AN1949
APPLICATION NOTE

RFID Reader using STMicroelectronics µPSD MCU and CRX14 Coupler to Implement a China National ID Card Reader

The National ID Reader is a demonstration of a reader to read the information from a National ID card and to display the information on a PC. The demonstration is a reference design for the China National second generation ID card reader using a µPSD32xx or µPSD33xx 8051 core for the microcontroller, and the CRX14 as the contactless coupler. The demonstration communicates with the PC using the USB or RS232 interface. The demonstration complies fully with the ISO14443 type-B standards.

HARDWARE DESCRIPTION OF THE NATIONAL ID READER

The whole system can be divided into two main parts: one is the main control board; the other is the boosted antenna board. Figure 1. shows the whole system block diagram.

Figure 1. System Block Diagram

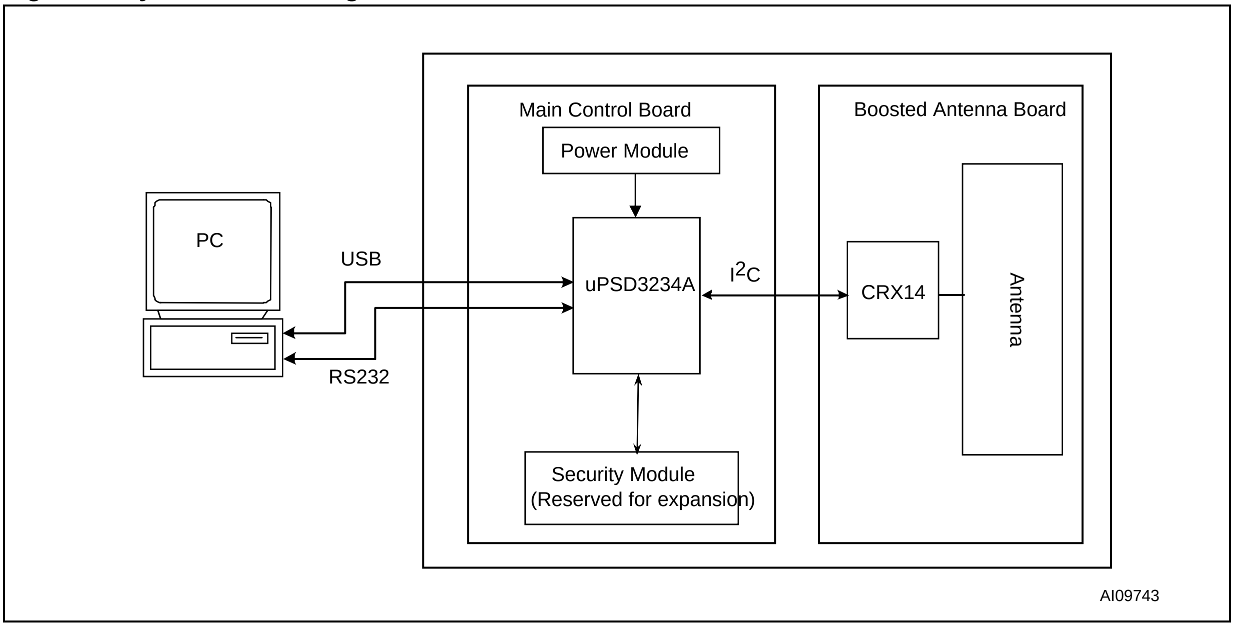


TABLE OF CONTENTS

Main Control Board

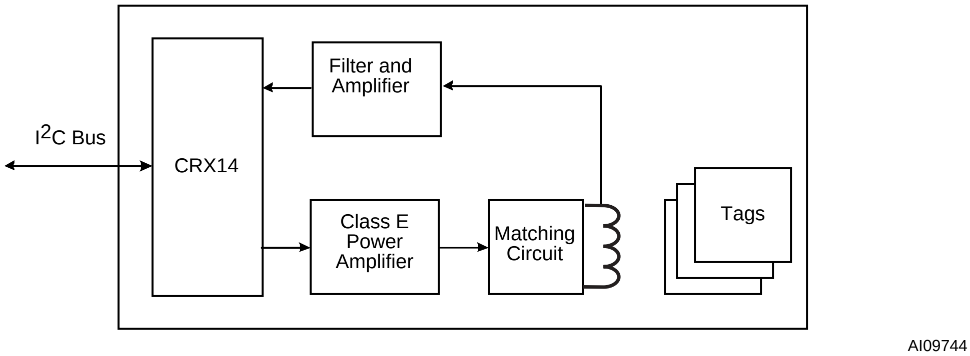
The main control board contains a uPSD3234A Control Unit, a security module and a power module.

- The uPSD3234A communicates with the PC using a USB/RS232 interface, and controls the card-reader chip's operation via the I^2C bus. If the security module is used, the uPSD3234A is also responsible for controlling and communicating with the security module.
- The security module on this demonstration is reserved for future expansion. If the security authentication mechanism is to be used, users can just simply put their security module into the reserved space on the board, and program their own firmware. The basic operation to the security module is fully compatible with ISO7816 (see the ISO7816 standard and the specific security module datasheet). The clock of the security module can be provided either from the uPSD3234A's PWM pin, or directly from an external crystal oscillator. For details, see Figure 10..
- The power module provides a stable 5V DC voltage supply to the system. The user should provide the system with a 9 to 12V DC voltage through the power connector. In this demonstration, for compatiblity with the µPSD33xx family, a 3.3V DC voltage is also provided.

Boosted Antenna Board

The range of the CRX14, to the tags, can be increased if the transmitted power is increased, and the receive signal is amplified. The boosted antenna circuit block diagram is shown in Figure 2..

Figure 2. Boosted Antenna Block Diagram



CRX14

CRX14 is a short range contactless coupler chip, compliant with the ISO14443 type B proximity standard. It generates a 13.56MHz signal modulated according to ISO14443 type B proximity standard. The CRX14 features the ST anti-collision mechanism, which allows the reader to detect and identify all the tags that are present in the operating range, and to access them individually. Because the CRX14 implements the France Telecom-proprietary anti-clone function, the reader can also perform authentication of tags that are equipped with the France Telecom anti-clone capability.

The CRX14 coupler interfaces between:

- the memory tags, on one side, through input/output buffers and the ISO14443 type B radio frequency protocol, and
- the system master processor, on the other side, through a 400kHz I^2C bus.

Operating from a 5V power supply, and delivered in a SO16N package, the CRX14 coupler chip is an excellent solution for building contactless readers, embedded in the final equipment, and offering a good compromise between operating range and cost.

Debugging Procedure

Some debugging might be needed, to make the boasted antenna circuit work properly.

Test the Output of the CRX14. The signal on the RF_{OUT} pin should be 10% ASK modulated on a carry frequency of 13.56MHz.

Test the Output of the Class E Amplifier. The signal after L7 (test point 2, on the schematic diagram) should be the amplified signal of RF_{OUT}, as shown in Figure 3.. If there is no signal, you should check the transistor and the passive components around it.

Optimize the Antenna. The Input Impedance of the antenna (Z_a), and its matching circuit, should match the output impedance of the Class E amplifier, see Figure 4.. Because of slight differences among PCB materials, or other problems, the features of the antenna may vary, such as the inductance (L_a) of the antenna, and the resistance (R_p). It is necessary to make some adjustment of the antenna matching circuit to achieve high performance.

