \text{day}$

Time in Receipt Transfer Mode =  $15 \text{ transactions} * 15 \text{ secs} * (1 / 3600) * 8 \text{ hrs} = 0.5 \text{ hours} / \text{day}$

Time in Standby Mode =  $8 - 4 - 0.5 = 3.5 \text{ hours} / \text{day}$

Standby Mode Power =  $3.70 \text{ W} * 3600 \text{ seconds/hour} = 13.32 \text{ kWh}$

Receipt Creation Power =  $4.50 \text{ W} * 3600 \text{ seconds/hour} = 16.20 \text{ kWh}$

Receipt Transfer Power =  $3.88 \text{ W} * 3600 \text{ seconds/hour} = 13.97 \text{ kWh}$

Weighted Average =

$$\frac{3.5 \text{ hours}}{24 \text{ hours}} * 3.70 \text{ Wh} + \frac{4 \text{ hours}}{24 \text{ hours}} * 4.50 \text{ Wh} + \frac{0.5 \text{ hours}}{24 \text{ hours}} * 3.88 \text{ Wh} + \frac{16 \text{ hours}}{24 \text{ hours}} * 0 = 1.370 \text{ W} / \text{hour}$$

Total power usage per year =  $1.370 \text{ W/h} * 24 \text{ hrs/day} * 365 \text{ days/year} = 12.00 \text{ kWh} / \text{year}$

## Background Reading

### RFID and NFC on Mobile Phones

The article titled "RFID and NFC on Mobile Phones" [11] described how NFC is used on mobile phones. At a top level view, NFC is used by two devices to communicate over radio frequencies (RF). There is protocols in place that outline how two devices can communicate with each other. The devices use the NFC Data Exchange Format (NDEF) to send information between them.

We plan on using the NFC to communication between the Android application and the Altera board. In our system, the NIOS II will send the NDEF messages to the attached NFC module. The module will then connect with the Android phone and transfer the NDEF messages to the application on the device.

### Asymmetric Encryption

The article titled "An Asymmetrical Encryption Based Automated Trust Negotiation Model" [22] describes a model which is much simpler than the traditional Automated Trust Negotiation (ATN) called the Asymmetrical Encryption Based ATN Model (AEBM). This new model allows for strangers to communicate information in an encrypted manner with a protocol which is much more efficient and simple than previous ATN strategies.

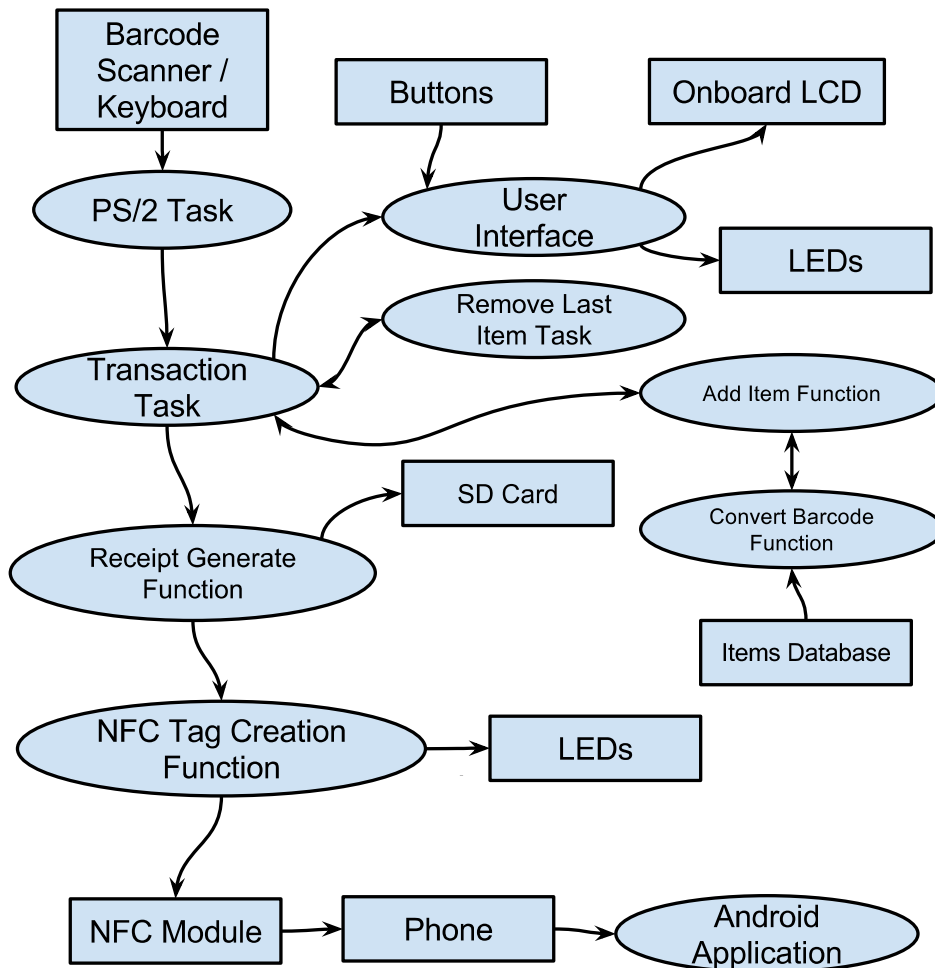
The article was used to learn public and private key encryption process and how the two work together to provide a secure way to store and transfer data. We will be using private and public keys when working on the encryption enhancement for the project. Once progress is made toward the enhancement, a section in Software Design will be created to describe the operation of the encryption algorithm.

