• **Hagenberg** • Linz • Steyr • Wels

Technical Report
NFC Test Suite

Dominik Krenner

Hagenberg, March 20, 2008

Version:	0.0.1
Status:	Draft
Created:	Dominik Krenner
Datei:	TechReport.pdf

Contents

List of Tables

List of Figures

1 Abstract

The implementation of a Near Field Communication (*NFC*) test system had the goal to invent a system which is capable of testing *NFC* devices concerning the amplitude and frequency of the emitted electro magnetic field, just like the range in which other *NFC* devices can be detected. Therefore a cartesian coordinate robot was constructed. It can move the device under test (DUT) in three dimensions to a maximum of 200 mm along each axis. The three stepper motors used, provide an accuracy of 0.25 mm. The main application which was developed features a 3D log viewer, a robot control panel and a test editor. Tests scripts can be written by the user to provide custom test scenarios. These scripts are written in Python and can access the robot, logs and the DUT using a Python module provided with the main application.

To test a device, custom software has to be installed on it, which provides communication with the main application and controls the device's *NFC* activities as requested by the main application and the test script. This test-client, a J2ME application, comes with an hardware abstraction layer, allowing us to use the test client like the *Nokia 6131 NFC*, *Samsung SGH-X700N* and the *Sagem's my700X*.

The *NFC* devices can be connected via TCP/IP or *Bluetooth*. Already implemented tests include field measurements in all three dimensions. Window tests can be used to verify if a specific device can detect a target in a certain range. It is also possible to determine a devices maximum range in each direction. Measured positions are stored separately in a database and can be analyzed in the log viewer. The result can be rotated and zoomed and provides all collected data of each point.

Automated testing of *NFC* devices and their operating modes can be done by this system in a very convenient and quick way. Through customized test cases allow individual tests with different *NFC* devices without Software adaptation. The results of the tests indicate the functionality of the *NFC* devices in different operating modes.

2 Introduction

The project *NFC Testing* was a students project at the university of applied sciences in Hagenberg (Austria). The goal was to develop a system which can test *NFC* devices automatically. Five *Hardware/Software Systems Engeneering* students developed the systems as part of their studies over a period of two semesters.

2.1 *NFC*- The Basics

Near Field Communication (*NFC*) is a new technology which allows devices to communicate wirelessly over a short distance. It is possible to use one active and one passive device or two active devices. *NFC* devices operate at a frequency of 13.56 MHz and are able to interact with RFID devices. Because *NFC* is based on the ISO 14443A standart, it can be used with other existing smart card infrastructures like *Mifare* (*NXP Semiconductors*) or *FeliCa* (*Sony*). *NFC* provides a quantity of possible uses, it can replace a conventional key, can be used as a virtual purse or allows the user to establish a *Bluetooth* connection without the overhead of entering passkeys or the like. Another usage is to place passive *NFC* devices (so called *Tags*) behind a poster or advertisement and provide additional information like a short trailer to users of *NFC* devices, which can be accessed by placing a *NFC* capable mobile phone near the poster.

2.2 What will be tested?

The developed test system is able to position a *NFC* device using a cartesian coordinate robot. An oscilloscope provides access to amplitude and frequency measurement. Active *NFC* devices can be controlled via a software protocol and provide information if other *NFC* devices can be detected or not.

3 Concepts and realization

3.1 Overview

The System consists of six main components; The *NFC* devices, the mobile server, the robot, an oscilloscope, a database server and a workstation running the main application.

