NFC Inventory Management System Project

Robert Dinh
Version 1
Tue Mar 3 2020

Table of Contents

Data Structure Index

Data Structures

Here are the data structures with brief descriptions:

File Index

File List

Here is a list of all documented files with brief descriptions:

Data Structure Documentation

PresenceData Struct Reference

Data Fields

- int **proximity**
  Proximity - Combination of upper byte and lower byte.

- int **luminosity**
  Luminoxity - Combination of upper byte and lower byte.

Detailed Description

Definition at line **14** of file **Presence.h**.

The documentation for this struct was generated from the following file:

- **Presence.h**

File Documentation

main.c File Reference

Main Program that will run the VNCL 4010, PN532 and LCD.

```
#include <stdio.h>
#include <stdlib.h>
#include <signal.h>
#include <string.h>
#include <unistd.h>
#include <nfc/nfc.h>
#include "Presence.h"
#include "NDEFData.h"
```

Functions

- int **main** (void)
  Function used to initialize Presence and setup the NFC reader.

Detailed Description

Main Program that will run the VNCL 4010, PN532 and LCD.

Author

Robert Dinh

Date

01MAR2020

Definition in file **main.c**.

Function Documentation

int main (void)

Function used to initialize Presence and setup the NFC reader.

The fuction sets up the VNCL4010 and PN532 using the Wiring Pi library through I2C. It then sets up the registers of the devices so that data can be retrieved. When a presence is detected by the VNCL4010 it triggers the NFC reader into a read state to to grab an NDEF encoded message from a mobile device.

Definition at line **27** of file **main.c**.

main.c

```
00001
00008 #include <stdio.h>
00009 #include <stdlib.h>
00010 #include <signal.h>
```

```
00011 #include <string.h>
00012 #include <unistd.h>
00013 #include <nfc/nfc.h>
00014 #include "Presence.h"
00015 #include "NDEFData.h"
00016
00017
00027 int main (void)
00028 {
00029     initNFC();
00030     int fd = initPresence();
00031
00032     int present = 0;
00033
00034     fprintf(stdout,"%d \n",fd);
00035     present = detectPresence(fd);
00036     if (present)
00037         getNdef();
00038
00039     stopFunction(2);
00040     return 0;
00041 }
```

NDEFData.c File Reference

Various Functions to read an NDEF message from a mobile device.

```
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <signal.h>
#include <nfc/nfc.h>
```

Functions

- void **stopFunction** (int sig)
  Function used to stop the NFC reader in the case of a System Interrupt.

- void **initNFC** (void)
  Function used to initialize the PN532 NFC Reader.

- int **CardTransmit** (nfc_device *pnd, uint8_t *capdu, size_t capdulen, uint8_t *rapdu, size_t *rapdulen)
  Function used to send APDU commands.

- void **getNdef** (void)
  Function used to initialize Presence.

Variables

- nfc_device * **pnd**
- nfc_context * **context**

Detailed Description

Various Functions to read an NDEF message from a mobile device.

Author

Robert Dinh

Date

01MAR2020

The following does contain non-original code but was modified to suit certain needs. Original code was developed by NFC - TOOLS. `http://nfc-tools.org/index.php/Libnfc:APDU_example`

Definition in file **NDEFData.c**.

Function Documentation

int CardTransmit (nfc_device * *pnd*, uint8_t * *capdu*, size_t *capdulen*, uint8_t * *rapdu*, size_t * *rapdulen*)

Function used to send APDU commands.

Definition at line **70** of file **NDEFData.c**.

void getNdef (void)

Function used to initialize Presence.

Definition at line **106** of file **NDEFData.c**.

void initNFC (void)

Function used to initialize the PN532 NFC Reader.

Definition at line **42** of file **NDEFData.c**.

void stopFunction (int *sig*)

Function used to stop the NFC reader in the case of a System Interrupt.

Definition at line **28** of file **NDEFData.c**.