### Type 2 Tag

The article "Type 2 Tag Operation" [25] provides technical Specifications on the NDEF forum type 2 tag. It provides an in-depth description of the structure that the NDEF messages are stored within memory. It also describes the communication protocol and commands necessary in order read and write the NDEF messages.

The article was used to generated the communication code necessary to emulate the NDEF messages on the PN532 breakout board that was used in our project. The document provided the information for generating the packages with appropriate headers and package lengths. The appropriate ACK's and NACK's are outlined and these were used in debugging and testing during the development state.

## Software Design



## System Software

### Interfacing with the PS/2 Barcode Scanner & Keyboard

The barcode scanner and keyboard were connected to the DE2 board via the PS/2 interface. The barcode scanner communicated its data as if it was a keyboard, so we were able to use the Altera provided PS/2 hardware component and drivers for keyboard and mouse input. The keyboards send data by generating unique make and break codes for each key pressed. These codes are captured and stored by the hardware component. We then read and decoded these codes which converted them into ASCII strings broken up by a special character. The ASCII strings were then passed on to the DE2 board for further processing.

### Interfacing with the SD Card

The SD card was interfaced using an open source library called the Embedded File System Library (EFSL). This library provided all the functionality to create, read, and write to files on a SD card in the DE2 environment. We built specialized functions on top of the EFSL to provide

our system with the exact options we needed like creating unique files to store the generated receipts and reading in the item database from a file at system startup.

### Scanning Items / Generating Receipt

Once we are ready to start scanning items, we begin by hitting the “start” button on the DE2 board. After the button is pressed, the previous transaction information and barcode reader buffer is cleared. It also allows the PS/2 task to start reading from the scanner. We scan an item; the PS/2 task is polling the barcode reader and notifies the transaction task there is a barcode available. If a barcode is not available, the PS/2 task continues to poll. When there is a barcode available, the barcode is fetched from the PS/2 task and compared against the database. The item database is in volatile memory which we loaded from the SD card on startup. When we scan an item and the barcode matches with one in the database, it is added to the receipt structure that can dynamically grow as more items are scanned; if the item already exists in the receipt, the quantity is incremented. Next, the transaction subtotal is incremented and the LCD is updated with the name of the last item scanned.

### Saving to SD Card

Once we are done scanning items, the user can hit the “end” button. An interrupt sends a signal disabling any further scanning by disabling the scanner and the receipt begins to prepare for saving to the SD card. First we compute the tax and total; next we format the receipt into data payload. This payload is written line by line onto the SD card in a new, uniquely named file within the receipts directory.

### Interfacing with the NFC Module

The NFC module is connected to the DE2 board with the I2C protocol. The NFC board is put into card emulation mode when the system is first powered on. This makes the NFC module appear as a NFC Type 2 tag to any reader that comes into range. Then during the receipt transfer to the Android device, the NFC module receives read commands from the Android device and responds to them by sending the data over in 16 byte chunks according to the Type 2 tag specification.

