|  |  |
| --- | --- |
| logo ngan.png | **MINISTRY OF EDUCATION AND TRAINING** |

|  |
| --- |
| **FPT UNIVERSITY** |
| Capstone Project Document |
| **Design and Implement wireless RF 2.4 GHz Mouse** |
|  |
| |  |  | | --- | --- | | **TouchpadRF Team** | | | **Group Members** | Lê Tùng Bách – Leader – BachLT  Nguyễn Chí Cường – Member – CuongNC | | **Supervisor** | **Nguyễn Trọng Phúc** | | **Ext Supervisor** | N/A | | **Capstone Project code** | TouchpadRF | |

Ho Chi Minh City, 12/2013

**Acknowledgements**

We really appreciate the help and guidance of Mr. Nguyen Trong Phuc – our Research Supervisor, and because of all the supports, recommendations, we could have come this far to be able to complete this project.

In addition, we are thankful for the help of borrowing equipment from the previous team.

Contents

[I. INTRODUCTION 10](#_Toc374223257)

[1. Project Information 11](#_Toc374223258)

[2. Project Team 11](#_Toc374223259)

[3. Project Introduction 11](#_Toc374223260)

[a. Ideas 11](#_Toc374223261)

[b. Existing technologies and products 12](#_Toc374223262)

[c. About our Project 15](#_Toc374223263)

[II. SOFTWARE PROJECT MANAGEMENT PLAN (SPMP) 17](#_Toc374223264)

[1. Problem Definition 18](#_Toc374223265)

[a. Capstone Project Name 18](#_Toc374223266)

[b. Project Abstract 18](#_Toc374223267)

[c. Project Overview 18](#_Toc374223268)

[2. Project Organization 21](#_Toc374223269)

[a. Develop Process Model 21](#_Toc374223270)

[b. Constraints 22](#_Toc374223271)

[c. Assumptions 22](#_Toc374223272)

[d. Scope Overview 22](#_Toc374223273)

[e. Budget Overview 22](#_Toc374223274)

[f. Project Team 23](#_Toc374223275)

[g. Tools and Techniques 23](#_Toc374223276)

[3. Project Management Plan 25](#_Toc374223277)

[a. Risk 25](#_Toc374223278)

[b. Task 28](#_Toc374223279)

[c. Task Sheet: Assignment and Timetable 31](#_Toc374223280)

[d. All Meeting Minutes 31](#_Toc374223281)

[4. Coding Convention 39](#_Toc374223282)

[5. Other Material 39](#_Toc374223283)

[III. SOFTWARE REQUIREMENTS SPECIFICATIONS (SRS) 40](#_Toc374223284)

[1. User Requirement Specification 41](#_Toc374223285)

[2. System Requirement Specification (Specific Requirements) 42](#_Toc374223286)

[a. External Interface Requirement 42](#_Toc374223287)

[b. Functional Requirement Specification 43](#_Toc374223288)

[c. Non-Functional Requirement Specification 55](#_Toc374223289)

[3. Other Material (if any) 56](#_Toc374223290)

[IV. SOFTWARE DESIGN SPECIFICATION (SDS) 57](#_Toc374223291)

[1. Design Overview 58](#_Toc374223292)

[2. System Architectural Design 58](#_Toc374223293)

[a. Device 61](#_Toc374223294)

[3. Diagram 76](#_Toc374223295)

[a. Transmitter Overview Diagram 76](#_Toc374223296)

[b. Receiver Overview Diagram 78](#_Toc374223297)

[4. Software Design 80](#_Toc374223298)

[a. Overview System Sequence Diagram 80](#_Toc374223299)

[b. Flow Chart RF 83](#_Toc374223300)

[c. Transmitter Sequence Diagram 84](#_Toc374223301)

[d. Receiver Sequence Diagram 89](#_Toc374223302)

[V. SYSTEM IMPLEMENTATION & TEST (SIT) 91](#_Toc374223303)

[1. Introduction 92](#_Toc374223304)

[a. Overview 92](#_Toc374223305)

[b. Test Approach 92](#_Toc374223306)

[2. Implementation 92](#_Toc374223307)

[a. Transmitter 92](#_Toc374223308)

[b. Receiver 92](#_Toc374223309)

[3. Test Plan 92](#_Toc374223310)

[a. Features to be tested 92](#_Toc374223311)

[b. Features not to be tested 94](#_Toc374223312)

[c. Testing Tools and Environments 94](#_Toc374223313)

[d. Test Pass/Fail Criteria 94](#_Toc374223314)

[4. Test Case 94](#_Toc374223315)

[a. Component Testing 95](#_Toc374223316)

[b. System Testing 113](#_Toc374223317)

[VI. SOFTWARE USER’S MANUAL 128](#_Toc374223318)

[1. Package Contents 129](#_Toc374223319)

[2. Installation Guide 130](#_Toc374223320)

[3. User Guide 132](#_Toc374223321)

[VII. REFERENCES 133](#_Toc374223322)

Table of Figures

[Figure 1 - System Boundary 19](#_Toc374223169)

[Figure 2 – Iterative Development Model 21](#_Toc374223170)

[Figure 3 – Touchpad Interface 42](#_Toc374223171)

[Figure 4 – Overview System Process 59](#_Toc374223172)

[Figure 5 – MSP430G2553 Chip 62](#_Toc374223173)

[Figure 6 – MSP430G2553 PIN Diagram 63](#_Toc374223174)

[Figure 7 - MSP430G2553 Functional Block Diagram 63](#_Toc374223175)

[Figure 8 – MSP430F6659 Chip 65](#_Toc374223176)

[Figure 9 – MSP430F6659 PIN Diagram 66](#_Toc374223177)

[Figure 10 – nRF24L01 69](#_Toc374223178)

[Figure 11 – nRF24L01 PIN Diagram 70](#_Toc374223179)

[Figure 12 – TM5957 System Interface Block Diagram 74](#_Toc374223180)

[Figure 13 – Touchpad Module Design Dimensions 75](#_Toc374223181)

[Figure 14 – Transmitter Overview Flowchart 76](#_Toc374223182)

[Figure 15 – Receiver Overview Flowchart 78](#_Toc374223183)

[Figure 21 – Overview System Sequence Diagram 80](#_Toc374223184)

[Figure 22 – RF working process flowchart 83](#_Toc374223185)

[Figure 23 – Left-Click Action Sequence Diagram 84](#_Toc374223186)

[Figure 24 – Right Click Action Sequence Diagram 85](#_Toc374223187)

[Figure 25 – Move Action Sequence Diagram 86](#_Toc374223188)

[Figure 26 – Vertical Scroll Action Sequence Diagram 87](#_Toc374223189)

[Figure 27 – Receiver Sequence Diagram 89](#_Toc374223190)

Table of Tables

[Table 1 – MSP430G2553 Microcontroller Requirements 61](#_Toc374223162)

[Table 2 – MSP430G2553 Microcontroller Requirements 64](#_Toc374223163)

[Table 3 – MSP430F6659 Features 67](#_Toc374223164)

[Table 4 – nRF24L01 Requirements 69](#_Toc374223165)

[Table 5 – Transmitter Operation 70](#_Toc374223166)

[Table 6 – Receiver Operation 71](#_Toc374223167)

[Table 7 – Features to be Tested 94](#_Toc374223168)

# INTRODUCTION

## Project Information

* Project Name: Design and Implement wireless RF 2.4 GHz Mouse
* Project Code: TouchpadRF
* Abbreviation: Wireless Mouse, RF 2.4 GHz, TI
* Timeline: September 10th 2013 to December 9th 2013.

## Project Team

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Name | Role | Contact |
| 1 | Lê Tùng Bách | Leader | * Phone: 090835441 * Y!M: * Email: bachlt |
| 2 | Nguyễn Chí Cường | Member | * Phone: 01238700255 * Y!M: bluebubble.monster20 * Skype: angelhuker * Email: angel.huker@gmail.com |

## Project Introduction

### Ideas

With a few ideas of a pointer device like:

* An input device which can work as touchpad
* The touchpad has a larger area than others.
* The device needs to be easy-to-use and don’t require user to learn any new skills
* The device has compatibility with many kind of Operating System: Windows XP/7/8, Linux, Mac.
* Compact & wireless: people require a device which small, portable, plug-and-play…

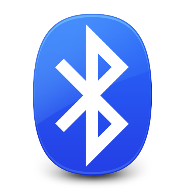
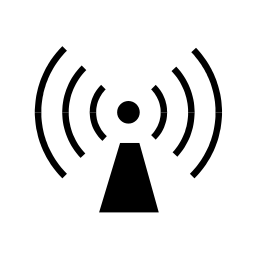
Bring the above ideas, we start to work on it and create a product – TouchPad

### Existing technologies and products

Nowadays, touch screen is one of the technology trends that can bring joys to users and create brand new experience. There are three majors touch sense technologies:

* **Resistive**: Include 2 thin, transparent electrically-resistive layers separated by a thin space (spacer dots). When active, the current will pass through the screen, the circuit layer will interact with each other and determine where was touched on.
  + Advantage: Cheap, Easy to manufacture, Can use anything to touch (fingertips, keys, pens,… )
  + Disadvantage: easy to get scratched, low durability, single point touch sense, low brightness.
* **Capacitive**: have an insulator such as glass, coated with a transparent conductor such as indium tin oxide (ITO). The sensor's controller can determine the location of the touch indirectly from the change in the capacitance as measured from the four corners of the panel: the larger the change in capacitance, the closer the touch is to that corner.
  + Advantage: Multi-point touch sense, high brightness, high durability, hard to get scratched
  + Disadvantage: expensive, only some materials can “touch”.
* **Infrared grid**: Sensors are located on the top and around screen, emit infrared beams, create infrared grid. When we touch the infrared gird, it will be broken, thereby, touched point will be determined.
  + Advantage: extremely sensitive, extremely accurate, can use anything to touch
  + Disadvantage: very expensive.

Talk about the transmitter, there are about 3 types wireless technologies that we can possibly learn or try to use:

* **Infrared (IR) **
* **Bluetooth**
* **Radio Frequency (RF)** 

And all of those three are using PS/2 or USB ports on computer. For PS/2-compatible versions, they divide into 2 ports, 1 PS/2 port for mouse and another one for keyboard. For USB-compatible versions, they only need to use 1 USB port for both.

And here is some critical clear about the advantages and disadvantages of these 3 technologies.