Figure 3. Testing the Output of Class E Amplifier

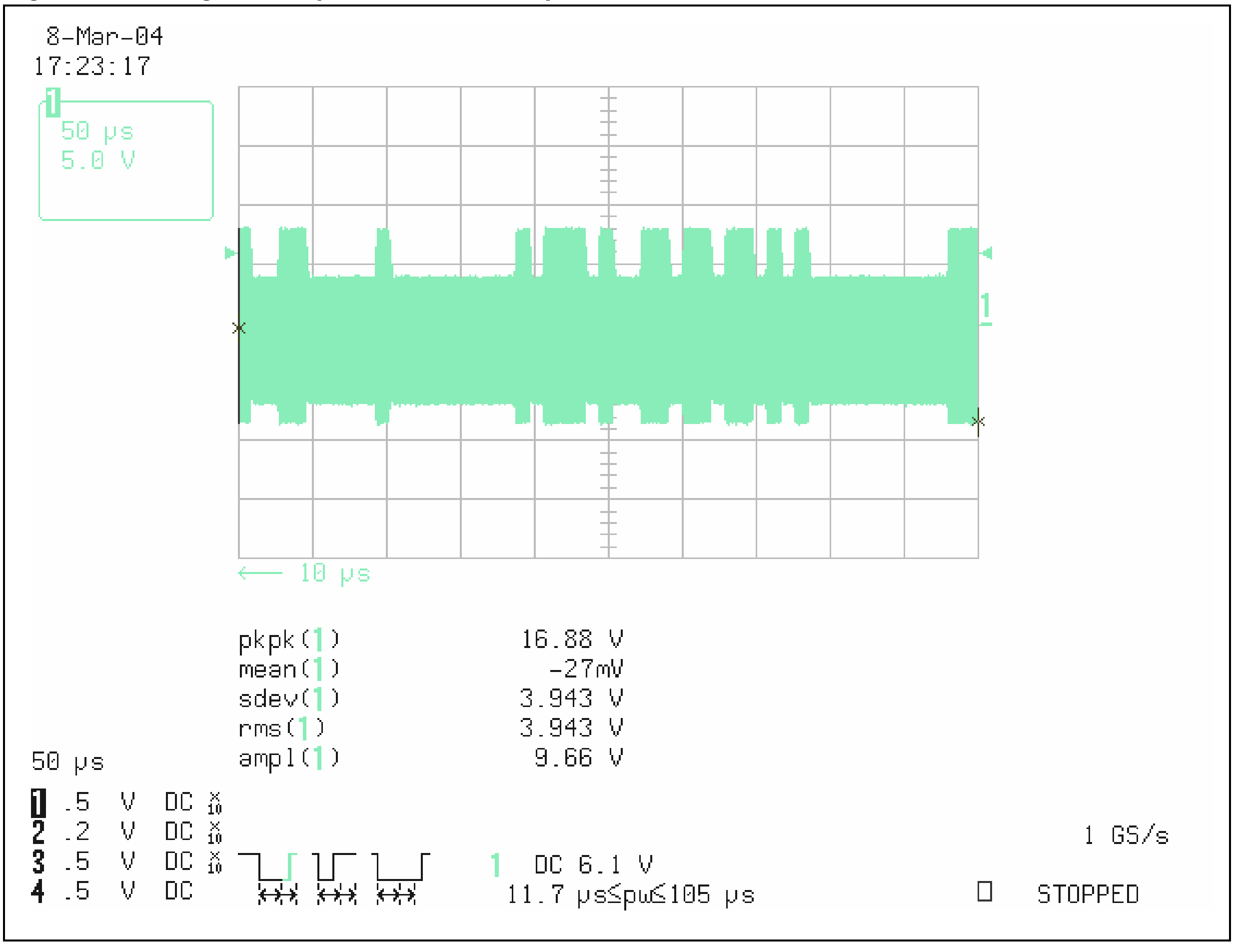
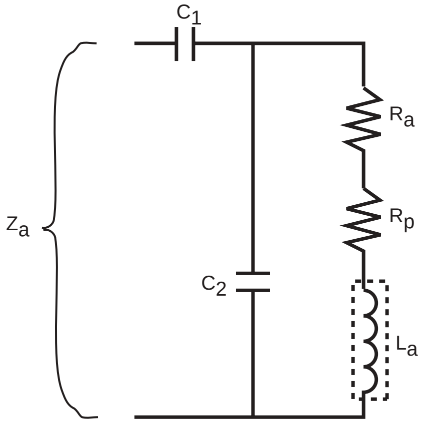


Figure 4. Circuit for the Antenna System on the Board



Since the difference will not be large, the adjustment to the antenna matching circuit is fairly simple, using an oscilloscope. Using two variable capacitors, substitute these for the C26+C27 and the C28+C31. Their adjustable range shall be around the original value of C26+C27 and C28+C31 (27pF and 125pF). Then carefully adjust the capacitors. When the voltage on the antenna loops reaches its maximum value, the correct value for the capacitor has been found.

For more details, see *AN1806: Antenna (and Associated Components) Matching-Circuit Calculation for the CRX14 Coupler.*

Check the Filter and Amplifier. The signal after the operational amplifier (point 3, on the schematic diagram) should be about 3.5V (peak-to-peak) as shown in Figure 5.. The signal on the RF_{IN} pin should be 4.6V (peak-to-peak), as shown in Figure 6..