### Preparing Data for NFC Transfer

Research is still being performed on the best way to communicate with the NFC module and how the module will communicate with the phone. A quick overview and educated guesses for the software design is as follows:

1. parse the receipt structure and populate data structure which will be sent to the NFC module (most likely a NDEF package)
2. send package to NFC module
3. check the checksums sent back from the NFC module that the file was sent correctly

### Android Application

The Android application developed will have the ability to receive a new receipt through NFC and to view previous receipts. The main screen will have buttons with the options to view past receipts or to accept a new receipt. When the “view receipts” button is selected, the application transitions to a view which displays the contents of the data folder which holds all of the receipts. The user then has the ability to select a receipt from the list and view its contents. When the “receive receipt” button is selected, a different view opens prompting the user to

collect a new receipt through NFC. Once the phone makes the NFC connection, the user is notified through the application. The application will receive the file and save it to the data folder specific for the application.

## Testing Plan

### External Hardware and I/O

#### Barcode Scanner

- Add the PS/2 controller to the FPGA board and use the Altera drivers to build a modified driver for reading from a PS/2 device
  - Connect and interface with a PS/2 keyboard to read input from the keyboard to the console.
  - The program now takes input from the PS/2 controller and parses it into a string that is printed to the console. Allows us to verify correct input.
- Attach the barcode scanner and scan barcodes; the scanner is designed to interface like a human input device (HID).
  - The barcode scanner will beep when a successful barcode has been scanned.
  - User confirms string is correct by viewing the console.
- Next, a testing task was created to pause until input was available from the PS/2 controller. When a barcode became available, it would read it and print it to the console
  - This tested that the flow control of the PS/2 driver was working correctly.
  - Simulated the final design of signaling when input was available and transferring it.
- Finally, we integrated the PS/2 driver with the Transaction Task code and confirm that the flow control and operation is correct.

#### NFC Board

- With the NFC board soldered, an FTDI cable can be used to connect the board to a Linux or Apple computer. Then a demo open source program can be run to have the board check for nearby NFC devices.
  - This will confirm that the board hardware is functioning correctly.
- Next connect the NFC board to the DE2 board and use the app note from a previous group [12] to configure the NFC board to operate in tag reader mode
  - This will confirm that the communication between the NFC board and the DE2 board is working correctly.
- Then modify / develop the NFC driver to operate in peer to peer mode and send simple test data between the NFC board and another NFC device.
- Next integrate the NFC components into the overall project to send receipts to an Android application.
- Visual Identifiers to test NFC board:
  - When the board is ready for an Android device, send a message to the DE2 boards LCD.
  - When the NFC board detects an NFC device, print message to console.
  - Have a blinking LED light when there is a data transfer.
  - Have a light turn on when the transfer is completed successfully.

## System Software

- Test the search functionality by sending a test barcode to the task
  - Verified the item could be found within the database
- Scan item before “start button” is pressed
  - Ensured barcode scanner is flushed before a transaction has started
- Hit start button twice to check that transaction flow does not become corrupt
- Print contents of receipt after each scan
  - Verify the receipt is dynamically allocating memory
- Print contents of receipt after memory deallocated
- Disable barcode reader and try scanning items
- Mid-transaction hit reset button to see if process can restart correctly
- Hit “Item Remove” button multiple times
  - Ensure only last item scanned can be removed

## Android Application

- Build single page “hello world” app and install on Android phone to ensure compatibility between the phone and the development environment.
- Experiment with a larger application including different pages and transitions.
  - Testing the functionality of transitioning between different views and windows.
- Create a debugging feature to generate an on demand receipt to test storage on device.
  - When a test button is pressed a test file is generated with the contents as a timestamp to verify the File I/O between the android application and phone.
- Add the ability to view the folders where the debugging receipts are being stored and the actual debugging receipts.
- Test code which performs the NFC communication.
  - Test code using NFC tags until the NFC module has been implemented
  - Once the NFC breakout board communicates with the DE2, Static messages were generated and receiving these NDEF messages were tested within the application
  - The application was then tested by sending it dynamically generated NDEF messages

## Results of Experiment and Characterization

### Android Application

Some experimentation has been done with the development of the Android application. Because no one in our group has worked on Android development before some very basic experiments were performed. First a small single screen hello world application was developed using the Android development environment. Once that application was successfully compiled it was packaged and installed on an a device.

Next experiments and testing were done with different application components such as buttons, transitions, and different windows. The three buttons that are currently used can be seen in Figure 1. After the basic framework for different windows and the transitions were set, file IO



was tested within the application. A “Generate Test” button was added to the main screen which creates a simple text file, representative of a receipt, and stores that file within the application data folder. The Receipt view window now queries the application data folder for all the receipts stored, this can be seen in Figure 2. When an item is selected from the list, the contents of the file are read out to a new window.