* **Infrared (IR):** Infrared technology allows computing devices to communicate via short-range wireless signals. With infrared, computers can transfer files and other digital data bi-directionally. The infrared transmission technology used in computers is similar to that used in consumer product remote control units. Some of advantages of infrared are portable and low power requirement, therefore ideal for laptops, telephones, personal digital assistants. Besides, the directionality of the beam helps ensure that data isn't leaked or spilled to nearby devices as its transmitted make infrared communication is higher security. One of advantages of infrared is high noise immunity, not as likely to have interference from signals from other devices. However, some of disadvantages make the infrared communication make obstacle in itself growing. First, line of sight - transmitters and receivers must be almost directly aligned (i.e. able to see each other) to communicate. Area range is too short and performance drops off with longer distances. Place two infrared devices within a few feet (no more than 5 meters) of each other when networking them. It can be not affected by other signal from other devices, on the contract it’s too sensitive to light and weather (direct sunlight, rain, fog, dust, pollution). So, it only use indoor or evening. In the other hand, Infrared is blocked by common materials: people, walls, plants, etc. can block transmission. Data transmission speed is lower than typical wired transmission (2400 bps to 4 Mbps).
* **Bluetooth:** a touchpad using Bluetooth protocol is a touchpad connects to its master device via Bluetooth. There are many advantages of Bluetooth such as compatibility, mobility. Its compatibility presents the most at the ability to be used in every OS: Windows, Linux, Android, iOS and Mac OS. And the mobility is that it can be used by many different devices and can be taken anywhere. And since Bluetooth does not require to be in a clear line of sight between the synced devices, the devices need not be facing each other or it is still possible for devices to communicate when both are in separate rooms. However, this technology still have some problems. There are four mains disadvantages of Bluetooth. Firstly, Bluetooth does not have user-friendly. Bluetooth devices may be easy to use once they are setup and configured, the initial configuration may be too complex for some users. Secondly, Bluetooth devices is that they lack any kind of significant security due to the wide range of the device. Users with Bluetooth keyboard do not have to worry about data theft from hackers but they do have to deal with possible interference from other Bluetooth devices. In addition, data sent between two Bluetooth devices has a maximum transfer speed of one megabyte per second. The slow transfer speed makes Bluetooth not an ideal choice for data transfer when faster connection methods are available. Furthermore, another form of this is range of Bluetooth wave. Bluetooth only has a range of 15 to 30 feet depending on the device. The small range is a disadvantage for some who may want to use a Bluetooth device outside of that 30-foot radius. In addition, Bluetooth is not secure completely, since data is transferred over radio waves using Bluetooth, it is much easier for a hacker to break into than secure wireless networks. Finally, Bluetooth must deal with battery problems. It uses the battery power of a particular device in order to operate. Many cell phone makers send phones out with Bluetooth powered off in order to maximize the battery life of the phone. Turning Bluetooth on, while convenient, can drain a cell phone or computer battery, causing it to run out of juice long before it would have if Bluetooth was not powered on. Although Bluetooth is a solution that solve wireless connection relatively, it still has some problems such as user-unfriendly, security, narrow range, transfer rates, and battery.
* **Radio Frequency (RF)**: It provides an exact solution to meet our specific needs. All the configuration about distance, channel, throughput,… can be modified as our wish. And it can be extremly secure, so that the customer can be comfortable of using it without being worried about information leaking. Futhermore, it can work in a low-power state, and can help transmitter device save energy. And RF wireless can carry 10Mbps, and it’s not light sensitive or depend on weather or eviroment

### About our Project

#### Features

Our products not only have some common features like the regular wireless touchpad but also have some improvements.

* Plug-and-play: without need to install any software, users only have to plug the receiver and use the device.
* Easy-to-use: by using the common style of touchpad, we bring users the easiest way to interact.
* Multiplatform: compatible with many operating systems such as Windows (XP, 7, 8), Linux (Ubuntu, Fedora,…), Mac OS and any platform which supports USB HID.
* Multi device: can use on any devices support USB port.
* Security: our product is divided into two part transmitter and receiver. With hardware pairing, one transmitter only pair with one receiver which has the same memory allocation and channel. Besides, data transfer between transmitter and receiver is encrypted with AES (Advanced Encryption Standard) so very difficult to steal data package.
* Ultra-Low Power: our product use ultra-low power microchip to reduce energy cost.

#### Scope

* Our product are divided into two part
* Transmitter:
  + Touchpad:
    - Buttons: No Buttons
    - Size:
      * Length: 65.80 mm
      * Width: 49.80 mm
      * Thickness: 4.80 mm
      * Weight: < 14 g
    - Active Sensing Area:
      * Length: 61.80 mm
      * Width: 45.80 mm
* Receiver:
  + Board:
    - Length: 60 mm
    - Width: 30 mm
* Budget: 3.000.000 VND
* Schedule: September 10th 2013 to December 9th 2013

# SOFTWARE PROJECT MANAGEMENT PLAN (SPMP)

## Problem Definition

### Capstone Project Name

Project Full Name: Design and Implement Wireless RF 2.4 GHz Mouse

### Project Abstract

Design and implement a Wireless RF 2.4GHz TouchPad using TI microchip and RF technology. And this wireless touchpad can work on any kind of OS

### Project Overview

#### The Current System

Currently, they are using infrared of Bluetooth technology for the system, and it have lots of problems like: high costs energy, narrow range, data transmission speed, low security and unfriendly-UI,…

#### The Proposed System

We intend to build a system from TI MSP430 microchip, Nordic board for RF transmit and Cirque Touchpad Module, and it’s going to be the lowest cost that we could do. By using an USB HID standard, the device will not need to be installed any additional driver or any configuration, and because of that, the device can work on many platforms. From the materials that we used, we believe that we are building a touchpad device which can be low cost, energy-saving, wide range and user friendly. Components:

* USB HID Specifications
* TI MSP430G2553
* TI MSP430F6659
* RF Protocol
* I2C & SPI communication Interface
* Cirque TM5957
* RF 2.4 GHz Antenna Design
* C language for microcontroller

#### Boundaries of the System

* Inner system includes Transmitter and Receiver parts
* Outer system includes devices which runs on OS platform support USB HID

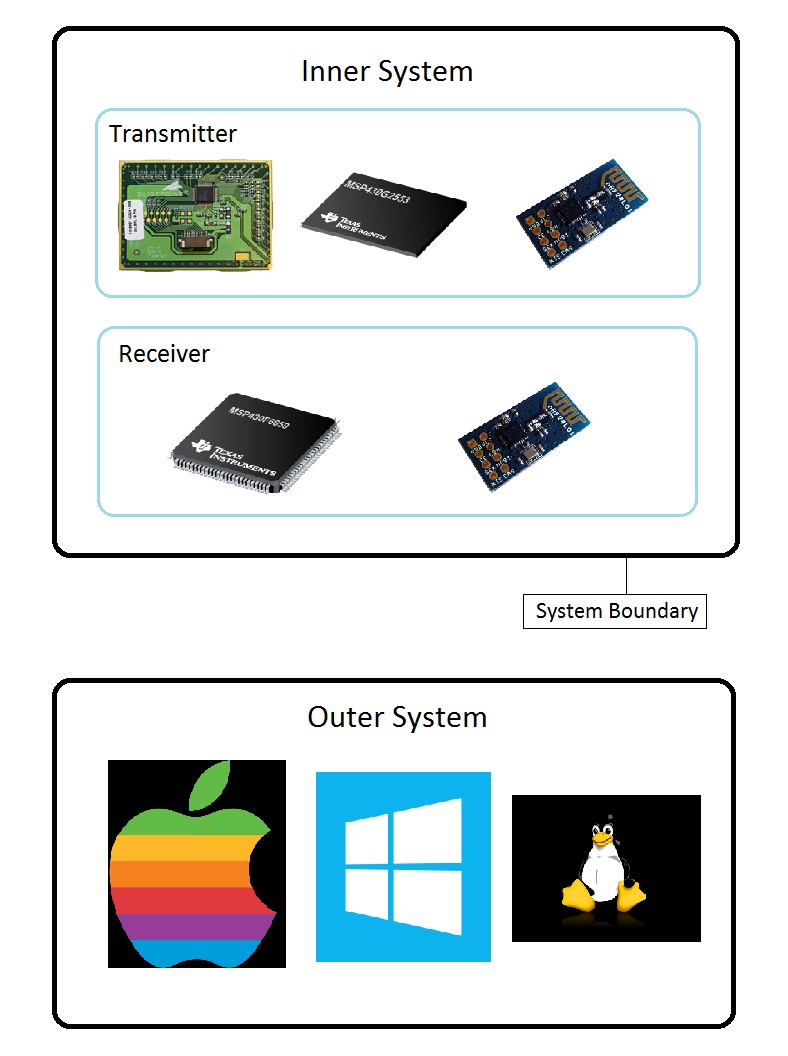


Figure 1 - System Boundary

#### Development Environment

In order to build the system, we use CCS (Code Composer Studio™) as an integrated development environment (IDE), it is a Texas Instruments (TI) developer KIT for embedded processor families. CCS includes a suite of tools used to develop and debug embedded applications, compliers for each of TI’s device families, source code editor, project-build environment, debugger, profiler, simulators, real-time operating system and many other features. The intuitive IDE provides a single user interface taking you through each step of the application development flow. Familiar tools and interfaces allow users to get started faster than ever before and add functionality to their application thanks to sophisticated productivity tools.

#### Statement of Work

This effort includes the following:

* Studying about Nordic nRF24L01, TI MSP430F6659 and TI MSP40G2553
* Create project management plan
* Define critical function and requirement
* Develop Hardware/ Software
* Write progress reports
* Testing
* Debugging
* Write final report

## Project Organization

### Develop Process Model

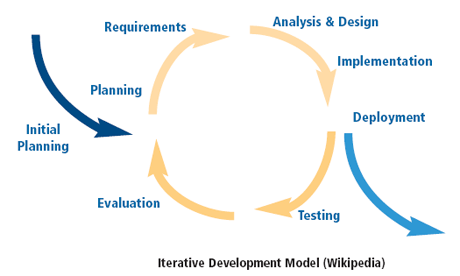


Figure 2 – Iterative Development Model

The reasons why we choose iteration model in our project are:

* **Maintainability**: Creating a separate process integration layer allows users to define and maintain the business process independent from the implementation of the individual functions. This increases maintainability and reduce the skill set requirements for the personnel who maintain the process definition.
* **Reusability**: Because the existing applications are not dependent on the process management layer, the functions inside these applications can be reused in multiple process definitions.
* **Flexibility**: The process manager can support a variety of configurations that would be difficult to implement in many traditional programming models. For example, parallel execution of multiple tasks, synchronization points, timeouts, and escalations are all configurations that would be difficult to implement in a traditional programming model. Supporting a variety of configurations gives the process manager the flexibility to adapt to many different business requirements.
* **Reporting** **capability**: Because the process instances are maintained centrally, it becomes feasible to extract statistical information that spans multiple process steps and process instances. Such reports can provide answers to questions such as "How long does it take on average to fulfill an order?" or "How many orders are on hold due to insufficient inventory?" This type of reporting can be instrumental in making informed business decisions.

### Constraints

Below are the limitations and constraints of the project:

* Limited time and budge
* Limited lab equipment
* Depend on unsure resources on the Internet for learning and studying.

### Assumptions

Below are the assumptions when developing this Project Charter:

* No hardware failure during implementation
* Team members are available at least 80% of time

### Scope Overview

Below are features need to be implements to build the final function:

* Touchpad can transmit user’s interaction on it
* Coordinates of moving or actions can be detected by the microchip
* Coordinates of moving or actions can be transmit to receiver by RF
* Receiver can be receive and report back the coordinates and actions to device by USB HID
* This touchpad system can be used on any device which support mouse HID such as Windows, Ubuntu, MacOS,…

### Budget Overview

Estimated budget for the project: 3,000,000 VND

* Borrow MSP-TS430PZ100USB and FET430UIF for debugging on MSP430F6xxx families.
* About Chip, we intend to get free sample from TI cause there will be only 2 or 3 chips needed
* Buy MSP-EXP430G2 Launchpad for debugging on MSP430G2xxx families
* Get free sample from Cirque TM5957
* Buy 2 new RF components – Nordic’s nRF24L01

### Project Team

Below are people and organizations are stakeholders in this project and included in the project planning. Additional project team members are added as needed.

Executive Sponsors: Commit resources & advocate for project

|  |  |  |
| --- | --- | --- |
| Name | Role | Responsibilities |
| Nguyen Trong Phuc | Supervisor | Support, advise and tracking project status |
| Le Tung Bach | PM, TL, Dev, Tester | * Create plan * Control plan, tracking project status * Solve risk * Build and test hardware * Build and test software |
| Nguyen Chi Cuong | QA, Dev, Tester | * Creating document * Build and test hardware * Build and test software |

### Tools and Techniques

#### Tools

* Code Composer Studio v5.4.0
* MSP-TS430PZ100USB
* MSP-EXP430G2

#### Techniques

* SPI, UART, I2C, Timer.
* USB HID (USB Human Interface Device).
* Touch Sensing.