Figure 5. Checking the Amplifier Output

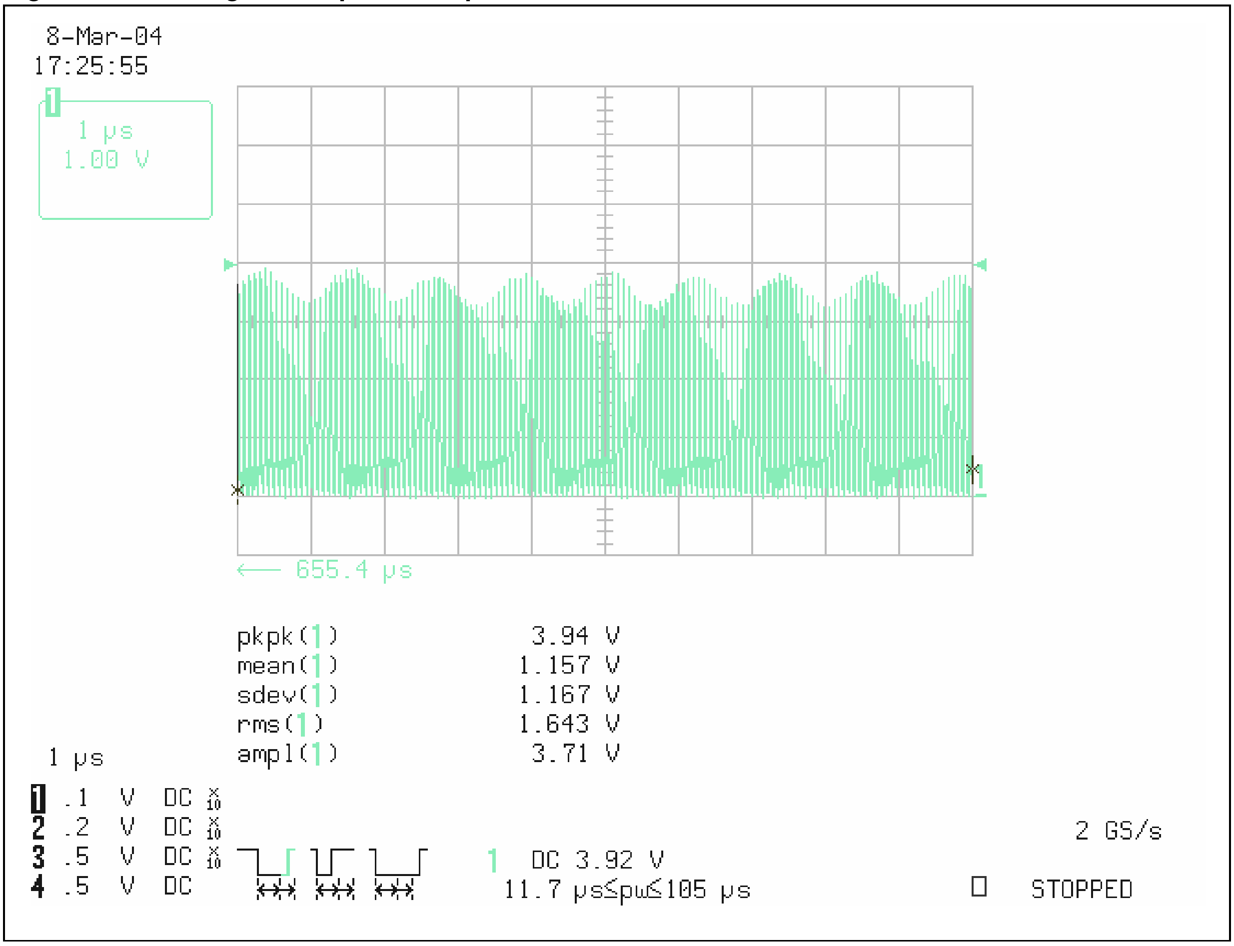
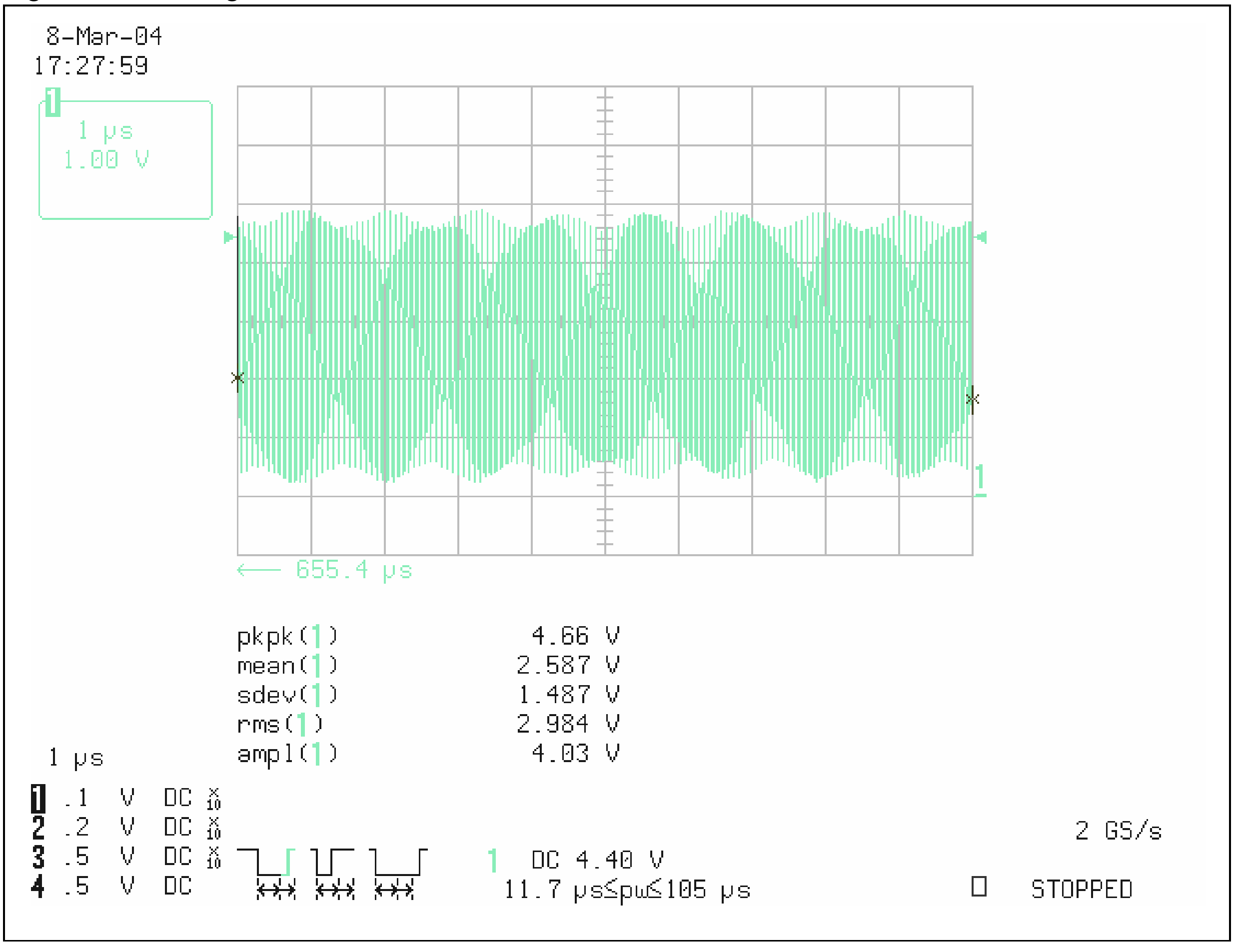
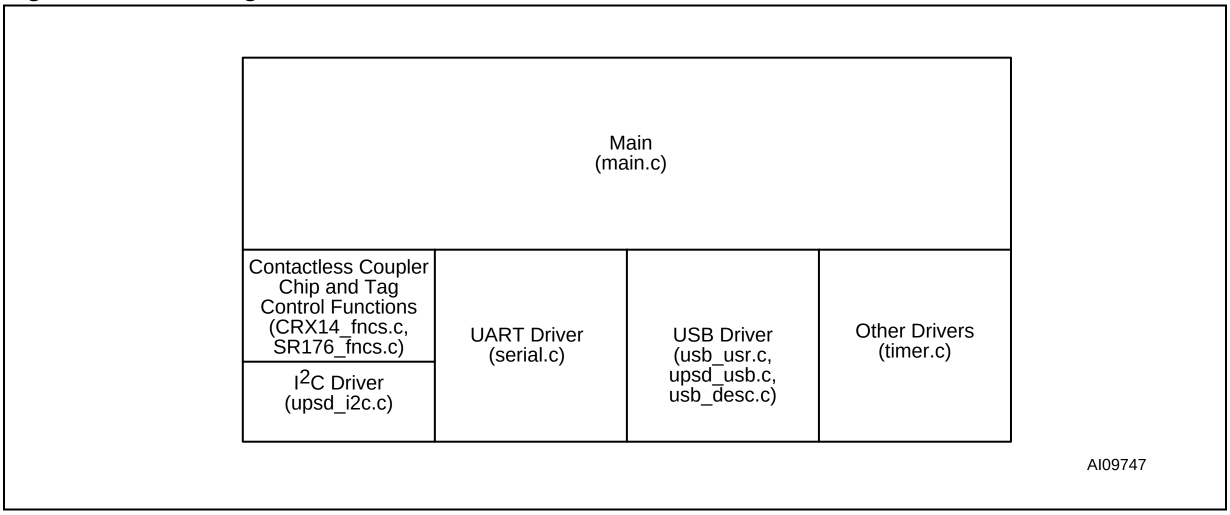