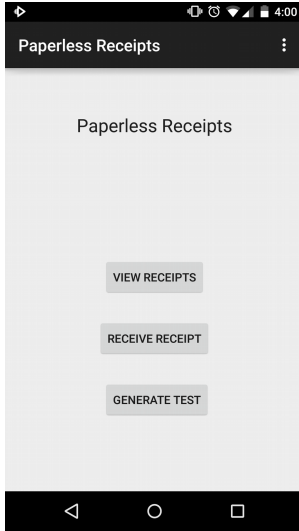


Figure 1.

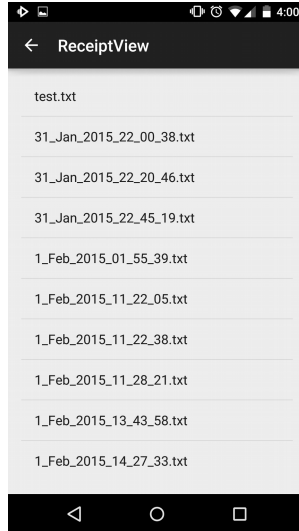


Figure 2.

Once the code for the NFC communication was added, Experiments were done to test that the application was able to detect another NFC device. After testing and some changes to the code the application was able to confirm that it was making NFC connections. The testing of the NFC code started by connecting the NFC module to a test laptop and using test code from the open source lib-nfc library. The test codes emulated an NFC forum type two tag and the android application was tested to see if it could read the tags.

After communication between the altera DE2 board and the NFC Breakout board the application was tested for the ability to read messages generated by the DE2 and loaded on the NFC module. The application was first sent static messages for testing purposes, which it was able to receive and process. Following the development of dynamic message generation by the DE2 the application was tested to ensure that those messages could be received as well.

## PS/2 Scanner

The final version of the PS/2 scanner was able to successfully scan more than 2 barcodes every second (21 barcodes successfully scanned in 10 seconds). If two barcodes are visible in the field of view of the scanner, either one of the barcodes may be scanned. To ensure that only one barcode is scanned at a time, care must be taken to make sure only one barcode is in the field of view of the scanner when scanning.

Tradeoffs between using polling and interrupt driven approaches to read data from the PS/2 device were considered. The PS/2 controller device has an interrupt signal available in the Qsys component setup. However the PS/2 controller required the use of the legacy interrupts in the

NIOS II system, and we encountered difficulties having legacy and non legacy interrupts that are used by other tasks to co-exist. Due to this reason, we decided to use polling to check for input from the PS/2 scanner. Even though it is not as efficient as the interrupt driven approach, throughout all our testing the extra CPU usage was never an issue. While the system is polling for data from the PS/2 controller, almost no other important calculations are being performed by other tasks.

## NFC Module

Once the NFC breakout board was built by soldering the pin headers to it, the functionality of the module was tested by connecting the board to an open source program that runs on Linux called libnfc. This confirmed that the NFC breakout board hardware was functioning as required. This program was also used to reverse engineer the steps required to send data to another device using NFC.

We first copied the functionality of the example program from libnfc so that the DE2 board would do the same thing. This example program sent a simple URL to any NFC reader in range. This showed that we had an understanding of how the communication protocol was used in NFC and allowed us to build off of it to expand the functionality of our breakout board. We then modified this program so that it was able to send at first static text messages, and finally dynamically generated text messages up to 1 KB in size. This upper limit was imposed by the Type 2 tag that we had chosen to emulate. This was the final goal of the NFC module and was tested extensively by sending dynamically generated receipts from the DE2 board to the mobile application on the Android device.

## Safety

The only external power source used is the 9V power supply for the DE2 board. The NFC board is powered by a 3.3V pin on the board GPIO header. The barcode scanner is powered via the PS2 interface on the DE2 board. The current and power draw of all the devices are not significantly large. Since the voltage, current, and power usage are all very low they are not major safety concerns. The barcode scanner uses a red light for contrast. Care will have to be taken so that the light is not pointed into a person's eyes. The light is not intense enough to do damage to the retina as it is not a laser. There are no moving parts in the design. The Android application does not have any safety concerns. The design may heat up slightly but not at high enough temperatures draw concern. To protect all the equipment, grounding straps must be used to prevent static discharge from damaging any of the components.

## Environmental Impact

This project does not contain any hazardous materials for operation. The following chart describes the RoHS compliance status of the devices and parts used.