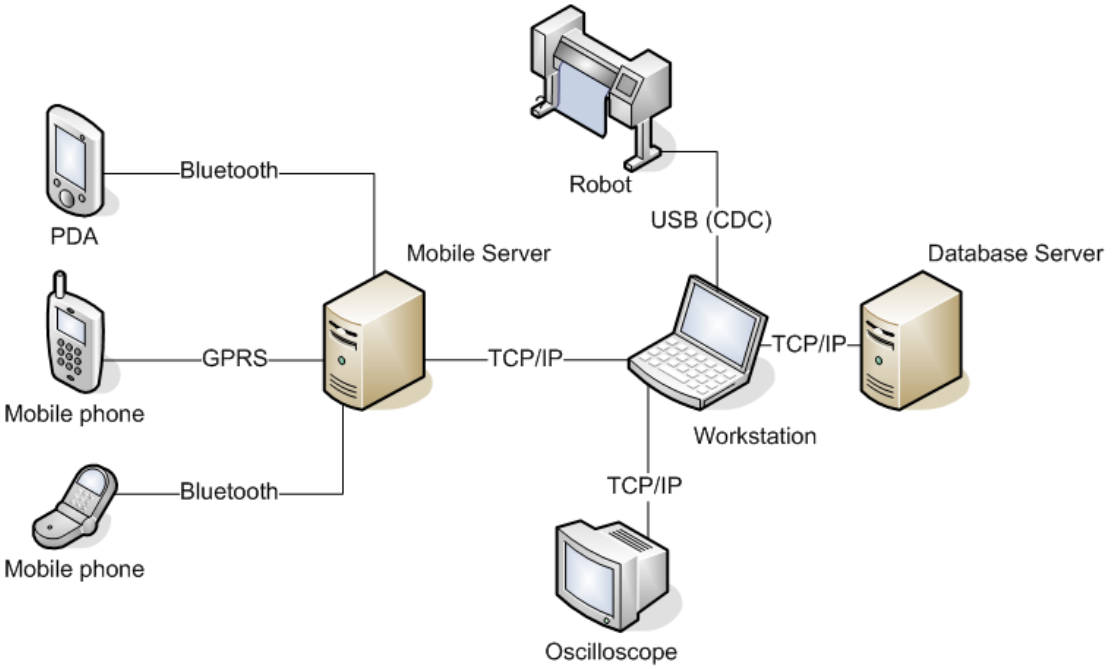

Figure 1: The test system

3.2 Why this concept?

To test a *NFC* device, the device has to be moved through space and communication between the main application and the device has to be possible. Because the distance between two *NFC* devices over which communication takes place is around 10 cm, the positioning has to be very precise to get accurate results. Therefore the robot moves using three stepper motors providing a 0.25 mm resolution each. Each device has different physical measurements, so the robot has an adaptable bracket to match the device-under-test's physical appearance. The main application has to communicate with all kind of *NFC* devices, each using different types of connectivity. To abstract the connection type, a proxy application, the mobile server, is used. Because test cases should be easy to create and modify, there has to be a mechanism which allows the operator to configure new tests or change existing ones, which is achieved via scripting. To measure amplitude and frequency of the created electro magnetic field, an oscilloscope with an attached antenna is being used.

3.3 Software

The main application coordinates attached devices as the robot, oscilloscope or the device-under-test. It provides test scripts with initialized and configured objects which represent the testing environment, like the robot, the oscilloscope and the device under test. It also provides a graphical user interface (GUI), a test-script editor, a robot control panel and a log viewer which presents collected data in 3D. It was developed using the object oriented scripting language *Python*. *Python* offers lots of features and packages which allow the easy creation of complex and powerful software, while still being nearly independent of the operating system. It also provides mechanisms to run scripts (also written in *Python*) and pass objects, which already exist in the main application, to them. These scripts can be created and modified while the main application is already running. The graphical user interface (GUI) was developed using the wxPython package which is a port of the wxWidgets class library. The main application also provides predefined classes which can be used by test scripts for fast and easy test case generation.

3.4 Devices

The first devices which were tested were the *Nokia 6131 NFC*, the *Samsung SGH-X700N* mobile phones and the *Dell Axim X51v* PDA. Each of the three devices uses a different connection method and runs a software providing the proprietary communication protocol between the mobile server and the device. This software also controls the *NFC* activities of the device as requested by the main application and the current test. Currently this application has to be customized to fit the devices architecture.

3.5 Robot

The cartesian coordinate robot was constructed out of three linear modules and three stepper motors, which allow the robot to move the device under test through space in three dimensions. It can move a device 20 cm in each direction, with an accuracy of 0.25 mm. A circuit board featuring a *ATmegaUSB* microcontroller controlls these motors and is connected to a working station via the USB protocol. To communicate with the host, it uses the USB communication device class and can be accessed by software like a serial communication port. A proprietary text based protocol is used to control the robot via software. The robot can recalibrate itself using six press switches at the end of each linear module which also work as emergency stops. The robot stores the maximum distance along each axis and its current position. This information is used by the robot to determine if a move-command can be executed safely or not and avoids that the robot tries to move the device under test too far and damage the devices.