Figure 6. Checking RF_{IN}



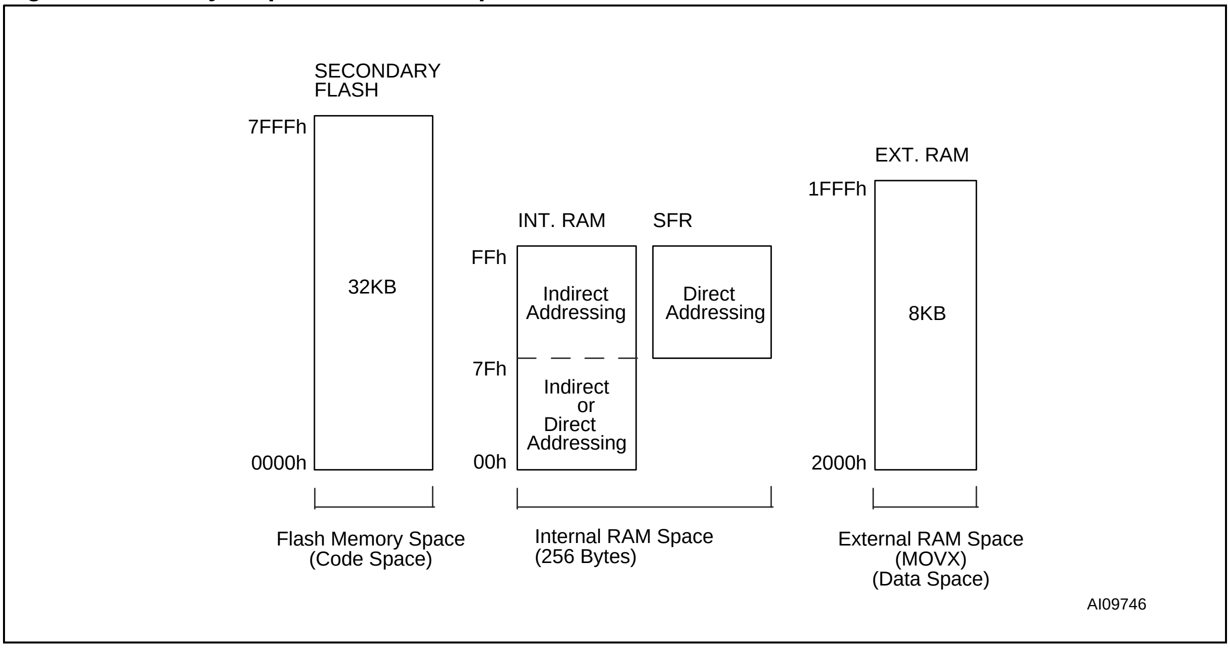
THE FIRMWARE

Figure 7. Block Diagram



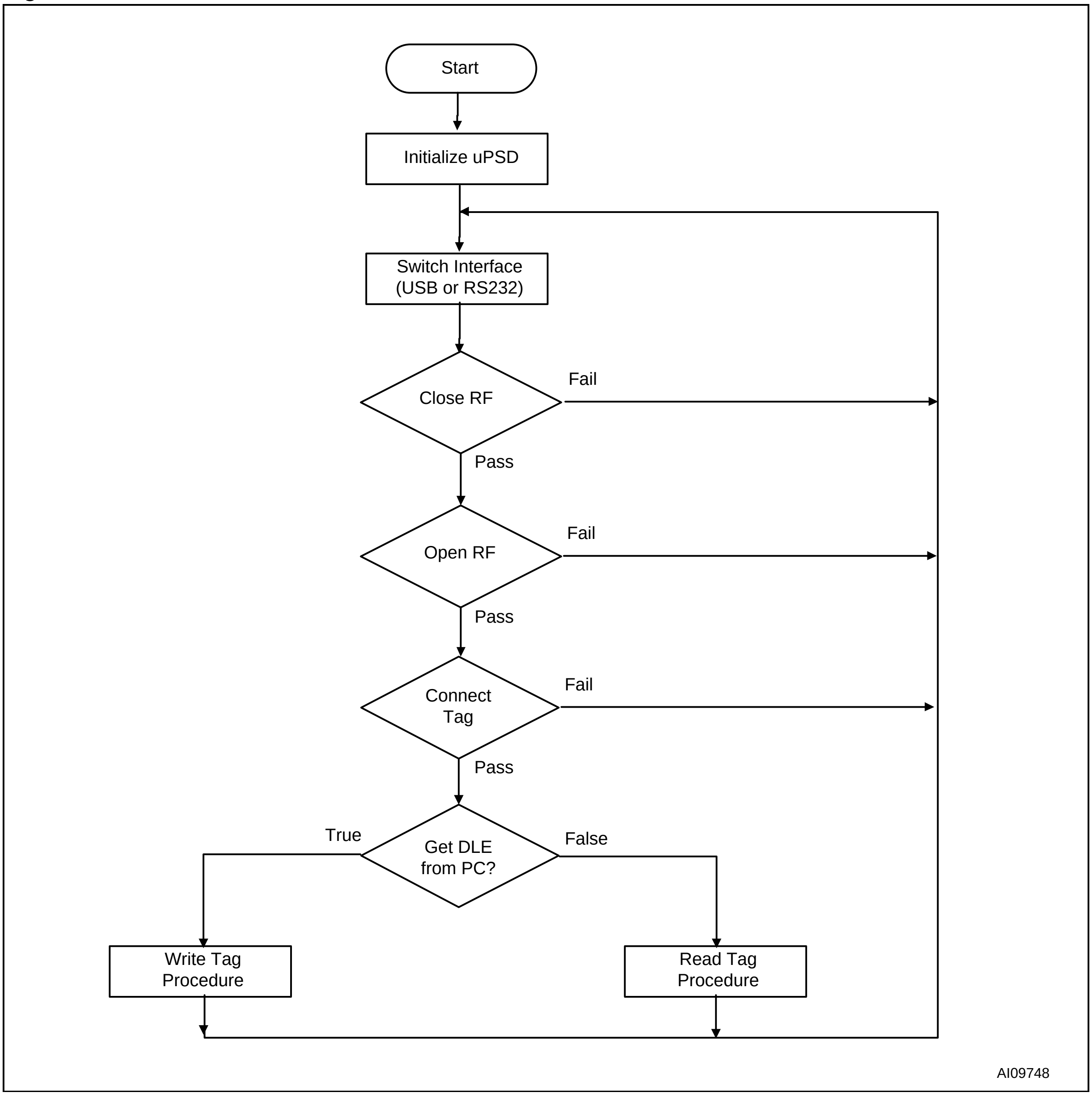
For the purposes of this demonstration, only the Secondary Flash memory of the µPSD has been used. In Application Programming (IAP) is possible if both main and secondary Flash memory of the µPSD are used.

Figure 8. Memory Map and Address Space



Firmware running on the µPSD will mainly control the operation of the CRX14, and communicate with the PC. Firmware will continuously control the CRX14 to send enquiry signals to the proximity field. If there is a National ID card within the field, the µPSD will command the CRX14 to read the information from the card, and to send the information back to the PC via the USB or RS232. When receiving a write command from the PC, the µPSD will command the CRX14 to write the information to the National ID card.