## Project Management Plan

### Risk

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Risk | Description | Category | Root Cause | Triggers | Potential Responses | Probability |
| Communication | People in project  misunderstand each other well. | People | Project  members lack  of  communication | Misunderstanding,  delay of project | Meeting  frequently,  communicate face  to face | High |
| Team spirit and  attitude | Team spirit go down and do not  want to do everything or  thinking not creative | People | Difficult  problems, busy  with personal  stuffs | Team members  become tired and  lazy  Delay of project | Change to other  environments or  built a teambuilding meeting to  encourage and  motivate the team  spirit | High |
| Circuit | In printing circuit process,  sometime the circuit failed and  can’t run | Process | Design wrong  circuit  Don’t’ have  experiment in  printing circuit | Delay a phase of  project | Check the problem  Print circuit again | High |
| Sickness | Project members out of work  because of sickness | People | Sickness | Delay a phase of  project | Arrange backup  plan, buffer  workload | Medium |
| Material quality | The quality of device that  supply in Vietnam sometimes  not good enough | Material | Depend on  suppliers | Delay a phase of  project | Find others  suppliers | Medium |
| Material  supplying | The supplier not supply the  material on time  In project, material can be  broken by wrong action of  member in team. So need time  to request from supplier and  time for supplying | Material | Don’t have  experiment on  choosing the  prestige  supplier  Don’t have  experiment on  manipulation  of device | Delay a phase of  project | Ask supervisor for  advice | Medium |
| Not meet  deadline | Not meet the deadline that plan  in project management plane | Process | A lot of root  causes | Delay a phase of  project | Work overtime to  meet deadline | Medium |
| Money | Lack of money for buying  material | Budget | Shortage of  money | Delay a phase of  project | Loan from friends,  investigator | Low |

### Task

#### Task 1: System Analysis

|  |  |
| --- | --- |
| **Description** | Study about system to know what the system do. The functional and non-functional requirements must be listed. Furthermore, define what chips, technology can be used to deal with |
| **Deliverables** | System requirement description document (SRS) |
| **Resources Needed** |  |
| **Dependencies and Constraints** |  |
| **Risk** | Maybe can’t figure out all the function or non-functional requirements. The selected chips may not be appropriated |

#### Task 2: Hardware Design

|  |  |
| --- | --- |
| **Description** | After this task, a schematics and PCB layout of hardware should be designed. The schematic must be follow what are defined in SRS document and they must enable to create physical board |
| **Deliverables** | Hardware schematics and PCB layout, system design description (SDD) |
| **Resources Needed** |  |
| **Dependencies and Constraints** |  |
| **Risk** | Cause lacking of knowledge, the result mays unrealistic or wrong principal schematics and PCB layout |

#### Task 3: Hardware Implement

|  |  |
| --- | --- |
| **Description** | In order to build an embedded product, there are two parts must be implemented, one is hardware development and another is software development. So after this task, a physical boards with chips and another features are simply created followed by schematic and PCB layout |
| **Deliverables** | Physical boards with chips |
| **Resources Needed** |  |
| **Dependencies and Constraints** |  |
| **Risk** | Bug happens, all the time, since the chips have been welded on the board, debugging seems to be impossible |

#### Task 4: Hardware Testing

|  |  |
| --- | --- |
| **Description** | After finishing boards, to guarantee that everything works fine, it must be tested |
| **Deliverables** | Testing report |
| **Resources Needed** |  |
| **Dependencies and Constraints** |  |
| **Risk** | **:** Can only test on all the cases that we could possibly think of. So maybe can’t detect all the errors |

#### Task 5: Software Design

|  |  |
| --- | --- |
| **Description** | This task is about defining how the software will be coded – allocation, dataflow, system behavior and all things that needed for implementation |
| **Deliverables** | System design description (SDD) |
| **Resources Needed** |  |
| **Dependencies and Constraints** |  |
| **Risk** |  |

#### Task 6: Software Implement

|  |  |
| --- | --- |
| **Description** | Coding follow the software detail design |
| **Deliverables** | Source code can run on hardware |
| **Resources Needed** |  |
| **Dependencies and Constraints** |  |
| **Risk** |  |

#### Task 7: Software Testing

|  |  |
| --- | --- |
| **Description** | Testing source code with the test cases to ensure that it meets detail design |
| **Deliverables** | System Implementation & Test (STD) |
| **Resources Needed** |  |
| **Dependencies and Constraints** |  |
| **Risk** |  |

#### Task 8: Integrated Testing

|  |  |
| --- | --- |
| **Description** | After testing hardware and software, source code must be implemented into chips. Objective of this task is to test whether the source code run on hardware is correct or not |
| **Deliverables** | Software test document (STD) |
| **Resources Needed** |  |
| **Dependencies and Constraints** |  |
| **Risk** |  |

### Task Sheet: Assignment and Timetable



### All Meeting Minutes

#### Kick-off Meeting – Sep 3rd 2013

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Subject | | Kick-off Meeting | | | Date | Sep 3, 2013 |
| Facilitator | | PhucNT | | | Time | 7:45 – 8:30 |
| Location | | Quang Trung Software Park | | | Scribe | CuongNC |
| Attendees | | BachLT, CuongNC | | | | |
| Absent | |  | | | | |
| Key Points Discussed | | | | | | |
| No. | Topic | | Highlights | | | |
| 1. | Team member introduction | | Introduce all members of team to supervisor | | | |
| 2. | Project introduction | | Introduce briefly about the project | | | |
| Action Plan | | | | | | |
| No. | Action Item(s) | | | PIC | | Target Date |
|  | N/A | | |  | |  |

#### Weekly meeting – Sep 10th 2013

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Subject | | Weekly meeting 1 | | | Date | Sep 10, 2013 |
| Facilitator | | PhucNT | | | Time | 7:45 – 8:30 |
| Location | | Quang Trung Software Park | | | Scribe | CuongNC |
| Attendees | | BachLT, CuongNC | | | | |
| Absent | |  | | | | |
| Key Points Discussed | | | | | | |
| No. | Topic | | Highlights | | | |
| 1. | Discuss about software technical requirement | | Make technical requirements clear | | | |
| 2. | Discuss about report 1 | |  | | | |
| Action Plan | | | | | | |
| No. | Action Item(s) | | | PIC | | Target Date |
| **1.** | Create report 1 | | | CuongNC | | Sep 14 |
| **2.** | Study MSP430G2553 | | | BachLT | |  |
| **3.** | Study MSP430F6659 | | | CuongNC | |  |

#### Weekly meeting – Sep 17th 2013

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Subject | | Weekly meeting 2 | | | Date | Sep 17, 2013 |
| Facilitator | | PhucNT | | | Time | 7:45 – 8:30 |
| Location | | Quang Trung Software Park | | | Scribe | CuongNC |
| Attendees | | BachLT, CuongNC | | | | |
| Absent | |  | | | | |
| Key Points Discussed | | | | | | |
| No. | Topic | | Highlights | | | |
| 1. | Discuss about report 2 | |  | | | |
| 2. | Borrow devices | |  | | | |
| Action Plan | | | | | | |
| No. | Action Item(s) | | | PIC | | Target Date |
| **1.** | Create report 2 | | | CuongNC | | Sep 21 |
| **2.** | Study MSP430G2553 | | | BachLT | |  |
| **3.** | Study MSP430F6659 | | | CuongNC | |  |

#### Weekly Meeting – Sep 24th 2013

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Subject | | Weekly meeting 3 | | | Date | Sep 17, 2013 |
| Facilitator | | PhucNT | | | Time | 7:45 – 8:30 |
| Location | | Quang Trung Software Park | | | Scribe | CuongNC |
| Attendees | | BachLT, CuongNC | | | | |
| Absent | |  | | | | |
| Key Points Discussed | | | | | | |
| No. | Topic | | Highlights | | | |
| 1. | Discuss about report 3 | |  | | | |
| Action Plan | | | | | | |
| No. | Action Item(s) | | | PIC | | Target Date |
| **1.** | Create report 3 | | | CuongNC  BachLT | | Sep 28 |

#### Weekly Meeting – Oct 1st 2013

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Subject | | Weekly meeting 4 | | Date | Oct 1, 2013 |
| Facilitator | | PhucNT | | Time | 7:45 – 8:30 |
| Location | | Quang Trung Software Park | | Scribe | CuongNC |
| Attendees | | BachLT, CuongNC | | | |
| Absent | |  | | | |
| Key Points Discussed | | | | | |
| No. | Topic | | Highlights | | |
| 1. | Discuss about report 3, correct report 3 | |  | | |
| 2. | Discuss about USB HID | |  | | |
| Action Plan | | | | | |
| No. | Action Item(s) | | PIC | | Target Date |
| **1.** | Complete report 3 | | CuongNC  BachLT | |  |
| **2.** | Study USB HID | | CuongNC | |  |
| **3.** | Study SPI MSP430G2553 | | BachLT | |  |
| **4.** | Study SPI MSP430F6659 | | CuongNC | |  |
| **5.** | Study I2C MSP430G2553 | | BachLT | |  |

#### Weekly Meeting – Oct 8th 2013

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Subject | | Weekly meeting 5 | | Date | Oct 8, 2013 |
| Facilitator | | PhucNT | | Time | 7:45 – 8:30 |
| Location | | Quang Trung Software Park | | Scribe | CuongNC |
| Attendees | | BachLT, CuongNC | | | |
| Absent | |  | | | |
| Key Points Discussed | | | | | |
| No. | Topic | | Highlights | | |
| 1. | Discuss about report 4 | |  | | |
| 2. | Checking project progress | |  | | |
| Action Plan | | | | | |
| No. | Action Item(s) | | PIC | | Target Date |
| **1.** | Create report 4 | | CuongNC  BachLT | |  |
| **2.** | Study USB HID | | CuongNC | |  |
| **3.** | Study SPI MSP430G2553 | | BachLT | |  |
| **4.** | Study SPI MSP430F6659 | | CuongNC | |  |
| **5.** | Study I2C MSP430G2553 | | BachLT | |  |

#### Weekly Meeting – Oct 15th 2013

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Subject | | Weekly meeting 6 | | Date | Oct 15, 2013 |
| Facilitator | | PhucNT | | Time | 7:45 – 8:30 |
| Location | | Quang Trung Software Park | | Scribe | CuongNC |
| Attendees | | BachLT, CuongNC | | | |
| Absent | |  | | | |
| Key Points Discussed | | | | | |
| No. | Topic | | Highlights | | |
| 1. | Discuss about report 4 | |  | | |
| 2. | Checking project progress | |  | | |
| Action Plan | | | | | |
| No. | Action Item(s) | | PIC | | Target Date |
| **1.** | Complete report 4 | | CuongNC  BachLT | |  |
| **2.** | Study Touchpad TM5957 | | BachLT | |  |
| **3.** | Study nRF24L01 | | CuongNC | |  |

#### Weekly Meeting – Oct 22nd 2013

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Subject | | Weekly meeting 7 | | Date | Oct 22, 2013 |
| Facilitator | | PhucNT | | Time | 7:45 – 8:30 |
| Location | | Quang Trung Software Park | | Scribe | CuongNC |
| Attendees | | BachLT, CuongNC | | | |
| Absent | |  | | | |
| Key Points Discussed | | | | | |
| No. | Topic | | Highlights | | |
| 1. | Discuss about report 4 | |  | | |
| 2. | Checking project progress | |  | | |
| Action Plan | | | | | |
| No. | Action Item(s) | | PIC | | Target Date |
| **1.** | Complete report 4 | | CuongNC  BachLT | |  |
| **2.** | Study Touchpad TM5957 | | BachLT | |  |
| **3.** | Study nRF24L01 | | CuongNC | |  |

#### Weekly Meeting – Oct 29th 2013

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Subject | | Weekly meeting 8 | | Date | Oct 29, 2013 |
| Facilitator | | PhucNT | | Time | 7:45 – 8:30 |
| Location | | Quang Trung Software Park | | Scribe | CuongNC |
| Attendees | | BachLT, CuongNC | | | |
| Absent | |  | | | |
| Key Points Discussed | | | | | |
| No. | Topic | | Highlights | | |
| 1. | Discuss about report 5 | |  | | |
| 2. | Discuss about the Guide Implementation | |  | | |
| Action Plan | | | | | |
| No. | Action Item(s) | | PIC | | Target Date |
| **1.** | Create report 5 | | CuongNC  BachLT | |  |
| **2.** | Coding SPI protocol for MSP430 | | BachLT | |  |
| **3.** | Coding USB HID | | CuongNC | |  |