NDEFData.c

```
00001
00013 #include <stdlib.h>
00014 #include <string.h>
00015 #include <unistd.h>
00016 #include <signal.h>
00017 #include <nfc/nfc.h>
00018
00019 //NFC Variables
00020  nfc_device *pnd;
00021  nfc_context *context;
00022
00023
00028 void stopFunction(int sig)
00029 {
00030
00031     printf("Stopping now.\n");
00032     nfc_close(pnd);
00033     nfc_exit(context);
00034     exit(EXIT_SUCCESS);
00035
00036 }
00037
00042 void initNFC(void)
00043 {
00044     signal(SIGINT,stopFunction);
00045     nfc_init(&context);
00046
00047     if (context == NULL) {
00048         printf("Unable to init libnfc (malloc)\n");
00049         exit(EXIT_FAILURE);
00050     }
00051
00052     pnd = nfc_open(context, NULL);
00053
00054     if (pnd == NULL) {
```

```
00055         printf("ERROR: %s", "Unable to open NFC device.");
00056         exit(EXIT_FAILURE);
00057     }
00058
00059     if (nfc_initiator_init(pnd) < 0) {
00060         nfc_perror(pnd, "nfc_initiator_init");
00061         exit(EXIT_FAILURE);
00062     }
00063
00064 }
00065
00070 int CardTransmit(nfc_device *pnd, uint8_t * capdu, size_t capdulen, uint8_t * rapdu,
size_t * rapdulen)
00071 {
00072     int res;
00073     size_t  szPos;
00074
00075     printf("=> ");
00076     for (szPos = 0; szPos < capdulen; szPos++)
00077     {
00078         printf("%02x ", capdu[szPos]);
00079     }
00080     printf("\n");
00081
00082     if ((res = nfc_initiator_transceive_bytes(pnd, capdu, capdulen, rapdu,
*rapdulen, 500)) < 0)
00083     {
00084         return -1;
00085     }
00086     else
00087     {
00088         *rapdulen = (size_t) res;
00089         printf("<= ");
00090
00091         for (szPos = 0; szPos < *rapdulen; szPos++)
00092         {
00093           printf("%02x ", rapdu[szPos]);
00094         }
00095
00096     printf("\n");
00097     return 0;
00098     }
00099 }
00100
00101
00106 void getNdef(void)
00107 {
00108     nfc_target nt;
00109
00110
00111     const nfc_modulation nmMifare = {
00112     .nmt = NMT_ISO14443A,
00113     .nbr = NBR_106,
00114     };
00115
00116     printf("Polling for target...\n");
00117
00118     while (nfc_initiator_select_passive_target(pnd, nmMifare, NULL, 0, &nt) <= 0);
00119     printf("Target detected!\n");
00120
00121     uint8 t capdu[264];
00122     size_t capdulen;
00123     uint8_t rapdu[264];
00124     size_t rapdulen;
00125
00126
00127     // Select application
00128     memcpy(capdu, "\x00\xA4\x04\x00\x07\xF0\x39\x41\x48\x14\x81\x00\x00", 13);
00129     capdulen=13;
00130     rapdulen=sizeof(rapdu);
00131     if (CardTransmit(pnd, capdu, capdulen, rapdu, &rapdulen) < 0)
00132         exit(EXIT FAILURE);
00133     if (rapdulen < 2 || rapdu[rapdulen-2] != 0x90 || rapdu[rapdulen-1] != 0x00)
00134         exit(EXIT_FAILURE);
00135
00136     printf("Application selected!\n");
00137
```

```
00138     // Select Capability Container