Figure 9. Firmware Flow Chart



Communication Protocol between PC and µPSD

<Need to add a few sentences to explain about this custom protocol, such as: is it same for RS-232 and USB? What is data transfer rate?>

Table 1. Read processing: Terminal to PC (Read Data Send) Data Frame Format

Command (20h)	Len (94h)	Data[0] to Data[m]	CRC16

Table 2. Write processing: PC to Terminal to PC (Write Data Send) Data Frame Format

Command (21h)	Len (94h)	Data[0] to Data[m]	CRC16

Table 3. Terminal to PC (Acknowledgement) Data Frame Format

Status (55h or AAh)

Note: 55h indicates that the write was successful; AAh indicates that there was a write error.

NATIONAL ID CARD DEMONSTRATION STRUCTURE

For this demonstration, the National ID card uses SRIX4K tag, from STMicroelectronics. It involves the storage of 147 bytes in 37 blocks, mapped in the way described next. In fact, the SRIX4K tag provides 121 blocks (484 bytes) to the user.

Block 7-9 (11 Bytes)

- ID number (BCD code): 9 bytes
- Gender and Nation: 1 byte
  - Gender: 1 indicates male; 0 indicates female.
  - Nation Code: an integer from 0 to 55, standing for 1 of 56 nations.
- Effective period (BCD code): 1 byte

Table 4.

Gender		Nation Code					
7	6	5	4	3	2	1	0

Block 10-13 (16 Bytes)

- Name (Unicode): at most 8 bytes (4 characters) (fixed)
- Birth Date (BCD code): 4 bytes
- Issue Date (BCD code): 4 bytes

Block 14-23 (40 Bytes)

- Issue Department (Unicode): at most 40 bytes (fixed)

Block 24-43 (80 Bytes)

- Address (Unicode): at most 80 bytes (40 characters) (fixed)

SUMMARY OF THE CRX14 CONTROL DRIVER

The CRX14 control driver consists of four files:

- **upsd_i2c.c**
- **upsd_i2cc.h**
- **CRX14_fncs.c**
- **CRX14_fncs.h**

The driver controls the CRX14's behavior via the I^2C interface..

Detailed Description of Key Routines

Explained below are the functions below that may be called by the main program.

```
int RF_ON(unsigned char *TX, unsigned char *RX)
```

This function turns on the CRX14's 13.56MHz carrier out.

– *TX pointer to the transmission buffer
– *RX pointer to the reception buffer
– Return value: 0 = success; not 0 = failure (to enable carrier out)

```
int RF_OFF(unsigned char *TX, unsigned char *RX)
```

This function turns off the CRX14's 13.56MHz carrier out.

– *TX pointer to the transmission buffer
– *RX pointer to the reception buffer
– Return value: 0 = success; not 0 = failure (to disable carrier out)

```
int Send_Receive_Data(unsigned char *TX, unsigned char *RX)
```

This function writes control bytes to the CRX14's Input/Output frame register, and receives the answer bytes, that are sent by the TypeB tag, from the Input/Output frame register.

– *TX pointer to the transmission buffer
– *RX pointer to the reception buffer
– Return value: 0 = success; not 0 = failure (to get data from the CRX14)

```
unsigned char upsd_i2c_init(unsigned char Slave_Addr, unsigned char* Data_Ptr, unsigned
char N)
```

This function initializes the I^2C line.

– Slave_Addr: Slave address of I^2C device
– Data_Ptr: address pointer to the transmission buffer
– N: length of data buffer to be transmitted
– Return value: not applicable

Porting Strategy

1. Include the four files in your application system
2. According to the the CRX14's I^2C address that you allocate in your hardware system, make the corresponding changes in *CRX14_fncs.h*. For example, if CRX14's I^2C address is 0xA2, then change

   #define CRX_ADDR 0xA0;

   into

   #define CRX_ADDR 0xA2;
3. In your application, set up the transmission and reception buffers for the CRX14 control driver. Each buffer should be at least 32 bytes in size.
4. According to the oscillator frequency of your hardware system, set the *Bus_Freq* parameter for the *upsd_i2c_init* function
5. Call the *upsd_i2c_init* function first, before you call other functions. Without a proper initialization process, the CRX14 may not work properly.

SUMMARY OF USB HID DRIVER

- Complies with the Universal Serial Bus specification Rev. 1.1
- Low speed (1.5Mbit/s) device capability
- Human Interface Device (HID) compatible

See *AN1886* for details.

Table 5. List of Header Files

Name	Description
APP_INTR.H	Header shared with Windows USB demonstration application, definition of message formats and USB demonstration command codes.
uPSD_USB.H	Short header with external definitions of uPSD_USB.c file
USB.H	The main header with USB descriptor and packet structures, USB and HID codes

Table 6. List of C Source Files

Name	Description
USB_USER.C	The main routine(s), USB ISR, command execution
uPSD_USB.C	The main USB driver
uPSD_CFG.C	Short USB configuration file, USB prescaler
uPSD_DEC.C	All USB descriptors and strings

Interface Switch

Firmware switches the interface between USB and RS232 automatically. If the demonstration is connected to a PC with a USB cable, the firmware chooses the USB interface, otherwise it chosen the RS232 interface.