#### Weekly Meeting – Nov 5th 2013

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Subject | | Weekly meeting 9 | | Date | Nov 5, 2013 |
| Facilitator | | PhucNT | | Time | 7:45 – 8:30 |
| Location | | Quang Trung Software Park | | Scribe | CuongNC |
| Attendees | | BachLT, CuongNC | | | |
| Absent | |  | | | |
| Key Points Discussed | | | | | |
| No. | Topic | | Highlights | | |
| 1. | Discuss about report 5 | |  | | |
| 2. | Discuss about the Guide Implementation | |  | | |
| Action Plan | | | | | |
| No. | Action Item(s) | | PIC | | Target Date |
| **1.** | Complete report 5 | | CuongNC  BachLT | |  |
| **2.** | Coding SPI protocol for MSP430 | | BachLT | |  |
| **3.** | Coding USB HID | | CuongNC | |  |

#### Weekly Meeting – Nov 12nd 2013

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Subject | | Weekly meeting 10 | | Date | Nov 12, 2013 |
| Facilitator | | PhucNT | | Time | 7:45 – 8:30 |
| Location | | Quang Trung Software Park | | Scribe | CuongNC |
| Attendees | | BachLT, CuongNC | | | |
| Absent | |  | | | |
| Key Points Discussed | | | | | |
| No. | Topic | | Highlights | | |
| 1. | Discuss about report 5 | |  | | |
| 2. | Discuss about the Guide Implementation | |  | | |
| Action Plan | | | | | |
| No. | Action Item(s) | | PIC | | Target Date |
| **1.** | Complete report 5 | | CuongNC  BachLT | |  |
| **2.** | Coding I2C protocol | | BachLT | |  |
| **3.** | Complete Receiver Part Code | | CuongNC | |  |

#### Weekly Meeting – Nov 19th 2013

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Subject | | Weekly meeting 11 | | Date | Nov 19, 2013 |
| Facilitator | | PhucNT | | Time | 7:45 – 8:30 |
| Location | | Quang Trung Software Park | | Scribe | CuongNC |
| Attendees | | BachLT, CuongNC | | | |
| Absent | |  | | | |
| Key Points Discussed | | | | | |
| No. | Topic | | Highlights | | |
| 1. | Discuss about coding | |  | | |
| 2. | Discuss about the Guide Implementation | |  | | |
| Action Plan | | | | | |
| No. | Action Item(s) | | PIC | | Target Date |
| **1.** | Complete report 5 | | CuongNC  BachLT | |  |
| **2.** | Coding I2C protocol | | BachLT | |  |
| **3.** | Coding touchpad code | | BachLT | |  |

#### Weekly Meeting – Nov 26th 2013

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Subject | | Weekly meeting 12 | | Date | Nov 26, 2013 |
| Facilitator | | PhucNT | | Time | 7:45 – 8:30 |
| Location | | Quang Trung Software Park | | Scribe | CuongNC |
| Attendees | | BachLT, CuongNC | | | |
| Absent | |  | | | |
| Key Points Discussed | | | | | |
| No. | Topic | | Highlights | | |
| 1. | Discuss about report 5 | |  | | |
| 2. | Testing | |  | | |
| Action Plan | | | | | |
| No. | Action Item(s) | | PIC | | Target Date |
| **1.** | Create report 5 | | CuongNC  BachLT | |  |
| **2.** | Test Transmitter part | | BachLT | |  |
| **3.** | Test Receiver part | | CuongNC | |  |

#### Weekly Meeting – Dec 3rd 2013

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Subject | | Weekly meeting 13 | | Date | Nov 26, 2013 |
| Facilitator | | PhucNT | | Time | 7:45 – 8:30 |
| Location | | Quang Trung Software Park | | Scribe | CuongNC |
| Attendees | | BachLT, CuongNC | | | |
| Absent | |  | | | |
| Key Points Discussed | | | | | |
| No. | Topic | | Highlights | | |
| 1. | Review All document | |  | | |
| 2. | Complete report 6 | |  | | |
| Action Plan | | | | | |
| No. | Action Item(s) | | PIC | | Target Date |
| **1.** | Create report 6 | | CuongNC  BachLT | |  |
| **2.** | Review all document | | CuongNC  BachLT | |  |

## Coding Convention

Here is our coding convention: <http://www.jetcafe.org/jim/c-style.html>

## Other Material

# SOFTWARE REQUIREMENTS SPECIFICATIONS (SRS)

## User Requirement Specification

When technology becomes a part of our life which are so important, and there has been a ton of improvement in human’s life lately, therefore, a lot of device has been created especially some top-up box or other devices need to be set up in a separate place, and to be controlled in a certain range, we need a pointer device, but it can’t be something with too many wires and make everything become a chaos. With that thought, we intend to create a pointer device – touchpad which can work wirelessly, bring joy to user’s experience.

* Surface for sensing
* Design and implement the wireless touchpad can work on any kinds of OS (MAC, Ubuntu, Windows,…)
* Use TI MSP430F6659 in receiver part
* Use TI MSP430G2553 in transmitter part
* Receiver use USB HID receiver
* Transmitter is transmit to receiver through Wireless RF 2.4GHz (Nordic nRF24L01 +module)
* nRF24L01 +module is communicate with MSP430G2553 by SPI protocol
* Touchpad use 2AA battery (5.0V0 to supply the power. Beside, receiver is supplied by power from USB port

## System Requirement Specification (Specific Requirements)

### External Interface Requirement

#### User Interface

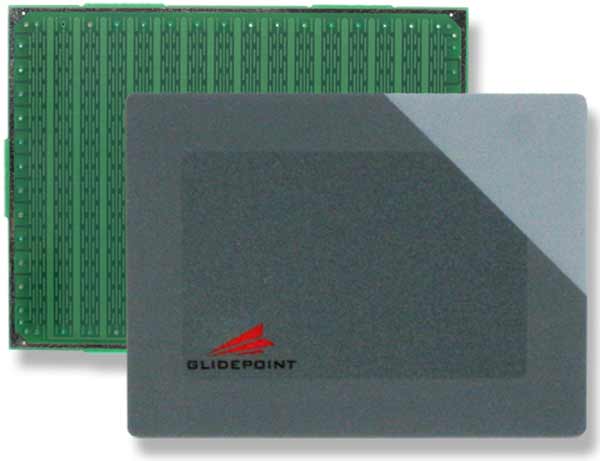


Figure 3 – Touchpad Interface

The user interface must be designed to be satisfied the following requirements:

* User Interaction part must be a touch sensor device
* Large enough that user can be able to interact

#### Hardware Interfaces

System contains 2 part: Receiver part and transmitter part

Receiver part include:

* A board which has a USB port
* MSP430F6659 microchip on the board
* nRF24L01

Transmitter part include:

* A board which has MSP430G2553 microchip on it
* nRF24L01
* Touchpad TM5957

#### Software Interfaces

* USB HID. Refer to:
* [www.usb.org/developers/devclass\_docs/HID1\_11.pdf](http://www.usb.org/developers/devclass_docs/HID1_11.pdf)
* <http://www.usb.org/developers/devclass_docs/Hut1_12v2.pdf>
* Operating System: Windows (XP, Vista, 7 or 8 are supported), Linux or Mac OS

#### Communication Protocol

Transmitter part:

* Between Touchpad and MSP430G2553: SPI
* Between MSP430G2553 and RF: SPI
* Between RF sender and receiver: RF wireless

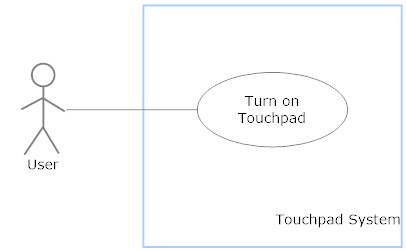
Receiver part:

* Between RF and MSP430G2553: SPI
* Between USB Hub and Computer: USB HID

### Functional Requirement Specification

#### Turn ON Touchpad

* Use Case-01 Diagram

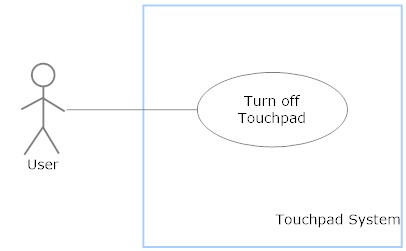


* Use Case-01 Specification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE SPECIFICATION** | | | |
| **Use-case No.** | UC001 | **Use-case Version** | 0.1 |
| **Use-case Name** | Turn ON Touchpad | | |
| **Author** | CuongNC | | |
| **Date** | 17/10/2013 | **Priority** | High |
| **Actor:**  User  **Summary:**  Turn ON Touchpad  **Goal:**  User turn ON Touchpad to use  **Triggers:**  When use press Power switch on touchpad  **Precondition:**   * PIN is setup into touchpad * Enough battery life to use   **Post condition:**   * **Success:** Touchpad is ON * **Failure:** Touchpad is not ON   **Main Success Scenario**   |  |  | | --- | --- | | Actor Actions | System Response | | 1. Press POWER switch in Touchpad | 1. Touchpad is ON |   **Alternative Scenario:**  **Exception:**  **Relationship:**  **Business Rules:** | | | |

#### Turn OFF Touchpad

* Use Case-02 Diagram

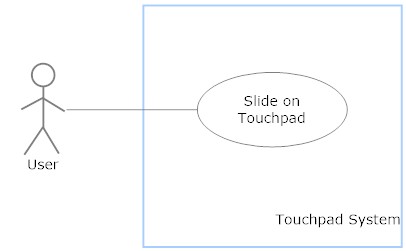


* Use Case-02 Specification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE SPECIFICATION** | | | |
| **Use-case No.** | UC002 | **Use-case Version** | 0.1 |
| **Use-case Name** | Turn ON Touchpad | | |
| **Author** | CuongNC | | |
| **Date** | 17/10/2013 | **Priority** | High |
| **Actor:**  User  **Summary:**  Turn OFF Touchpad  **Goal:**  User turn OFF Touchpad to use  **Triggers:**  When use press Power switch on touchpad  **Precondition:**   * Touchpad is OFF   **Post condition:**   * **Success:** Touchpad is OFF * **Failure:** Touchpad is ON   **Main Success Scenario**   |  |  | | --- | --- | | Actor Actions | System Response | | 1. Press POWER switch in Touchpad | 1. Touchpad is OFF |   **Alternative Scenario:**  **Exception:**  **Relationship:**  **Business Rules:** | | | |

#### Slide on Touchpad

* Use Case-03 Diagram

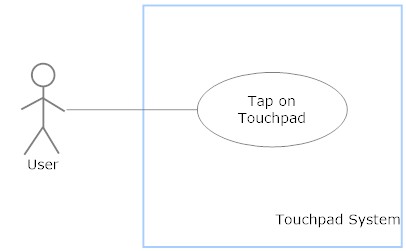


* Use Case-03 Specification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE SPECIFICATION** | | | |
| **Use-case No.** | UC003 | **Use-case Version** | 0.1 |
| **Use-case Name** | Slide on Touchpad | | |
| **Author** | CuongNC | | |
| **Date** | 17/10/2013 | **Priority** | High |
| **Actor:**  User  **Summary:**  Move the pointer as user action  **Goal:**  User can be able to move the pointer on screen  **Triggers:**  When user tap on the touchpad and hold it longer than 0.3s then move  **Precondition:**   * Touchpad is ON * Receiver is plugged in terminal device   **Post condition:**   * **Success:** Pointer moves as user’s action * **Failure:** Pointer moves wrong or doesn’t move at all   **Main Success Scenario**   |  |  | | --- | --- | | Actor Actions | System Response | | 1. Slide on Touchpad | 1. Pointer moves as the slide goes |   **Alternative Scenario:**  **Exception:**  **Relationship:**  **Business Rules:** | | | |