00139     memcpy(capdu, "\x00\xa4\x00\x0c\x02\xe1\x03", 7);
00140     capdulen=7;
00141     rapdulen=sizeof(rapdu);
00142     if (CardTransmit(pnd, capdu, capdulen, rapdu, &rapdulen) < 0)
00143         exit(EXIT_FAILURE);
00144     if (rapdulen < 2 || rapdu[rapdulen-2] != 0x90 || rapdu[rapdulen-1] != 0x00) {
00145         capdu[3]='\x00'; // Maybe an older Tag4 ?
00146     if (CardTransmit(pnd, capdu, capdulen, rapdu, &rapdulen) < 0)
00147         exit(EXIT_FAILURE);
00148     }
00149
00150     printf("Capability Container selected!\n");
00151
00152     // Read Capability Container
00153     memcpy(capdu, "\x00\xb0\x00\x00\x0f", 5);
00154     capdulen=5;
00155     rapdulen=sizeof(rapdu);
00156     if (CardTransmit(pnd, capdu, capdulen, rapdu, &rapdulen) < 0)
00157         exit(EXIT_FAILURE);
00158     if (rapdulen < 2 || rapdu[rapdulen-2] != 0x90 || rapdu[rapdulen-1] != 0x00)
00159         exit(EXIT_FAILURE);
00160
00161     printf("Capability Container header:\n");
00162
00163     size_t  szPos;
00164     for (szPos = 0; szPos < rapdulen-2; szPos++)
00165     {
00166         printf("%02x ", rapdu[szPos]);
00167     }
00168     printf("\n");
00169
00170     // NDEF SELECT
00171     memcpy(capdu, "\x00\xA4\x00\x0C\x02\xE1\x04", 7);
00172     capdulen=7;
00173     rapdulen=sizeof(rapdu);
00174     if (CardTransmit(pnd, capdu, capdulen, rapdu, &rapdulen) < 0)
00175         exit(EXIT_FAILURE);
00176     if (rapdulen < 2 || rapdu[rapdulen-2] != 0x90 || rapdu[rapdulen-1] != 0x00)
00177         exit(EXIT_FAILURE);
00178
00179     printf("NDEF SELECTED!\n");
00180
00181     // NDEF Read Binary
00182     memcpy(capdu, "\x00\xb0\x00\x00\x02", 5);
00183     capdulen=5;
00184     rapdulen=sizeof(rapdu);
00185     if (CardTransmit(pnd, capdu, capdulen, rapdu, &rapdulen) < 0)
00186         exit(EXIT_FAILURE);
00187     if (rapdulen < 2 || rapdu[rapdulen-2] != 0x90 || rapdu[rapdulen-1] != 0x00)
00188         exit(EXIT_FAILURE);
00189     printf("NDEF Read Binary NLEN!\n");
00190
00191     //READING NDEF DATA
00192     memcpy(capdu, "\x00\xb0\x00\x00\x0f", 5);
00193     capdulen=5;
00194     rapdulen=sizeof(rapdu);
00195     if (CardTransmit(pnd, capdu, capdulen, rapdu, &rapdulen) < 0)
00196         exit(EXIT_FAILURE);
00197     if (rapdulen < 2 || rapdu[rapdulen-2] != 0x90 || rapdu[rapdulen-1] != 0x00)
00198         exit(EXIT FAILURE);
00199
00200     printf("NDEF DATA ! \n\n");
00201     size_t  szPos2;
00202     char ndefMsg[100] = "";
00203     char chr[1] = "\0";
00204     int counter = 0;
00205
00206     for (szPos2 = 8; szPos2 < rapdulen-2; szPos2++) {
00207         ndefMsg[counter]=(char)rapdu[szPos2];
00208         counter++;
00209     }
00210
00211     printf("%s",ndefMsg);
00212
00213     char cmdS[100] = "python lcd.py Phone Scanned\n";
00214
```

```
00215
00216     system(cmdS);
00217     sleep(1);
00218     printf("\nDone...\n");
00219 }
```