RoHS Compliant	RoHS Non-Compliant
PN532 NFC Module	Altera DE2 Board

Barcode Reader/Scanner Module	Solder, Wires, Resistors
Nexus 5 Smartphone [21]	

## Sustainability

This project is split into two separate hardware sections, the Altera DE2 board with the PS2 scanner and NFC module as well as the Android device running the receipt application.

In Edmonton it currently costs 6.583 cents per kWh. The majority of the electricity produced in Edmonton is generated by burning coal. This produces 0.989 kg CO<sub>2</sub>/kWh

### Android Application

It is difficult to accurately measure how much energy a specific application uses on an Android device. However it has been calculated that the average power consumption of a smartphone device for a year is about 4.2 kWh [19]. If our application uses up to 1% of the entire power consumption of the device over the whole year (a high estimate), the cost over an entire year would be:

$$4.2 \text{ kWh / year} * 1\% * 0.06583 \text{ \$ / kWh} = 0.003\text{\$ / year}$$

The CO<sub>2</sub> produced by this energy would be:

$$4.2 \text{ kWh} * 1\% * 0.989 \text{ kg CO}_2\text{/kWh} = 0.042 \text{ kg CO}_2 \text{ / year}$$

### DE2 Board

For the DE2 board, there will be three different power modes to calculate. The modes are standby mode, receipt creation mode, and receipt transfer mode. Standby mode represents the low power state mode that the system is in when it is waiting for a receipt transaction to be initiated. Receipt creation mode represents the time from when the button is pressed to start a receipt until the button is pressed to stop the receipt creation. Finally receipt transfer mode represents the time taken to transfer the receipt to memory and to the Android device via NFC.

$$\text{Power in Standby Mode} = 0.41 \text{ A} * 9.03 \text{ V} = 3.70 \text{ W}$$

$$\text{Power in Receipt Creation Mode} = 0.50 \text{ A} * 9.00 \text{ V} = 4.50 \text{ W}$$

$$\text{Power in Receipt Transfer Mode} = 0.43 \text{ A} * 9.02 \text{ V} = 3.88 \text{ W}$$

There needs to be some assumptions on the usage of the system. First assume that the system will be powered on for 8 hours a day, every day of the year. Next assume that during every hour of operation there will be 15 transactions taking 2 minute each in Receipt Creation Mode and 15 seconds in Receipt Transfer Mode. The remaining time will be in Standby Mode. Therefore:

$$\text{Time in Receipt Creation Mode} = 15 \text{ transactions} * 2 \text{ min} * (1 / 60) * 8 \text{ hrs} = 4 \text{ hours / day}$$

$$\text{Time in Receipt Transfer Mode} = 15 \text{ transactions} * 15 \text{ secs} * (1 / 3600) * 8 \text{ hrs} = 0.5 \text{ hours / day}$$

$$\text{Time in Standby Mode} = 8 - 4 - 0.5 = 3.5 \text{ hours / day}$$

Standby Mode Power = 3.70 W \* 3600 seconds/hour  
= 13.32 kWh

Receipt Creation Power = 4.50 W \* 3600 seconds/hour  
= 16.20 kWh

Receipt Transfer Power = 3.88 W \* 3600 seconds/hour  
= 13.97 kWh

Weighted Average =

$$\frac{3.5 \text{ hours}}{24 \text{ hours}} * 3.70 \text{ Wh} + \frac{4 \text{ hours}}{24 \text{ hours}} * 4.50 \text{ Wh} + \frac{0.5 \text{ hours}}{24 \text{ hours}} * 3.88 \text{ Wh} + \frac{16 \text{ hours}}{24 \text{ hours}} * 0 = 1.370 \text{ W / hour}$$

Total power usage per year = 1.370 W/h \* 24 hrs/day \* 365 days/year  
= 12.00 kWh / year

Cost of power = 12.00 kWh \* 0.06583 \$ / kWh = 0.78 \$

The CO2 produced by this energy would be:  
12.00 kWh \* 0.989 kg CO2/kWh = 11.868 kg CO2 / year

## References

[1] NFC/RFID controller breakout board - v1.6

Adafruit - 01/15/2015

<http://www.adafruit.com/product/364>

[2] Barcode Reader/Scanner Module - CCD Camera - PS/2 Interface

Adafruit - 01/30/2015

<https://www.adafruit.com/products/1202>

[3] How to use the Tiny Encryption Algorithm to Encrypt and Decrypt

Appnotes from Anita Has and Kristina Suen - 01/17/2015

<http://www.ece.ualberta.ca/~elliott/ece492/appnotes/2010w/TEA/>

[4] GPIO Application Note for Altera DE2 Development and Education Board

Appnotes from Robert Miller - 01/17/2015

[https://www.ualberta.ca/~delliott/local/ece492/appnotes/2013w/G2\\_GPIO\\_Interfacing/](https://www.ualberta.ca/~delliott/local/ece492/appnotes/2013w/G2_GPIO_Interfacing/)