#### Tap on Touchpad

* Use Case-04 Diagram

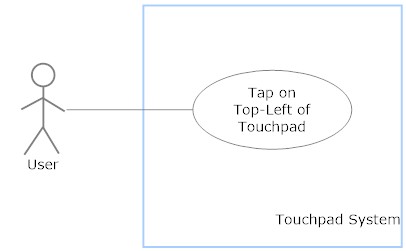


* Use Case-04 Specification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE SPECIFICATION** | | | |
| **Use-case No.** | UC004 | **Use-case Version** | 0.1 |
| **Use-case Name** | Tap on Touchpad | | |
| **Author** | CuongNC | | |
| **Date** | 17/10/2013 | **Priority** | High |
| **Actor:**  User  **Summary:**  Left click mouse action when user tap on touchpad  **Goal:**  User can be able to use left click mouse  **Triggers:**  User tap on the touchpad  **Precondition:**   * Touchpad is ON * Receiver is plugged in terminal device   **Post condition:**   * **Success:** The pointer have left-click action * **Failure:** The pointer doesn’t do anything at all or acting wrong action   **Main Success Scenario**   |  |  | | --- | --- | | Actor Actions | System Response | | 1. Tap on the touchpad | 1. Pointer do left-click action |   **Alternative Scenario:**  **Exception:**  **Relationship:**  **Business Rules:** | | | |

#### Tap on Top-Left of Touchpad

* Use Case-05 Diagram

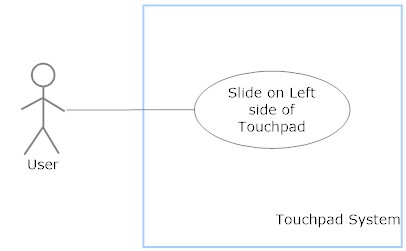


* Use Case-05 Specification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE SPECIFICATION** | | | |
| **Use-case No.** | UC005 | **Use-case Version** | 0.1 |
| **Use-case Name** | Tap on Top-Left of Touchpad | | |
| **Author** | CuongNC | | |
| **Date** | 17/10/2013 | **Priority** | High |
| **Actor:**  User  **Summary:**  Right click mouse action when user tap on Top-Left of touchpad  **Goal:**  User can be able to use right click mouse  **Triggers:**  User tap on Top-Left of touchpad  **Precondition:**   * Touchpad is ON * Receiver is plugged in terminal device   **Post condition:**   * **Success:** The pointer have right-click action * **Failure:** The pointer doesn’t do anything at all or acting wrong action   **Main Success Scenario**   |  |  | | --- | --- | | Actor Actions | System Response | | 1. User tap on Top-Left of touchpad | 1. Pointer do right-click action |   **Alternative Scenario:**  **Exception:**  **Relationship:**  **Business Rules:** | | | |

#### Slide on the Left side of Touchpad

* Use Case-06 Diagram

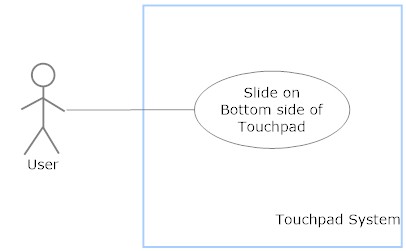


* Use Case-06 Specification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE SPECIFICATION** | | | |
| **Use-case No.** | UC006 | **Use-case Version** | 0.1 |
| **Use-case Name** | Slide on Left Side of Touchpad | | |
| **Author** | CuongNC | | |
| **Date** | 17/10/2013 | **Priority** | High |
| **Actor:**  User  **Summary:**  Vertical scroll when user slide along the Left side of Touchpad  **Goal:**  User can be able to scroll vertically  **Triggers:**  User slide along the Left side of Touchpad  **Precondition:**   * Touchpad is ON * Receiver is plugged in terminal device   **Post condition:**   * **Success:** The pointer have vertical scroll action * **Failure:** The pointer doesn’t do anything at all or acting wrong action   **Main Success Scenario**   |  |  | | --- | --- | | Actor Actions | System Response | | 1. User slide along the Left side of Touchpad | 1. The pointer have vertical scroll action |   **Alternative Scenario:**  **Exception:**  **Relationship:**  **Business Rules:** | | | |

#### Slide on the Bottom side of Touchpad

* Use Case-07 Diagram

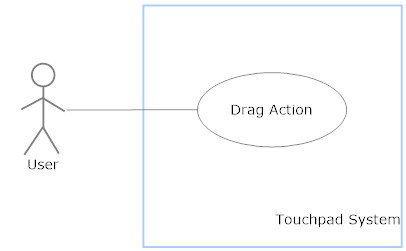


* Use Case-07 Specification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE SPECIFICATION** | | | |
| **Use-case No.** | UC007 | **Use-case Version** | 0.1 |
| **Use-case Name** | Slide on Bottom side of Touchpad | | |
| **Author** | CuongNC | | |
| **Date** | 17/10/2013 | **Priority** | High |
| **Actor:**  User  **Summary:**  Horizontal scroll when user slide along the Bottom side of Touchpad  **Goal:**  User can be able to scroll Horizontally  **Triggers:**  User slide along the Bottom side of Touchpad  **Precondition:**   * Touchpad is ON * Receiver is plugged in terminal device   **Post condition:**   * **Success:** The pointer have horizontal scroll action * **Failure:** The pointer doesn’t do anything at all or acting wrong action   **Main Success Scenario**   |  |  | | --- | --- | | Actor Actions | System Response | | 1. User slide along the Bottom side of Touchpad | 1. The pointer have horizontal scroll action |   **Alternative Scenario:**  **Exception:**  **Relationship:**  **Business Rules:** | | | |

#### Drag Action

* Use Case-08 Diagram



* Use Case-08 Specification

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE SPECIFICATION** | | | |
| **Use-case No.** | UC008 | **Use-case Version** | 0.1 |
| **Use-case Name** | Drag Action | | |
| **Author** | CuongNC | | |
| **Date** | 17/10/2013 | **Priority** | High |
| **Actor:**  User  **Summary:**  Horizontal scroll when user slide along the Bottom side of Touchpad  **Goal:**  User can be able to scroll Horizontally  **Triggers:**  User tap on touchpad then immediately slide  **Precondition:**   * Touchpad is ON * Receiver is plugged in terminal device   **Post condition:**   * **Success:** The pointer have drag action * **Failure:** The pointer doesn’t do anything at all or acting wrong action   **Main Success Scenario**   |  |  | | --- | --- | | Actor Actions | System Response | | 1. User tap on touchpad 2. Then immediately slide on it | 1. The pointer have drag action |   **Alternative Scenario:**  **Exception:**  **Relationship:**  **Business Rules:** | | | |

### Non-Functional Requirement Specification

#### Reliability

* Touchpad can work 24/7
* Touchpad don’t damage any hardware of PC or harm to users while it is working

#### Availability

* Touchpad have “Plug and Play” feature. So that user just connect receiver into USB port and use it. Nothing else is required to be done.

#### Security

* Communicate packet between transmitter and receiver is secured

#### Maintainability

* Transmitter use 2 AAA batteries to bring comfortable and cheapness, besides it’s easy to replace

#### Portability

* Can be used by any devices which support USB HID
* Doesn’t require any software or driver to be installed

#### Performance

* Touchpad can work as well with any kind off OS (Windows, Mac, Ubuntu,..)

## Other Material (if any)

# SOFTWARE DESIGN SPECIFICATION (SDS)

## Design Overview

* This system is divided into 2 small-systems: transmitter and receiver
* Transmitter and receiver system are connected and communicate by 2.4 GHz RF.

## System Architectural Design

Diagram about communicate protocol has been used in system

MicrocontrollerUnit

TouchPad

MicrocontrollerUnit

I2C

VCC

GND

SPI

SPI

Receiver

NRF24L01-IA

Transmitter

NRF24L01-IA

MicrocontrollerUnit

USB Port

Another Point of View how the system work:

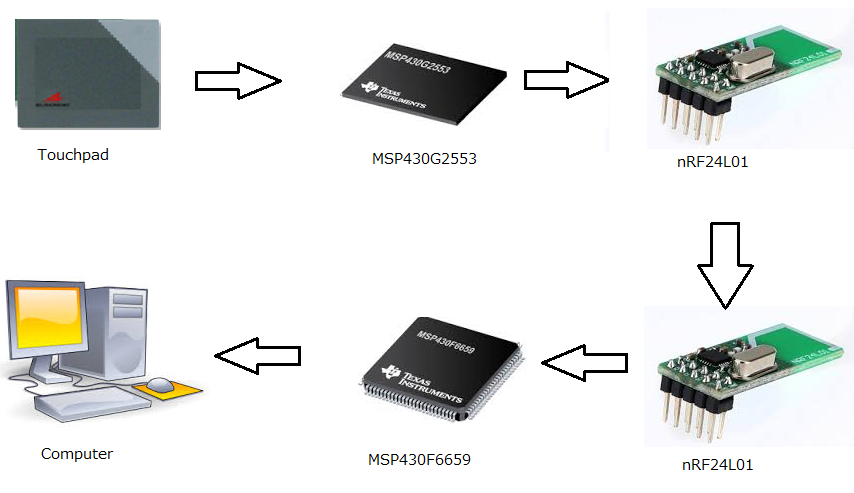


Figure 4 – Overview System Process

**Description**:

* After user’s interaction to the touchpad, a set of ordinates will be retrieved from register by MSP430G2553 chip. The retrieved action from MSP430G2553 will be processed to touchpad through I2C Protocol
* Then the MSP430G2553 microchip will process a certain action to divide user’s interaction into tap action, or move action. And then base on the coordinate, it’ll return an exact interaction which has been executed by user.
* From this exact action the chip has been recognized, through SPI protocol, MSP430G2553 microchip will send data to nRF24L01 for transmitting it to receiver part.
* On a specified channel and addresses, nRF24L01 will be able to communicate to each other, and data will be stored on the receiver part’s nRF24L01’s pipe
* At the time data’s been received, on the receiver part, the microchip – MSP430F6659 will be awaked, and receive the data has been sent.
* After some certain processes, a specified report will be created and MSP430F6659 will send it to the computer through USB HID.
* After this action, the process of interacting to pointer will completely depend on the OS’s been running on the device.

### Device

#### Microcontroller – MSP430G2553

* Overview:

The Texas Instruments MSP430 family of ultra-low-power microcontrollers consists of several devices featuring different sets of peripherals targeted for various applications. The architecture, combined with five low-power modes, is optimized to achieve extended battery life in portable measurement applications. The device features a powerful 16-bit RISC CPU, 16-bit registers, and constant generators that contribute to maximum code efficiency.

The digitally controlled oscillator (DCO) allows wake-up from low-power modes to active mode in less than 1 µs. The MSP430G2x13 and MSP430G2x53 series are ultra-low-power mixed signal microcontrollers with built-in 16-bit timers, up to 24 I/O capacitive-touch enabled pins, a versatile analog comparator, and built-in communication capability using the universal serial communication interface. In addition the MSP430G2x53 family members have a 10-bit analog-to-digital (A/D) converter.