APPENDIX A. SCHEMATIC DIAGRAMS

Figure 10. Main Control Board

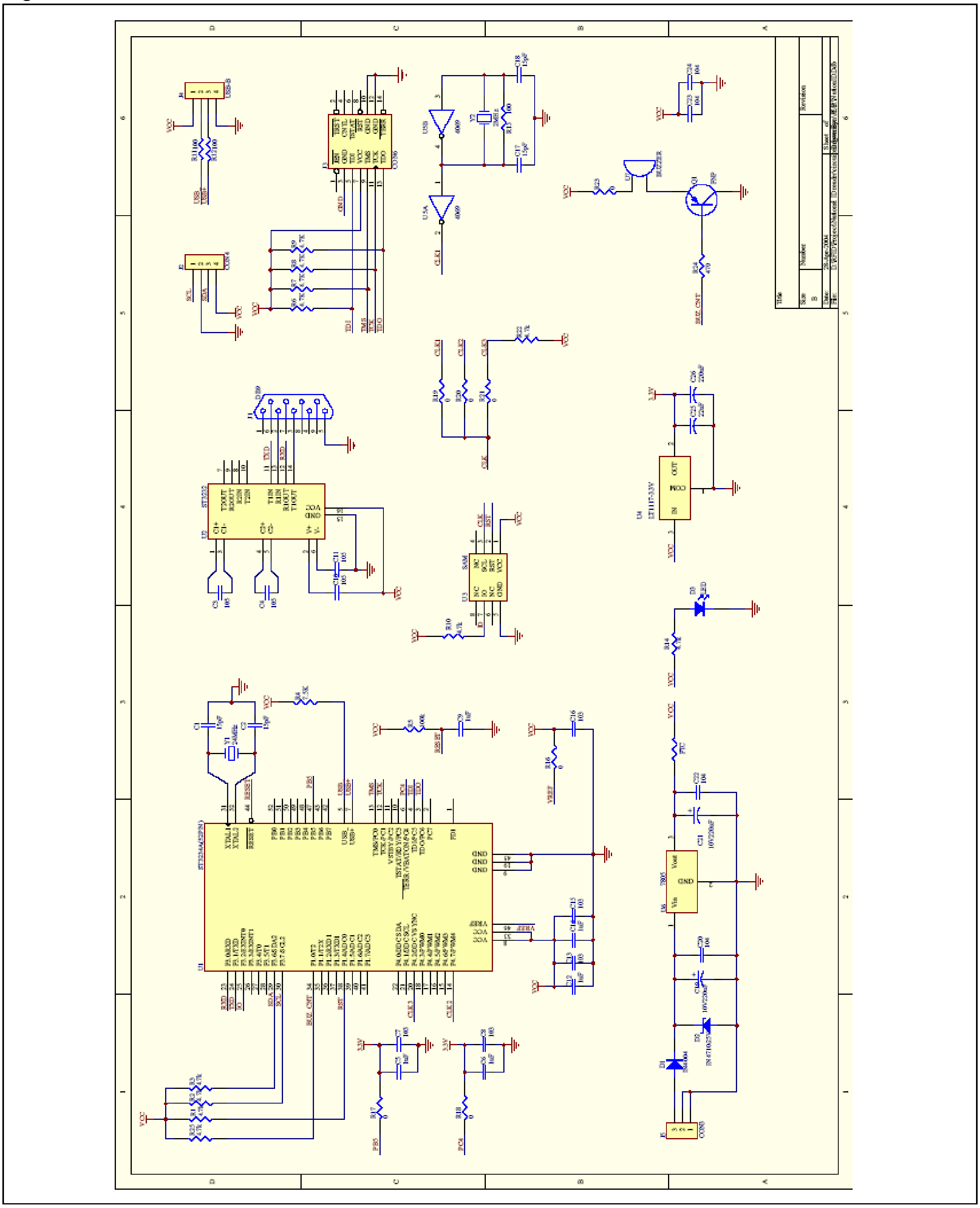
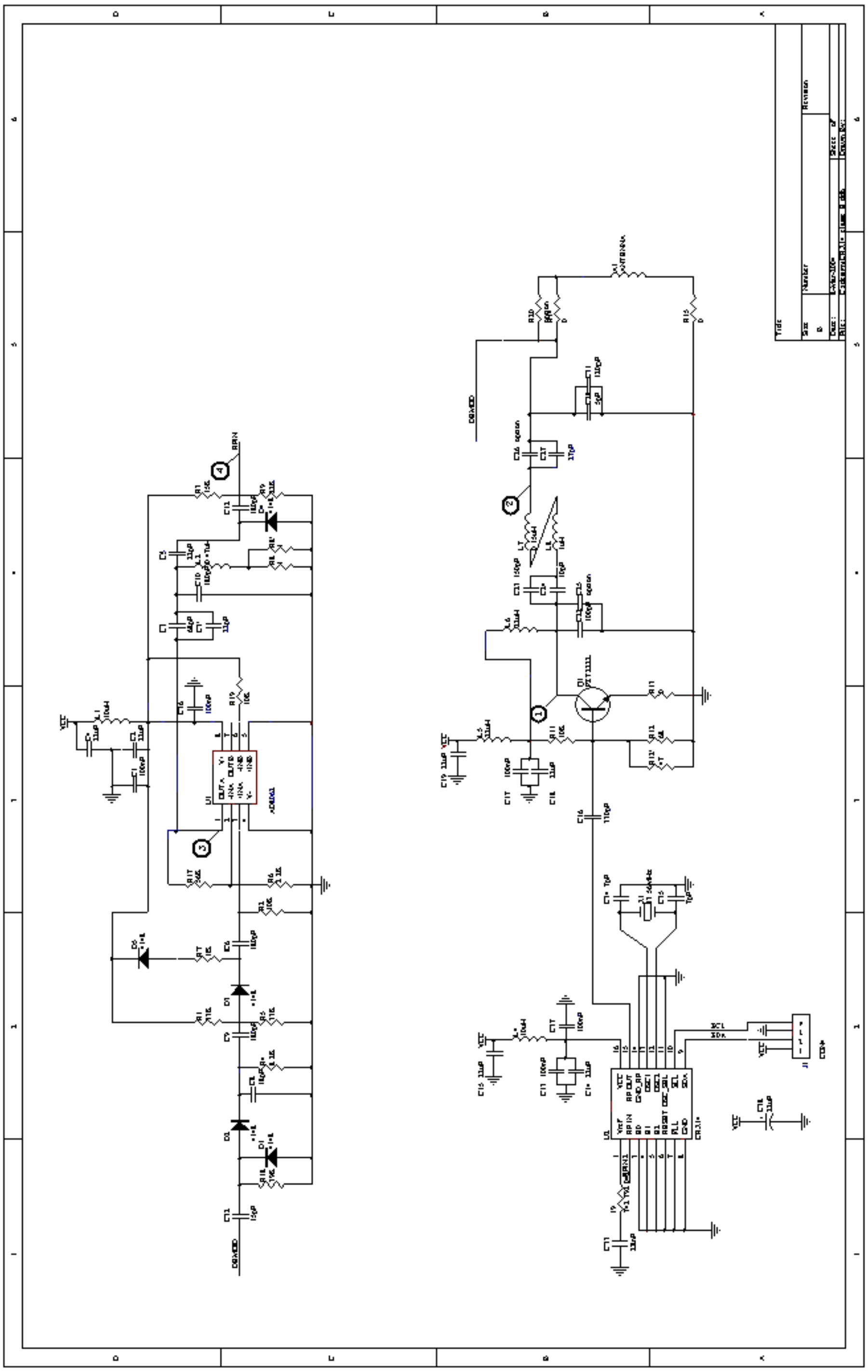