[5] I2C Device Integration

Appnotes from Braedan Jongerius, Michael Jun and Scott Hewson - 01/17/2015

[https://www.ualberta.ca/~delliott/local/ece492/appnotes/2013w/G6\\_I2C\\_Device\\_Integration/](https://www.ualberta.ca/~delliott/local/ece492/appnotes/2013w/G6_I2C_Device_Integration/)

[6] I2C Device Integration Improved

Appnotes from Daniel Fiske, Michael Lam, and Daniel Tiam - 01/17/2015

[https://www.ualberta.ca/~delliott/local/ece492/appnotes/2014w/G5\\_I2C\\_Driver\\_Improved/](https://www.ualberta.ca/~delliott/local/ece492/appnotes/2014w/G5_I2C_Driver_Improved/)

[7] GPIO Internal Pull-Up Resistor

Appnotes from Qingyue (Eric) Zhou and Elysia Jong - 01/17/2015

[https://www.ualberta.ca/~delliott/local/ece492/appnotes/2013w/GPIO\\_internal\\_pullup\\_resistor/AppNote\\_GPIO\\_Internal\\_Pullup\\_Resistor.pdf](https://www.ualberta.ca/~delliott/local/ece492/appnotes/2013w/GPIO_internal_pullup_resistor/AppNote_GPIO_Internal_Pullup_Resistor.pdf)

[8] Ethernet and Web Server - Altera DE2: DM9000A Ethernet Controller Application Notes

Appnotes from Tarek Kaddoura and Jigar Nahar - 01/17/2015

[https://www.ualberta.ca/~delliott/local/ece492/appnotes/2013w/Ethernet\\_DM9000A/](https://www.ualberta.ca/~delliott/local/ece492/appnotes/2013w/Ethernet_DM9000A/)

[9] Web Server App Note Appendix

Appnotes from Andrew Maier and Sila Luckanachai - 01/17/2015

[https://www.ualberta.ca/~delliott/local/ece492/appnotes/2013w/G9\\_WebServer\\_Appendum/WebServer/appnote.html](https://www.ualberta.ca/~delliott/local/ece492/appnotes/2013w/G9_WebServer_Appendum/WebServer/appnote.html)

[10] Interfacing SD Card

Appnotes from Jason Brown & Brady Thornton - 01/17/2015

[https://www.ualberta.ca/~delliott/local/ece492/appnotes/2013w/SD\\_card\\_interfacing/](https://www.ualberta.ca/~delliott/local/ece492/appnotes/2013w/SD_card_interfacing/)

[11] RFID and NFC on Mobile Phones

02/01/2015

<http://www.crcnetbase.com/login.ezproxy.library.ualberta.ca/doi/pdfplus/10.1201/9781420055009.ch21>

[12] NFC Reader / Writer Library

Appnotes from Daniel Fiske, Michael Lam, and Daniel Tiam - 01/17/2015

[https://www.ualberta.ca/~delliott/local/ece492/appnotes/2014w/G5\\_NFC\\_Reader\\_PN532/](https://www.ualberta.ca/~delliott/local/ece492/appnotes/2014w/G5_NFC_Reader_PN532/)

[13] Altera Embedded Processor

Nios II Processor: The World's Most Versatile Embedded Processor - 01/30/2015

<http://www.altera.com/devices/processor/nios2/ni2-index.html>

[14] Google Android Development

Android Development Studio Package - 01/25/2015

<http://developer.android.com/training/index.html>

[15] PS2 Controller

Altera IP Core - PS2 Controller - 02/02/2015

[http://www.altera.com/education/univ/materials/comp\\_org/ip-cores/unv-ip-cores.html](http://www.altera.com/education/univ/materials/comp_org/ip-cores/unv-ip-cores.html) - scroll down to PS2 controller section

[16] NFC Library and Tutorial's

Libnfc - Public platform independent Near Field Communication (NFC) library - 02/09/2015