Typical applications include low-cost sensor systems that capture analog signals, convert them to digital values, and then process the data for display or for transmission to a host system

|  |  |  |
| --- | --- | --- |
| Device Name | Requirements | Others |
| MSP430G2553 | * Low Supply-Voltage range: 1.8V to 3.6V * Ultra-Low Power Consumption * Support I2C and SPI protocol | * 20 input – output Endpoints |

Table 1 – MSP430G2553 Microcontroller Requirements



Figure 5 – MSP430G2553 Chip

* Features:
* Watchdog timer: The primary function of the watchdog timer (WDT+) module is to perform a controlled system restart after a software problem occurs. If the selected time interval expires, a system reset is generated. If the watchdog function is not needed in an application, the module can be disabled or configured as an interval timer and can generate interrupts at selected time intervals.
* Universal Serial Communications Interface (USCI): The USCI module is used for serial data communication. The USCI module supports synchronous communication protocols such as SPI (3 or 4 pin) and I2C, and asynchronous communication protocols such as UART, enhanced UART with automatic baudrate detection (LIN), and IrDA. Not all packages support the USCI functionality. USCI\_A0 provides support for SPI (3 or 4 pin), UART, enhanced UART, and IrDA. USCI\_B0 provides support for SPI (3 or 4 pin) and I2C.
* Low power mode: The MSP430 has one active mode and five software selectable low-power modes of operation. An interrupt event can wake up the device from any of the low-power modes, service the request, and restore back to the low-power mode on return from the interrupt program.

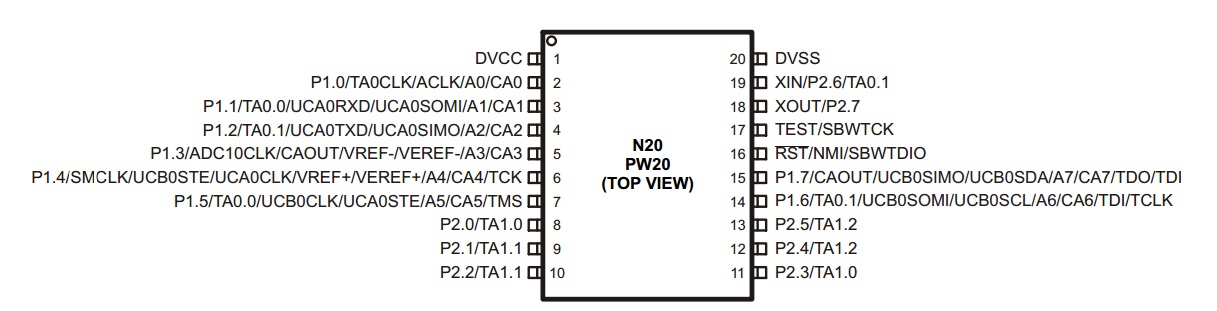
******

Figure 6 – MSP430G2553 PIN Diagram

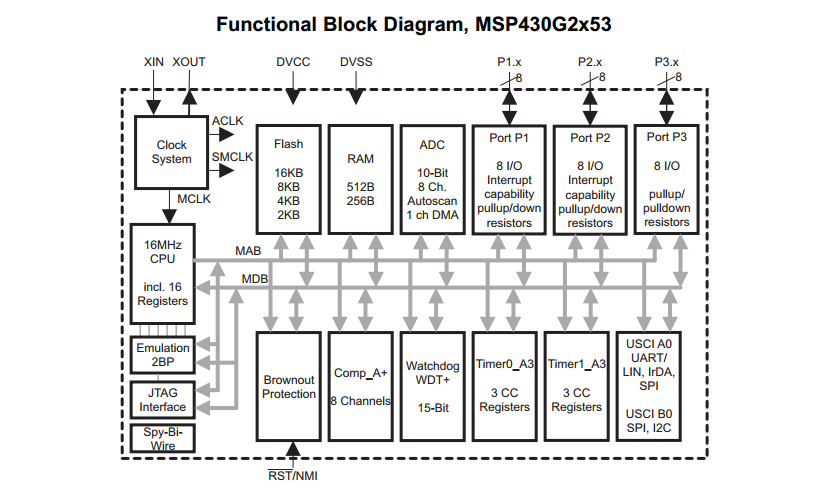


Figure 7 - MSP430G2553 Functional Block Diagram

#### Microcontroller – MSP430F6659

* Overview:

The Texas Instruments MSP430™ family of ultra-low-power microcontrollers consists of several devices that feature different sets of peripherals targeted for various applications. The architecture, combined with five low power modes, is optimized to achieve extended battery life in portable measurement applications. The device features a powerful 16-bit RISC CPU, 16-bit registers, and constant generators that contribute to maximum code efficiency. The digitally controlled oscillator (DCO) allows the device to wake up from low-power modes to active mode in 3 µs (typical).

The MSP430F665x and MSP430F565x series are microcontroller configurations with four 16-bit timers, a high performance 12-bit analog-to-digital converter (ADC) , three universal serial communication interfaces (USCIs), a hardware multiplier, DMA, a real-time clock (RTC) module with alarm capabilities, a comparator, USB 2.0, and up to 74 I/O pins.

Typical applications for these devices include analog and digital sensor systems, digital motor control, remote controls, thermostats, digital timers, and hand-held meters

|  |  |  |
| --- | --- | --- |
| Device Name | Requirements | Others |
| MSP430F6659 | * Low Supply Voltage Range: 3.6 V Down to 1.8 V * Ultra-Low Power Consumption * I2CTM * Synchronous SPI | * Easy to purchase * Low cost (free sample) * Full-Speed Universal Serial Bus (USB) * Eight Input and Eight Output Endpoints * Integrated 3.3-V and 1.8-V USB Power |

Table 2 – MSP430G2553 Microcontroller Requirements



Figure 8 – MSP430F6659 Chip

* Features:
* Watchdog timer: The primary function of the watchdog timer (WDT+) module is to perform a controlled system restart after a software problem occurs. If the selected time interval expires, a system reset is generated. If the watchdog function is not needed in an application, the module can be disabled or configured as an interval timer and can generate interrupts at selected time intervals.
* Universal Serial Communications Interface (USCI): The USCI module is used for serial data communication. The USCI module supports synchronous communication protocols such as SPI (3 or 4 pin) and I2C, and asynchronous communication protocols such as UART, enhanced UART with automatic baudrate detection (LIN), and IrDA. Not all packages support the USCI functionality. USCI\_A0 provides support for SPI (3 or 4 pin), UART, enhanced UART, and IrDA. USCI\_B0 provides support for SPI (3 or 4 pin) and I2C.
  + Low power mode: The MSP430 has one active mode and five software selectable low-power modes of operation. An interrupt event can wake up the device from any of the low-power modes, service the request, and restore back to the low-power mode on return from the interrupt program.
  + USB Universal Serial Bus: The USB module is a fully integrated USB interface that is compliant with the USB 2.0 specification. The module supports full-speed operation of control, interrupt, and bulk transfers. The module includes an integrated LDO, PHY, and PLL. The PLL is highly flexible and can support a wide range of input clock frequencies. When USB RAM is not used for USB communication, it can be used by the system

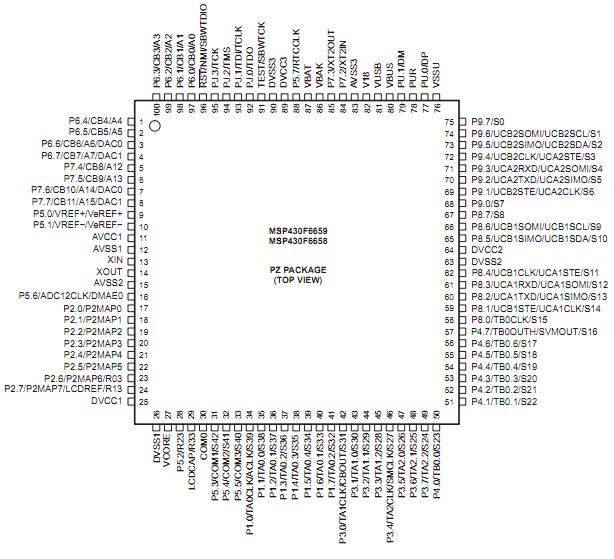
**

Figure 9 – MSP430F6659 PIN Diagram

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter Name | Value | Minimum requirements | Status |
| Architecture | 16-bit | 8/16 bit | OK |
| Frequency (MHz) | 20 |  |  |
| Flash (KB) | 512 |  |  |
| SRAM (kB) | 66 |  |  |
| LCD Segments | 160 |  |  |
| Watchdog | Yes | Yes | OK |
| Real-Time Clock | Yes | Yes | Ok |
| Operating Voltage Range (V) | 1.8 to 3.6 | Work at 3.3V | OK |
| I/O Pins | 74 | Flexible in configuration | OK |
| Capture/Compare/PWM  Peripherals | 4/4 | 2-PWM outputs | OK |
| Timers | 4 x 16-bit | 1 Timer | OK |
| Cost | Free sample | Low cost | OK |
| USB | 1 | 1 | OK |
| USCI\_A (UART/LIN/IrDA/SPI) | 3 |  | OK |
| USCI\_B (I2C & SPI) | 3 |  | OK |
| DMA | 6 |  | OK |
| Multiplier | 32x32 |  | OK |
| Comparators | 1 |  | OK |
| Temp Sensor | Yes |  | OK |
| Power Management | LDO, SVS/SVM, BOR |  | OK |
| ADC | 12-bit SAR |  | OK |
| ADC Channels | 16 |  | OK |
| DAC | 2 |  | OK |
| End Equipment Optimized | - |  | OK |

Table 3 – MSP430F6659 Features

#### Nordic SOC Module

* Overview:

The nRF24L01 is a single chip 2.4GHz transceiver with an embedded baseband protocol engine (Enhanced ShockBurst™), designed for ultralow power wireless applications. The nRF24L01 is designed for operation in the world wide ISM frequency band at 2.400 - 2.4835GHz. An MCU (microcontroller) and very few external passive components are needed to design a radio system with the nRF24L01.

The nRF24L01 is configured and operated through a Serial Peripheral Interface (SPI.) Through this interface the register map is available. The register map contains all configuration registers in the nRF24L01 and is accessible in all operation modes of the chip.

The embedded baseband protocol engine (Enhanced ShockBurst™) is based on packet communication and supports various modes from manual operation to advanced autonomous protocol operation. Internal FIFOs ensure a smooth data flow between the radio front end and the system’s MCU. Enhanced ShockBurst™ reduces system cost by handling all the high-speed link layer operations.

The radio front end uses GFSK modulation. It has user configurable parameters like frequency channel, output power and air data rate.

The air data rate supported by the nRF24L01 is configurable to 2Mbps. The high air data rate combined with two power saving modes makes the nRF24L01 very suitable for ultralow power designs.

Internal voltage regulators ensure a high Power Supply Rejection Ratio (PSRR) and a wide power supply range.