Figure 11. Boosted Antenna Board



APPENDIX B. PC SOFTWARE USER GUIDE

1. Run *setup.exe* and follow the instructions in the setup program (Figure 12. and Figure 13.).

Figure 12. Running *setup.exe*

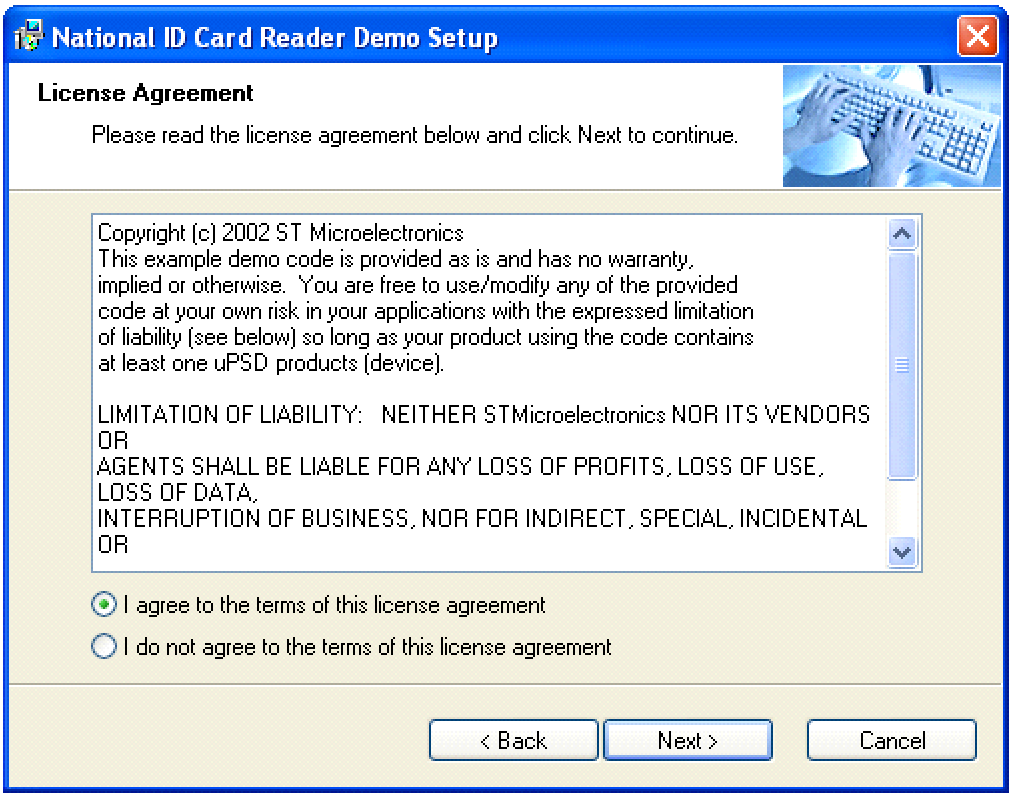
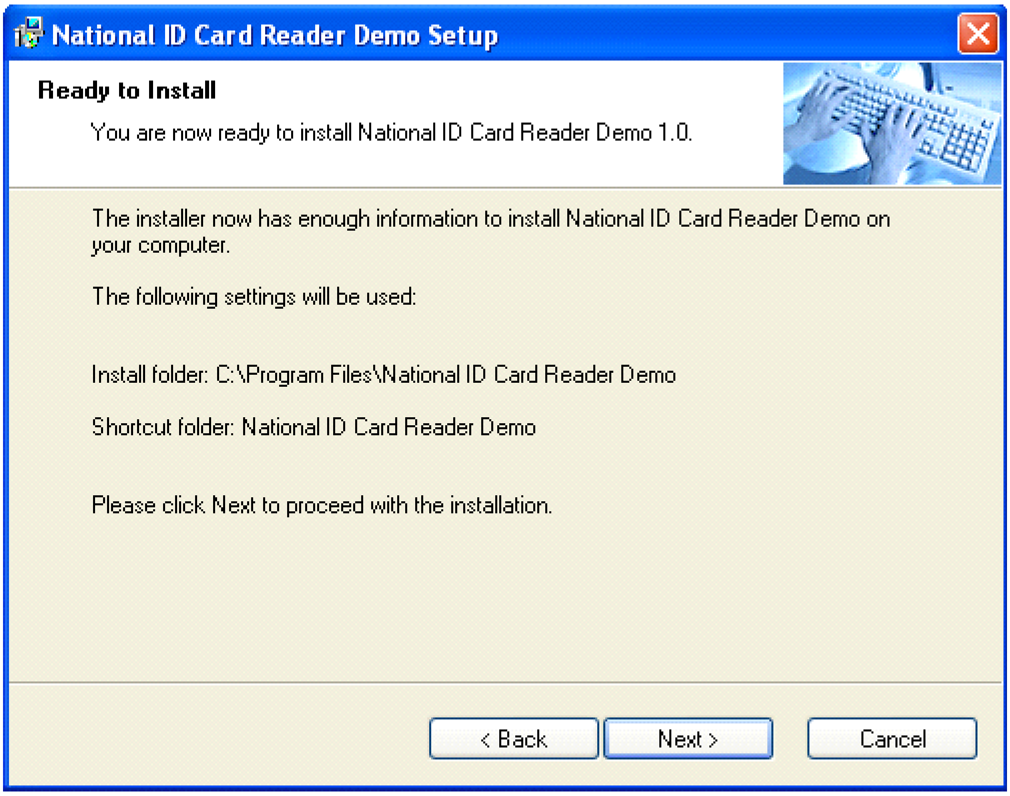
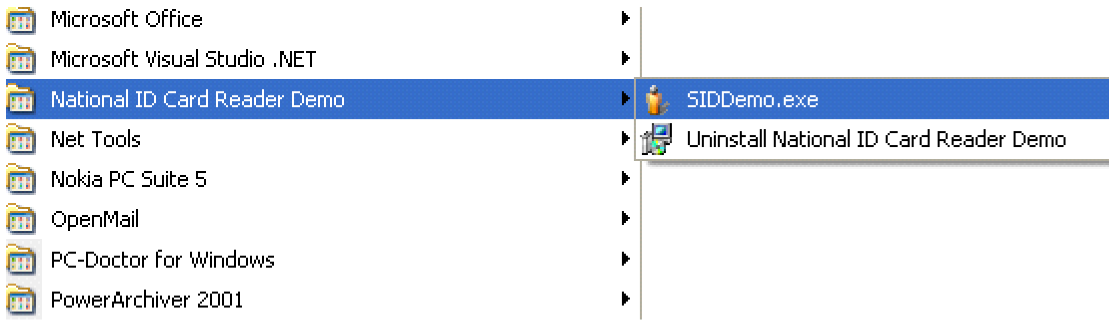


Figure 13. Running *setup.exe*



2. Run *SIDDemo.exe* (Figure 14.).

Figure 14. Running *SIDDemo.exe*



3. Select the appropriate connection, by clicking on its *connect* button (Figure 15.). (If the RS232 interface is selected, chose the correct com port: the one to which the RS232 cable is connected.)