<http://nfc-tools.org/index.php?title=Libnfc>

[17] PN532 Setup and Use

Adafruit Setup and Tutorial's for NFC PN532 board - 01/15/2015

<https://learn.adafruit.com/adafruit-pn532-rfid-nfc>

[18] I2C Controller

Altera I2C Controller Core - 01/30/2015

[http://www.altera.com/support/refdesigns/sys-sol/indust\\_mil/ref-i2c.html](http://www.altera.com/support/refdesigns/sys-sol/indust_mil/ref-i2c.html)

[19] Average Power Consumption of Smartphones

How Much Energy a Smartphone Uses in a Year - 02/28/2015

<http://lifesaver.com/5948075/how-much-energy-a-smartphone-uses-in-a-year-and-what-it-means-for-your-budget>

[20] Ndef Tools For Android, Open Source Project

ndeftools-1.2.3 - 02/28/2015

<https://code.google.com/p/ndef-tools-for-android>

[21] Nexus 5 Safety & Warranty datasheet

RoHS compliance and safety information - 02/28/15

<http://www.lg.com/us/support-mobile/lg-LGD820>

[22] A new and simplified model for Automated Trust Negotiation.

An Asymmetrical Encryption Based Automated Trust Negotiation Model - 03/01/2015

<http://ieeexplore.ieee.org/xpls/icp.jsp?arnumber=4635156>

[23] Keyboard Make Codes

Keyboard Scan Codes: Set 2 - 02/02/2015

<http://www.computer-engineering.org/ps2keyboard/scancodes2.html>

[24] Embedded File System Library

All source code for EFSL - 03/01/2015

<https://www.openhub.net/p/efsl/commits/68269113>

[25] NFC Forum Type 2 Tag Operation Specification

Description of the NFC forum type 2 tag - 03/05/2015

[http://members.nfc-forum.org/specs/spec\\_list/](http://members.nfc-forum.org/specs/spec_list/)

# Appendix A - Quick Start Manual

## Setup

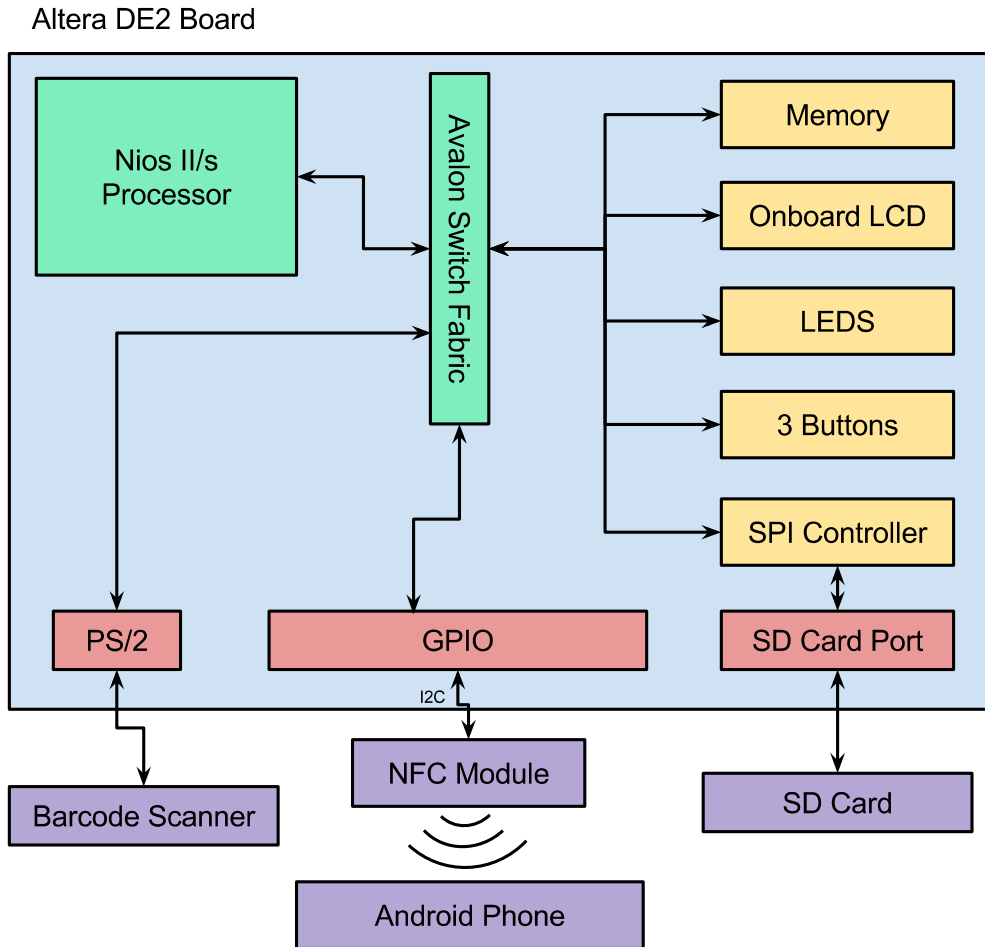
1. Download the Archived project from the course website
2. Open up Quartus and select Project > Restore Archived Project, and follow the steps to restore the project.
3. Connect all IO devices to the Altera Board including the Barcode Scanner, Keyboard, NFC Breakout Board, and SD card.
  - a. The file structure on the SD card needs to be copied from the SD\_File\_Structure.zip folder onto the SD card.
4. Power on the DE2 board.
5. Program the Altera DE2 with the .pof/.sof project files
6. Open the NIOS II Eclipse IDE and import the Software folder of the project
7. Generate the Board Support Package for the project by right clicking the project \_bsp folder > Nios II > Generate BSP.
8. Build the project by right clicking the project folder > Build Project.
9. Run the project by right clicking the project folder > Run As > Nios II Hardware.
10. Ensure that all components are initialized on the board, information will be provided on the LCD as well as the LED loading bar.
11. Unzip the android application “paperlessreceipts” and build it within Android development studios.
12. Run the application and select your chosen connected device for it to be programmed onto.