* Features:
* Worldwide 2.4GHz ISM band operation
* 126 RF channels
* Common RX and TX pins
* GFSK modulation
* 1 and 2Mbps air data rate
* 1MHz non-overlapping channel spacing at 1Mbps
* 2MHz non-overlapping channel spacing at 2Mbps
* Enhanced ShockBurstTM

|  |  |  |
| --- | --- | --- |
| Device Name | Requirements | Others |
| NRF24L01-IA | * Worldwide 2.4GHz ISM band operation * Up to 2Mbps on air data rate * Ultra low power operation * On chip voltage regulator 1.9 to 3.6V supply range | * Air compatible with nRF2401A, 02, E1 and E2 * Low cost BOM * ±60ppm 16MHz crystal * 5V tolerant inputs |

Table 4 – nRF24L01 Requirements

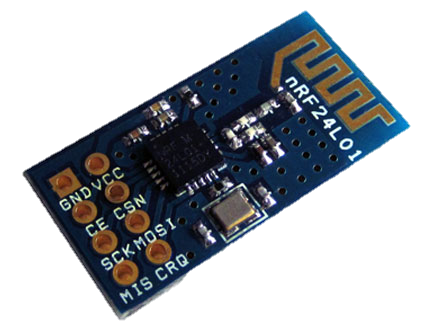


Figure 10 – nRF24L01

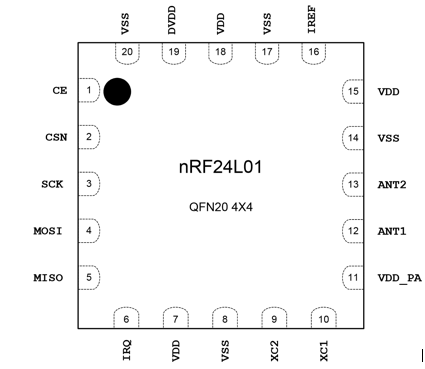


Figure 11 – nRF24L01 PIN Diagram

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Symbol | Parameter (condition) | Notes | Min. | Typ. | Max. | Units |
| PRF | Maximum Output Power | a |  | 0 | +4 | dBm |
| PRFC | RF Power Control Range |  | 16 | 18 | 20 | dB |
| PRFCR | RF Power Accuracy |  |  |  | ±4 | dB |
| PBW2 | 20dB Bandwidth for Modulated Carrier (2Mbps) |  |  | 1800 | 2000 | kHz |
| PBW1 | 20dB Bandwidth for Modulated Carrier (1Mbps) |  |  | 900 | 1000 | kHz |
| PRF1 | 1st Adjacent Channel Transmit Power 2MHz |  |  |  | -20 | dBm |
| PRF2 | 2nd Adjacent Channel Transmit Power 4MHz |  |  |  | -50 | dBm |

Table 5 – Transmitter Operation

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Symbol | Parameter (condition) | Notes | Min. | Typ. | Max. | Units |
| RXmax | Maximum received signal at <0.1% BER |  |  | 0 |  | dBm |
| RXSENS | Sensitivity (0.1%BER) @2Mbps |  |  | -82 |  | dBm |
| RXSENS | Sensitivity at (0.1%BER) @1Mbps |  |  | -85 |  | dBm |
| RX selectivity with nRF24L01 equal modulation on interfering signal | | | | |  |  |
| C/ICO | C/I Co-channel (@2Mbps) (Modulated carrier) | a |  | 11 |  | dB |
| C/I1ST | 1st Adjacent Channel Selectivity C/I 2MHz |  |  | 4 |  | dB |
| C/I2ND | 2nd Adjacent Channel Selectivity C/I 4MHz |  |  | -20 |  | dB |
| C/I3RD | 3rd Adjacent Channel Selectivity C/I 6MHz |  |  | -27 |  | dB |
| C/ICO | C/I Co-channel (@1Mbps) | b |  | 12 |  | dB |
| C/I1ST | 1st Adjacent Channel Selectivity C/I 1MHz |  |  | 8 |  | dB |
| C/I2ND | 2nd Adjacent Channel Selectivity C/I 2MHz |  |  | -21 |  | dB |
| C/I3RD | 3rd Adjacent Channel Selectivity C/I 3MHz |  |  | -30 |  | dB |

Table 6 – Receiver Operation

#### Touchpad TM5957

* Overview:

The Model TM5957 touchpad is designed for use as an embedded Serial Peripheral Interface (SPI) or Inter-Integrated Circuit (I2C) device. The dimensions of the TM5957 are ideal for integration into a wide variety of products where low power is required and space is a constraint.

* Benefits
* Superior Moisture Performance
* No Moving Parts – High Reliability – Durable
* Extremely Low Power Idle Mode
* Precise Positioning
* Environmentally Sealable Design
* Compact Size, Thin Profile, and Lightweight
* Low Stress / Low Fatigue Operation
* No Contact Pressure Required

Cirque Glide Point® touchpads provide smooth, effortless, tracking based on a patented electrical capacitance system. Beneath its durable surface is a sophisticated sensing array that accurately responds to even the most minute finger movements. No finger pressure is required. Simply move a finger across the touchpad or create a combination of taps and drags to establish the desired functionality.

The TM5957 incorporates the latest advancements in GlidePoint technology. Cirque touchpads have superior moisture performance, unsurpassed by other capacitance-based touchpads. A power-saving idle mode reduces current draw during periods of inactivity on the touchpad surface.

Cirque’s latest generation of touchpads defines a new standard in the touch sensor industry. These extremely durable, solid-state devices have no moving parts to breakdown, and never require disassembly for cleaning. Their compact size, thin profile and low weight allow them to be readily designed into a variety of OEM products, where they may be environmentally sealed if desired. Cirque touchpads, using patented GlidePoint technology, are the optimum pointing device for OEM integration

* General Specifications
* Operational Specifications
* Position Detection Method: Mutual capacitance sensing
* X/Y Position Sensing Resolution: Up to 40 counts/mm
* X/Y Position Reporting: Absolute
* Tracking Speed: Up to 1,016 mm/sec
* Touch Force: No contact pressure required
* Lifetime (Plastic Overlay): Minimum 10,000,000 strokes (500 km)
* Sample Rate: Up to 100 samples/sec.
* Electrical Specifications
* Power Supply Voltage: 3.0 V +15% (5.0 V +10% Optional)
* Environmental Specifications
* Operating Temperature: -40o to 85oC (as measured at component side)
* Operating Humidity: 5% to 95% relative humidity (non-condensing)
* Storage Temperature: -55o to + 150 oC
* ESD (applied to sensing surface): 15 kV when module is properly installed
* Interface Specifications
* Communication Protocol: Inter-Integrated Circuit (I2C)
* Mechanical Buttons: Up to three (3) mechanical switches
* Physical Specifications
* Refer to Figure 4 for a detailed drawing of the TM5957 module
* Overall Module Thickness: 1.90 + 0.25 mm (PCB + Overlay) 4.70 mm Max 0.25 mm (PCB + Overlay + Components)
* Module Width (including 2X tabs): 65.80 + 0.25 mm
* Module Height (including 4X tabs): 49.80 + 0.25 mm
* Module Weight: 14.0 grams
* Maximum Bezel Opening: 61.80 x 45.80 mm
* Interface Specifications

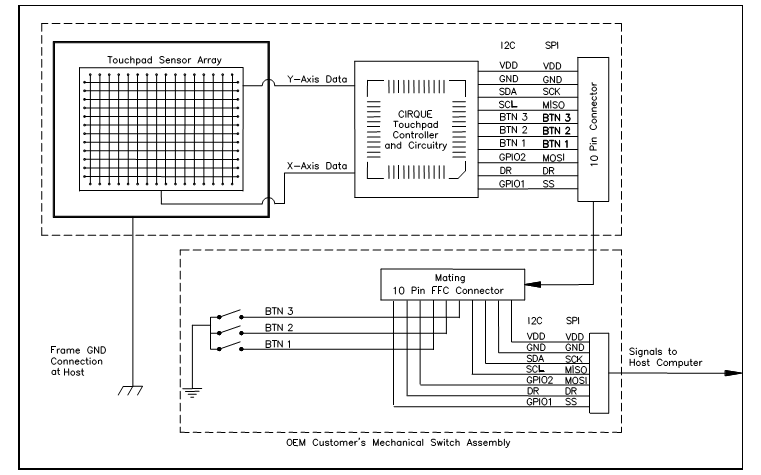


Figure 12 – TM5957 System Interface Block Diagram

* Physical Dimensions

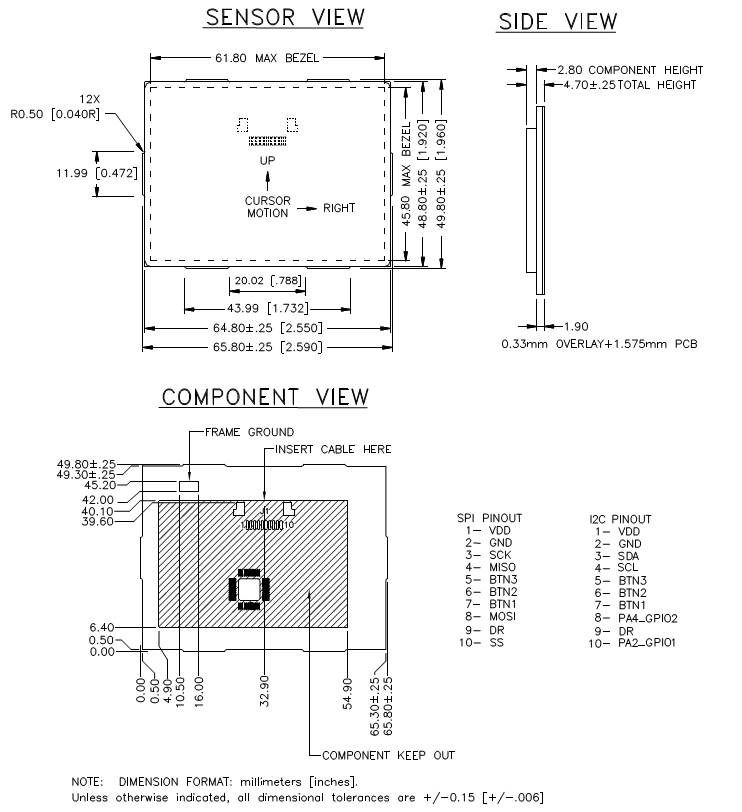


Figure 13 – Touchpad Module Design Dimensions

## Diagram

### Transmitter Overview Diagram



Figure 14 – Transmitter Overview Flowchart

Description: Transmitter part has been activated, then process will start by initializing ports, SPI communication to RF, I2C communication to Touchpad. Then the system will automatically return in to Low Power Mode and it will stay there until touchpad has been touched by user, the microcontroller MSP430G2553 will exit from Low Power Mode and get data from touchpad, after progressing data, it will be transfer to nRF24L01 for broadcasting to receiver. And then the microcontroller will be back at Low Power Mode.

### Receiver Overview Diagram



Figure 15 – Receiver Overview Flowchart

Description: After receiver has been plugged into PC, it will start at initialization step, after this step, PC will acknowledge receiver as a mouse device. Then microcontroller will return to Low Power Mode. Once data has been come to nRF24L01, it’ll awake microcontroller MSP430F6659, it will exit from Low Power Mode and progress received data, and send it to computer via USB HID. Then microcontroller will be back at Low Power Mode

## Software Design

### Overview System Sequence Diagram



Figure 21 – Overview System Sequence Diagram

Description: When user touch down the touchpad, the flag hasData (data ready HW\_DR, 9th PIN in touchpad) is set to 1.

In transmitter, the timer periodic check hasData flag. If this flag is not set, transmitter continue sleeping, else if this flag is set, transmitter will wake up (exit the low power mode) to process data.

Transmitter will send command to read data (0xA0 | address) via I2C communication. Transmitter will receive data and status variance to check the transmission. If the status is false (0x0A), transmitter will send read again command (0xB0) to read data, else status is true (0x0E) transmitter will send another read data command to read another byte or stop if last byte.

After receive 1 state (8 bytes) transmitter will process to find what user do (move mouse, click, scroll …) then create appropriate data and send to the transmitter nRF24L01 via SPI communication. After finish send data to nRF24L01 transmitter will sleep (enable LPM low power mode) and wait to another hasData flag.

When receive full 1 packet (5bytes), the transmitter’s nRF24L01 will send one by one data to the receiver’s nRF24L01. After sending 1 byte, CRC code will be response back to the transmitter to checking the transmission. If the CRC code is not correct, transmitter will send that byte again.

When the receiver’s nRF24L01 received full 1 packet, flag hasData (IRQ flag, 8th PIN in nRF24L01) is set to 1.

In receiver, the timer periodic check hasData flag. If this flag is not set, receiver continue sleeping, else if this flag is set, receiver will wake up to receive data.

Receiver send command (0x70) via SPI communication to read data from nRF24L01 then receive data and status flag from nRF24L01. If the status is false, receiver will send again command to read data, else status is true receiver will send another read data command to read another byte or stop if last byte.

After receiver all data, receiver will send data to computer via USB HID to change mouse position in computer.