Figure 15. Selecting the RS232 or USB Connection

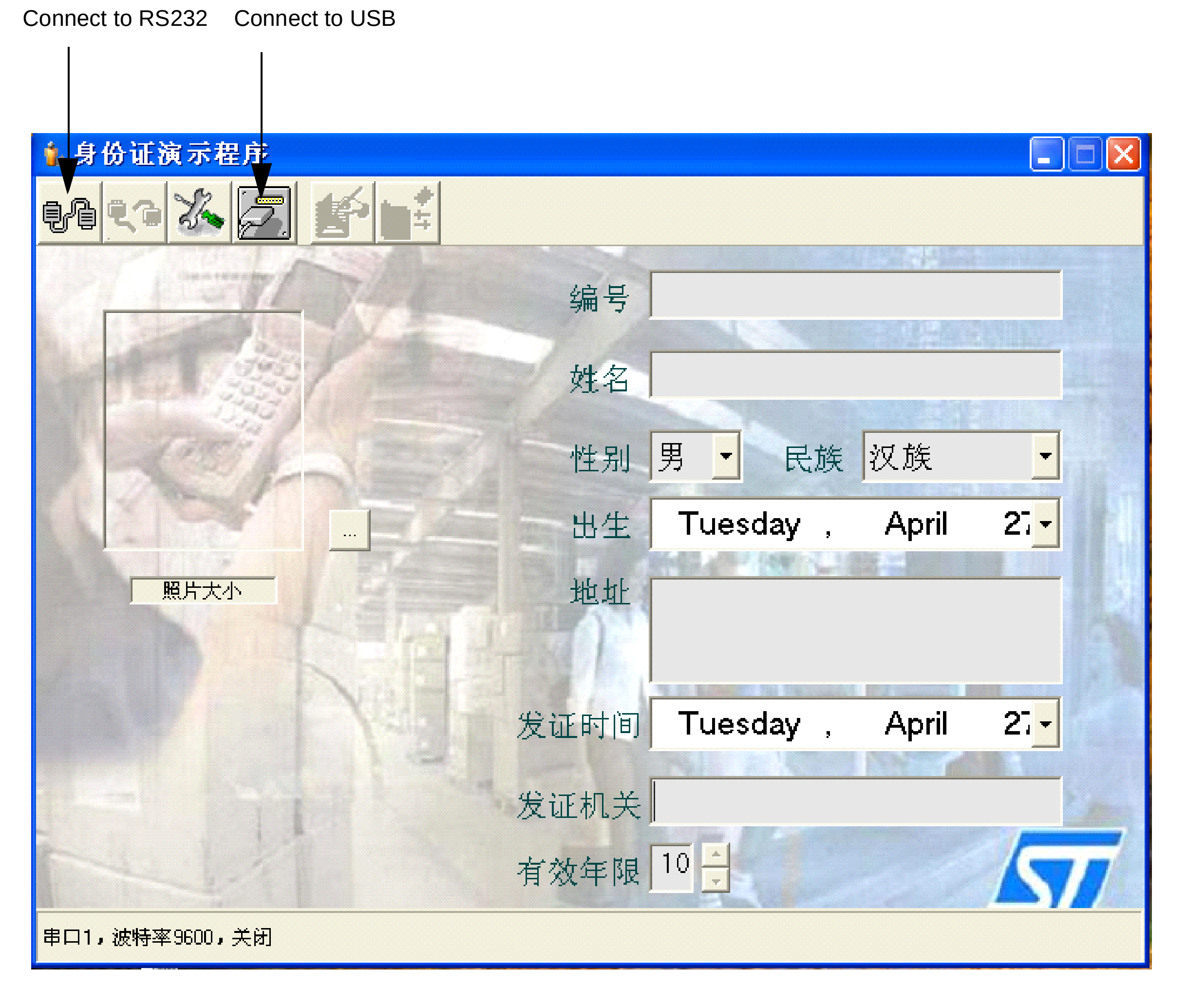
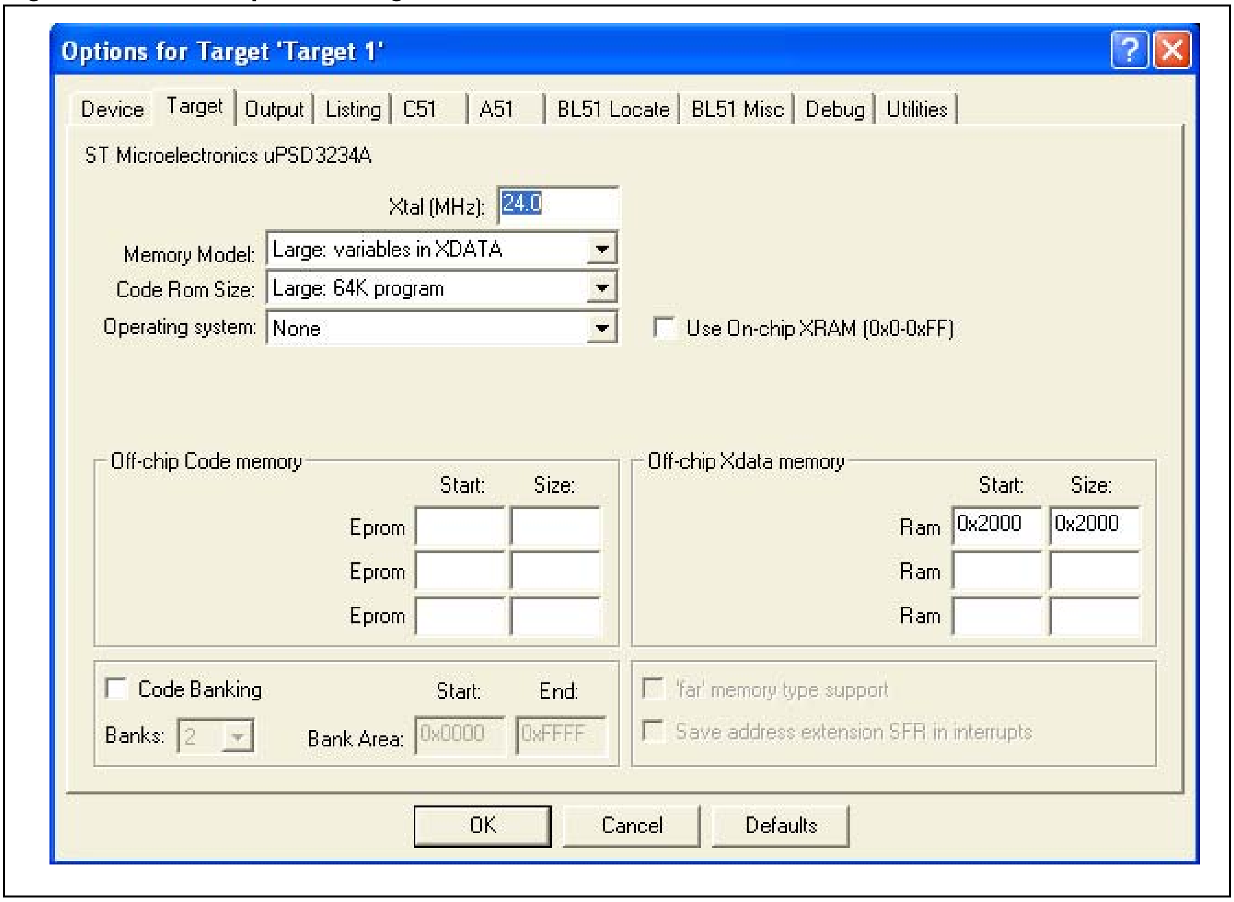


Figure 16. Keil Compiler Settings



APPENDIX C. PSDSOFT DESIGN SUMMARY FILE

<If the D-flip-flops (En_counter0, En_PA, En_PB, and EnDir0) are not used in this design, they should be removed from the PSDsoft project so that they will not create questions from customers.>

```
**********************************************************************
                         PSDsoft Express Version 8.00
                         Summary of Design Assistant
**********************************************************************
PROJECT    : rfid                           DATE : 04/19/2004
DEVICE     : uPSD3234A                      TIME : 12:38:27
MCU/DSP    : uPSD32XX
**********************************************************************

Initial setting for Program and Data Space:
===========================================

  Main PSD flash memory will reside in this space at power-up:      Data Space Only
  Secondary PSD flash memory will reside in this space at power-up: Program Space Only

Pin Definitions:
================

Pin          Signal                   Pin
Name         Name                     Type
------------ ------------------------ ------------
tdo          tdo                      Dedicated JTAG - TDO
tdi          tdi                      Dedicated JTAG - TDI
tck          tck                      Dedicated JTAG - TCK
tms          tms                      Dedicated JTAG - TMS
pd1          pd1                      GP I/O mode
USB-         USB_minus                USB- bus
USB+         USB_plus                 USB+ bus
_Reset_In    _Reset_In                Reset In
Vref         VREF                     VREF input
Xtal1        Xtal1                    Xtal1
Xtal2        Xtal2                    Xtal2

User defined nodes:
===================

Node         Node
Name         Type
------------ ------------
En_counter0  D-type register
En_PA        D-type register
En_PB        D-type register
En_Dir0      D-type register

Page Register settings:
=======================

pgr0 is not used
pgr1 is not used
pgr2 is not used
pgr3 is not used
pgr4 is not used
pgr5 is not used
pgr6 is not used
```

```
pgr7 is not used

Equations:
==========

rs0 = ((address >= ^h2000) & (address <= ^h3FFF));
csiop = ((address >= ^h0200) & (address <= ^h02FF));
csboot0 = ((address >= ^h0000) & (address <= ^h1FFF));
csboot1 = ((address >= ^h2000) & (address <= ^h2FFF));
csboot2 = ((address >= ^h3000) & (address <= ^h3FFF));
En_counter0.re = !_reset;
En_counter0.pr = Gnd;
En_PA := Vcc;
En_PA.ck = Vcc;
En_PB := Vcc;
En_PB.ck = Vcc;
En_Dir0.re = !_reset;
En_Dir0.pr = Gnd;
```

REVISION HISTORY

Table 7. Document Revision History

Date	Version	Revision Details
27-May-2004	1.0	First Issue