3.6 Oscilloscope

To enable frequency and amplitude measurements an oscilloscope with an antenna attached is used. It is configured via the main application or the test case to provide flexibility. The connection between the main application and the oscilloscope is established over TCP/IP and a C library provided by the manufacturer.

3.7 Mobile server

The mobile server abstracts the connection method of the *NFC* devices and operates as proxy between the main application and the devices. The main application only has to use a single TCP/IP connection to interact with the devices. If a request from the main application to a device is sent, the mobile server makes sure that the request is sent to the correct device.

3.8 Database

The database is used to store test scripts and test results alike. Each position at which data is collected is stored with the corresponding results. At the moment of writing the software uses a MySQL database server running on the same machine as the main application. But because the connection is established via TCP/IP, the database server can be run on any machine in the network.

4 Testing

There are some basic test cases already implemented in the main application which can be ran to analyze a device under test in a two dimensional plane, a single line or the whole field in which measurements can take place. In addition, tests defined by the *NFC Forum* are available. To create own test cases, a simple interface is provided which allows the implementation of custom tests by using a object oriented environment.

4.1 Example test script: NFC Forum test TP_2_1_1

4.1.1 Description

This simple test verifies if a device is able to detect a target (*Tag*) at different distances between 0 to 10 mm. If the device can detect the target in all ranges, it passes the test. The start point is

```
1  # Entry function of test script
2  def Run(TE):
3          # Default result: device failed the test.
4          TE.Result = "Failed"
5          # Start with the device/tag touching each other
6          Distance = 0
7
8          DeviceFound = False
9
10         # Loop until either tag wasn't found or distance is 10 mm
11         while DeviceFound and Distance < 11:
12                 # Check if the device under test can detect the tag
13                 DeviceFound = TE.DUT.SearchForDevice() == "1"
14                 # Move device under test one mm (= 4 steps) back
15                 TE.Robot.Move(-4, 0, 0)
16                 # Distance has increased
17                 Distance += 1
18
19         # End of test
20         # If distance is 11, DUT passed the test.
21         if Distance == 11 and DeviceFound:
22                 TE.Result = "Passed"
```

Figure 2: Python script performing *NFC Forum* test TP_2_1_1

the point where the device under test and the target touch each other.

4.1.2 The script

When the test is started, the main application calls a function *Run* in the test script and passes the testing environment (*TE*) to it. The testing environment holds objects like the robot, a log and the device under test (DUT) which can be used by the script. The example script defines that by the default the device will fail the test and that it starts testing with the device under test and the target touching each other. After that it enters a loop which will be executed until either the device under test could not be detected by the target or the maximum distance (10 mm) was reached. The loop consists of testing if the target can be detected (Line 13), moving the robot back by 4 steps (= 1 mm) (Line 15) and keeping track of the distance (Line 17). If the main loop is left, the test script determines if the device under test passed the test or not.

5 Predefined Tests

These tests are already defined in a module provided by the main application. On the one hand, these are tests as defined by the *NFC Forum*, on the other hand simple shapes are provided which are used for amplitude measurements. These tests are represented as *Python* classes and can be used as base classes for custom tests.

5.1 Window tests

5.1.1 Description

Window tests are used to verify that the device under test can detect a target in a specified window. Therefor the device under test is placed in the robots bracket and the target is positioned at a specified point in space. The robot moves the device to all required positions and the main application then sends a request to the device under test. It checks if it can detect a target and returns the result to the main application. If the device under test can detect the target at all positions, it passes the test.

5.1.2 Requirements

- Robot
- Device under test
- Target

5.1.3 Available tests and results

Test	**Range [mm]**	**Resolution [mm]**	***Nokia 6131 NFC***	***Samsung SGH-X700N***
TP_2_1_1	0 - 10	1	Passed	Passed
TP_2_1_2	10 - 30	5	Failed	Failed
TP_2_1_3	30 - 50	5	Failed	Failed
TP_2_1_4	50 - 100	5	Failed	Failed
TP_2_2_1	0 - 10	1	Passed	Passed
TP_2_2_2	10 - 50	5	Failed	Failed
TP_2_2_3	50 - 100	5	Failed	Failed

5.2 Range tests

5.2.1 Description

These tests verify a range in which a device under test can detect other *NFC* devices. When the test starts, the target and the device under test touch each other. The device under test is now moved step by step away from the target. At each step, the main application sends a request to the device under test which checks if it can detect a target and returns the result to the main application. The test ends when no device can be detected three times in succession. The result of the test is the distance closest to the target where the device under test can not detect the target anymore.