NDEFData.h File Reference

Function prototypes for NDEFData.

Detailed Description

Function prototypes for NDEFData.

Author

Robert Dinh

Date

01MAR2020

Definition in file **NDEFData.h**.

NDEFData.h

```
00001
00008 #ifndef NDEFData_H
00009 #define NDEFData_H
00010
00011
00013 // Function Prototypes
00014 void stopFunction(int sig);
00015 void initNFC(void);
00016 int CardTransmit(nfc_device *pnd, uint8_t * capdu, size_t capdulen, uint8_t * rapdu,
size_t * rapdulen);
00017 void getNdef(void);
00019
00020
00021
00022 #endif
```

Presence.c File Reference

Various functions to aid in Presence Detection with the VNCL4010.

```
#include <stdio.h>
#include <wiringPiI2C.h>
#include <unistd.h>
#include "Presence.h"
```

Functions

- int **initPresence** (void)
  Function used to initialize Presence.

- int **detectPresence** (int fd)
  Function used to detect the presence of a person.

- int **getProximity** (int fd)
  Function used get the proximity of the sensor.

- int **getLuminosity** (int fd)
  Function used get the luminosity of the sensor.

Detailed Description

Various functions to aid in Presence Detection with the VNCL4010.

Author

Robert Dinh

Date

01MAR2020

Definition in file **Presence.c**.

Function Documentation

int detectPresence (int *fd*)

Function used to detect the presence of a person.

Definition at line **44** of file **Presence.c**.

int getLuminosity (int *fd*)

Function used get the luminosity of the sensor.

Definition at line **80** of file **Presence.c**.

int getProximity (int *fd*)

Function used get the proximity of the sensor.

Definition at line **65** of file **Presence.c**.

int initPresence (void)

Function used to initialize Presence.

The fuction sets up the VNCL4010 using the Wiring Pi library through I2C. It then sets up the registers of the device so that data can be retrieved.

Definition at line **20** of file **Presence.c**.

Presence.c

```
00001
00008 #include <stdio.h>
00009 #include <wiringPiI2C.h>
00010 #include <unistd.h>
00011 #include "Presence.h"
00012
00013
00020 int initPresence(void)
00021 {
00022     //Setup Wiring Pi thorugh I2C
00023     int fd = wiringPiI2CSetup(0x13);
00024
00025     //Select Command Register
00026     wiringPiI2CWriteReg8(fd,0x80,0xFF);
00027
00028     //Select Prox Register
00029     wiringPiI2CWriteReg8(fd,0x82,0x03);
00030
00031     //Select Current for IR LEDto 200ma
00032     wiringPiI2CWriteReg8(fd,0x83,0x14);
00033
00034     //Select ALS Register
00035     wiringPiI2CWriteReg8(fd,0x84,0x9D);
00036
00037     return fd;
00038 }
00039
00044 int detectPresence(int fd)
00045 {
00046         presenceD pData  = {0};
00047         while (1)
00048         {
00049             pData.proximity = getProximity(fd);
00050             pData.luminosity = getLuminosity(fd);
00051
00052             fprintf(stdout,"Proximity:\t%d\t
Luminosity:\t%d\n",pData.proximity,pData.luminosity);
00053             if (pData.proximity > 2300)
00054                 return pData.proximity;
00055
00056             usleep(300000);
00057         }
00058
00059 }
00060
00065 int getProximity(int fd)
00066 {
```

```
00067     int prox1= 0;
00068     int prox2= 0;
00069
00070     prox1 = wiringPiI2CReadReg8(fd,0x87) *256; //Upper Byte
00071     prox2 = wiringPiI2CReadReg8(fd,0x88); //Lower Byte
00072
00073     return prox1+prox2;
00074 }
00075
00080 int getLuminosity(int fd)
00081 {
00082     int lux1= 0;
00083     int lux2= 0;
00084
00085     lux1 = wiringPiI2CReadReg8(fd,0x85) *256; //Upper Byte
00086     lux2 = wiringPiI2CReadReg8(fd,0x86); //Lower Byte
00087
00088     return lux1 + lux2;
00089 }
```

Presence.h File Reference

Constants, structures, function prototypes for Presence.
`#include <wiringPiI2C.h>`

Data Structures

- struct **PresenceData**

Typedefs

- typedef struct **PresenceData presenceD**

Detailed Description

Constants, structures, function prototypes for Presence.

Author

Robert Dinh

Date

01MAR2020

Definition in file **Presence.h**.

Presence.h

```
00001
00008 #ifndef PRESENCE_H
00009 #define PRESENCE_H
00010
00011 #include <wiringPiI2C.h>
00012
00013 // Structures
00014 typedef struct PresenceData
00015 {
00016         int proximity;
00017         int luminosity;
00018 }presenceD;
00019
00021 // Function Prototypes
00022 int initPresence(void);
00023 int detectPresence(int fd);
00024 int getProximity(int fd);
00025 int getLuminosity(int fd);
00027
00028
00029
00030 #endif
```

Index

INDEX