## Running

13. Button 1: Begins a receipt transaction
14. Once a transaction has begun, scan in items using the barcode scanner or use the keyboard for manual input.
15. Button 3: Can be used to remove the previously scanned in item.
16. Button 2: Ends the Receipt transaction.
17. Open the application on the android device and select “Receive Receipt.”
18. When prompted by the LCD screen, bring the android device within range of the PN532 Breakboard and hold the device there until the generated receipt is shown.



## Appendix B - Hardware Components



## Appendix C - Software

Interface	File Name	Description	Status
Android Application	MainActivity.java	Main screen code with switching activities for different task.	T
	Receipt.java	Code for generating the list of the receipts that are stored on the phone	T
	ReceiptView.java	Code for reading the selected receipt	T
	activity_main.xml	XML code for the visuals on the main screen	T
	activity_receipt.x	XML code for the displaying of a single selected	T

	ml	receipt	
	activity_receipt_view.xml	XML code for displaying the list of stored receipts	T
	activity_NFC_Transaction.java	The Code detects NFC fields and searches for NFC NDEF messages.	T
	activity_NFC_Transaction.java	XML Code for displaying information for the NFC transaction Page	T
PS/2 Controller	ps2_func.h	The header file for the ps2 driver	T
	ps2_func.c	The source file for the ps2 driver	T
NFC Controller	pn532.h	Header file for the modified pn532 driver	T
	pn532.c	Source file for the modified pn532 driver	T
	I2C.h	Header file for the I2C driver provided by Altera	T
	I2C.c	Source file for the I2C driver provided by Altera	T
Transaction Software	receipt_func.h	Header file for the receipt generation	T
	receipt_func.c	Source file for the receipt generation Compiled and tested but features will need to be added	T

## Appendix D - Future Work

The following items are enhancements that could be added to the project:

- Allow items to be added or removed to the systems database of scannable items
  - Approach 1: Use a web interface and connect to the DE2 board via Ethernet. Edit the items on the SD card via the web interface.
  - Approach 2: Create a new mode on the DE2 that allows for editing of the SD card's item database. This could be accomplished by implementing a switch on the board that changes the mode. Once in the correct mode, items can be iterated through using the onboard LCD and deleted using one of the buttons. When a user wants to add an item, the barcode scan be scanned and the name and price input using the keyboard.
- Add user accounts with contact information
  - Can be used to look up a specific person's receipts

- If user gets new device or it becomes stolen, receipts can be redownloaded to Android phone
- Users can be contacted (eg. if an item has been recalled by the manufacturer)
  - SMS, email, or device notifications
- Need to consider user privacy with encryption
- Add support for QR codes
  - would require new barcode scanner capable of scanning 2D barcodes
- Adding encryption to receipts so they cannot be modified by the store or created by customers
  - create cryptographic hash using a Secure Hash Algorithm (SHA), add has to bottom of receipt
  - The customer and store both have public-private keys using the RSA encryption standard
  - Encrypt hash on receipt located on the users phone and on the SD card using public-private keys
    - create functionality within system software to allow for receipt verification by comparing hash codes after private keys have been entered by store and customer