5.2.2 Requirements

- Robot
- Device under test
- Target

5.2.3 Available tests and results

Test	***Nokia 6131 NFC***	***Samsung SGH-X700N***
TP_2_1_5	14 mm	11 mm
TP_2_2_4	14 mm	11 mm

5.3 Amplitude measurements

5.3.1 Description

To analyze the electro magnetic field emitted by the device under test as seen by the target, an antenna is attached to the oscilloscope and placed instead or near the target. If a target is used, the antenna should be placed between the device under test and the target. The tests start with the device under test as close to the antenna as possible. The software on the device should permanently try to find other *NFC* devices. The device under test is then moved to all specified positions and the amplitude, as measured by the oscilloscope, is recorded.

5.3.2 Requirements

- Robot
- Device under test
- Oscilloscope
- *optional:* Target

5.3.3 Available tests and results

Test	Directions	Resolution	Remarks
Plain - XY	XY	5 mm	Two dimensional plane in X- and Y-direction
Plain - XZ	XZ	5 mm	Two dimensional plane in X- and Z-direction
Plain - YZ	YZ	5 mm	Two dimensional plane in Y- and Z-direction
Plain - All	XYZ	5 mm	Runs tests Plain - XY, Plain - XZ and Plain - YZ
Cube - 5	XYZ	5 mm	Tests positions in a 5 cm cube

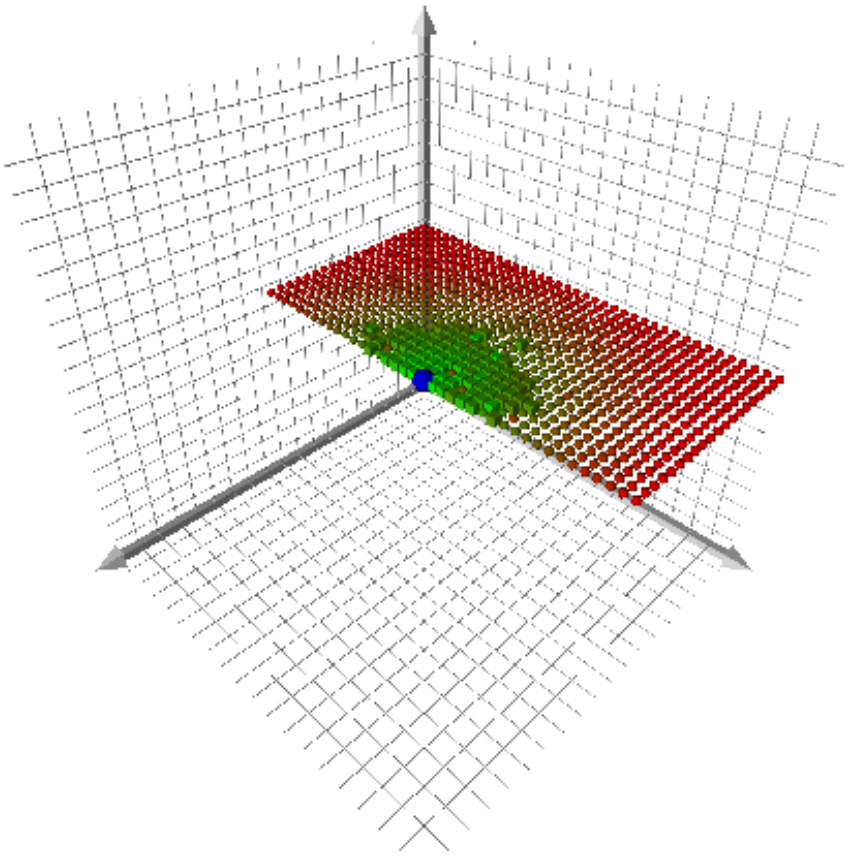

Figure 3: Result for amplitude test Cube - 5, *Samsung SGH-X700N*