### Flow Chart RF

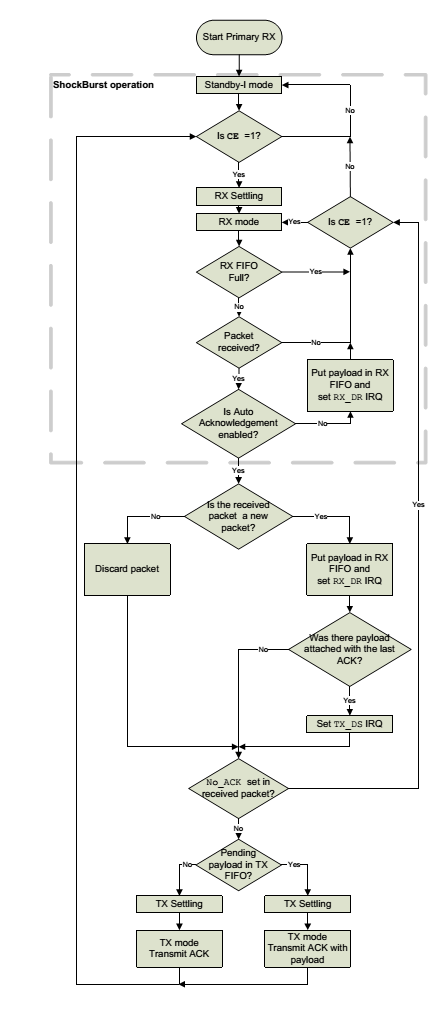


Figure 22 – RF working process flowchart

### Transmitter Sequence Diagram

#### Left-Click Action



Figure 23 – Left-Click Action Sequence Diagram

#### Right-Click Action



Figure 24 – Right Click Action Sequence Diagram

|  |  |
| --- | --- |
| Step | Addition |
| 2 Get First State | * If the first coordinate is recognized at Top-Left of the touch pad, the send last state will be known as a Right-Click action |

#### Move Action



Figure 25 – Move Action Sequence Diagram

#### Vertical Scroll Action



Figure 26 – Vertical Scroll Action Sequence Diagram

|  |  |
| --- | --- |
| Step | Addition |
| 2 Get First State | * If the first coordinate is recognized at Left side of the touch pad, the send last state will be known as a Vertical Scroll action |

### Receiver Sequence Diagram

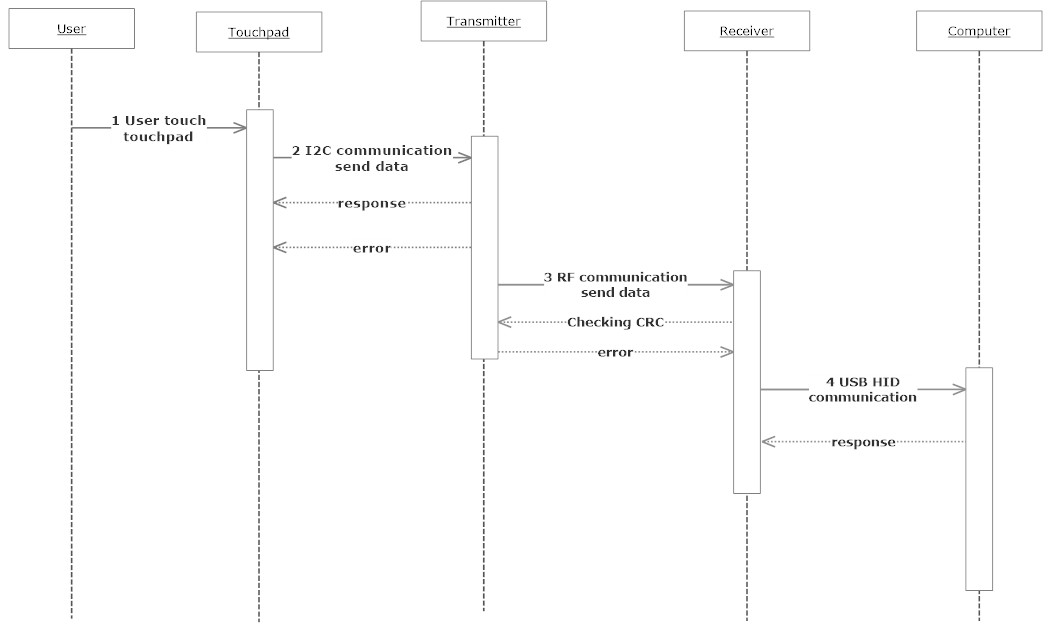


Figure 27 – Receiver Sequence Diagram

Description: After user’s iteration with touchpad, data from touchpad will be sent to transmitter part through I2C communication, once data has been arrived at transmitter, it will send back to touchpad acknowledge that the sending has been success or not.

After progressing data, transmitter will send to receiver through RF communication, and there will be an auto-checking CRC to ensure the data sending has been corrected.

Receiver will acknowledge the data and from that create a report to send back to PC through USB HID communication

# SYSTEM IMPLEMENTATION & TEST (SIT)

## Introduction

### Overview

This section includes all necessary information about test plan, test process, test approach, the environment for testing, test pass/fail criteria, test results and a checklist to check when test this system. It describes implementation details of test case or technical details of how the products features should work.

### Test Approach

Testing this project will include Unit test – White box testing, Black box testing and Integration testing, System testing. The process of testing will be declare as the following

* White box testing: Unit test – Developers test on code in which function they created
* Black box testing: Test on each modules’ function to ensure that it work probably right.
* Integration testing: Test if integrated functions make any conflict.
* System testing: Test the whole system

Each testing will be process after development phase and execute in group.

## Test Plan

In order to have a good testing, everything must be planned and have schedules. So that the main purpose of this phase is to list and verify the function in system that needed to test and plan for completing it.

### Features to be tested

Below are overview features that need to be tested

|  |  |  |  |
| --- | --- | --- | --- |
| Items | Test Description | Test Date | Responsibilities |
| Transmitter | Test Touchpad:   * Able to acknowledge tap action * Able to acknowledge slide action * Able to acknowledge slide-at-bottom-and-right-side-of-touchpad action * Able to acknowledge tap-on-the-top-left-of-touchpad action * Able to send coordinates |  | -BachLT |
| Test G2553 Chip’s I2C protocol |  | -BachLT |
| Test G2553 chip’s SPI protocol |  | -BachLT |
| Test RF communicate to G2553 |  | -BachLT |
| Test Touchpad code and RF code Integration |  | -BachLT |
| Receiver | Test USB HID code:   * Able to connect to PC * Able to send mouse click action * Able to send coordinates * Able to send vertical/horizontal coordinates * Able to acknowledge Drag Action * Able to acknowledge Double-Click Action |  | -CuongNC |
| Test F6659 chip’s SPI protocol |  | -CuongNC |
| Test RF communicate to F6659 |  | -CuongNC |
| Test USB HID code and RF code integration |  | -CuongNC |

Table 7 – Features to be Tested

### Features not to be tested

Software conflicts.

### Testing Tools and Environments

#### Environments

* Windows OS (Windows 7, Windows 8)
* MAC OS
* Ubuntu OS

#### Testing Tools

* Device Manager and USBDeview (Windows OS)

### Test Pass/Fail Criteria

* 90% of test cases must pass
* Test cases including critical function must pass
* All medium and high defects must be fixed
* Test coverage must be at least 90%

## Test Case

### Component Testing

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Test case description | Test case procedure | Expected result | Inter-test case dependence | Result | Test date | Creator | | Tester |
|  | Transmitter Module | | | | | | | | |
|  | * Communicate Protocol | | | | | | | | |
| CP01 | Test SPI’s initialization on MSP430G2553 is correct or not by wiring MISO-MOSI | 1. Wire MISO and MOSI 2. Using SPI initialization code and write a test code to send data through MOSI and receive sent data through MISO | 1. Success initializing SPI pins 2. Able to send data through MOSI and receive data through MISO |  | Passed | 13/11/13 | CuongNC | | BachLT |
| CP02 | Test SPI’s communication by using 2 MSP430G2553 chip | 1. Wire one chip’s MISO to another one’s MOSI and the same to 2 pins MOSI and MISO, wire 2 SLK pins 2. Using code test – making 1 chip is master and the other one is slave, send data from master chip and slave chip echo it back | 1. Success initializing SPI protocol 2. Check if master chip has been echoed back sent data | CP01 | Passed | 13/11/13 | CuongNC | | BachLT |
| CP03 | Test SPI protocol between MSP430G2553 and RF | 1. Wire 3 SPI pins between G2553 chip and nRF24L01 2. Using code test – Initialize SPI from G2553 chip and send command to RF | 1. Success in initializing SPI protocol 2. nRF24L01 acknowledge command sent from G2553 and echo back status (0x0E is success in communicate expected status) | CP02 | Passed | 13/11/13 | CuongNC | | BachLT |
| CP04 | Test I2C initialization and communication between 2 MSP430G2553 | 1. Wire 2 I2C pins between 2 G2553 chip 2. Using test code to initialize I2C and send data from 1 MSP430G2553 and the other one echo it back | 1. Success in initializing I2C protocol 2. Data send from 1 MSP430G2553 has been echoed back |  | Passed | 13/11/13 | | CuongNC | BachLT |
| CP05 | Test I2C communication between MSP430G2553 and touchpad TM5957 | 1. Wire 2 I2C pins between MSP430G2553 chip and TM5957 Touchpad 2. Using code test - Initialize I2C and send command to touchpad | 1. Success in initializing I2C protocol 2. Touchpad acknowledge command and send data back | CP04 | Passed | 13/11/13 | CuongNC | | BachLT |
|  | * Touchpad Communication | | | | | | | | |
| TC01 | Test touchpad initialization | 1. Wire I2C pins between MSP430G2553 and touchpad TM5957 2. Use test code – read STATUS register from touchpad TM5957 | 1. Success in initializing I2C protocol 2. Touchpad acknowledge command and send back STATUS register | CP05 | Passed | 13/11/13 | CuongNC | | BachLT |
| TC02 | Test touchpad code – acknowledge a tap action has been activated on the touchpad | 1. Wire I2C pins between MSP430G2553 and touchpad TM5957 2. Use test code – read register has coordinate of tap point 3. Tap on touchpad | 1. Success in initializing I2C protocol 2. Receive coordinate of tap point | TC02 | Passed | 13/11/13 | CuongNC | | BachLT |
| TC03 | Test touchpad code – acknowledge a slide action has been activated on the touchpad | 1. Wire I2C pins between MSP430G2553 and touchpad TM5957 2. Use test code – read register has coordinates of points 3. Slide on touchpad | 1. Success in initializing I2C protocol 2. Retrieve coordinates of slide points | TC02 | Passed | 13/11/13 | CuongNC | | BachLT |
| TC04 | Test touchpad code – acknowledge a tap action has been activated on the touchpad (tap on the Top-Left side of the touchpad) | 1. Wire I2C pins between MSP430G2553 and touchpad TM5957 2. Use test code – read register has coordinate of tap point 3. Tap on Top-Left area of touchpad | 1. Success in initializing I2C protocol 2. Receive coordinate of tap point is Top-Left | TC02 | Passed | 13/11/13 | CuongNC | | BachLT |
| TC05 | Test touchpad code – acknowledge a slide action has been activated on the touchpad (slide on the Left Side of touchpad) | 1. Wire I2C pins between MSP430G2553 and touchpad TM5957 2. Use test code – read register has coordinates of points 3. Slide on the left side of touchpad | 1. Success in initializing I2C protocol 2. Retrieve coordinates of slide points is left side of touchpad | TC02 | Passed | 13/11/13 | CuongNC | | BachLT |
| TC06 | Test touchpad code – acknowledge a slide action has been activated on the touchpad (slide on the Bottom Side of touchpad) | 1. Wire I2C pins between MSP430G2553 and touchpad TM5957 2. Use test code – read register has coordinates of points 3. Slide on the bottom side touchpad | 1. Success in initializing I2C protocol 2. Retrieve coordinates of slide points is bottom side of touchpad | TC02 | Passed | 13/11/13 | CuongNC | | BachLT |
|  | * Integration RF code and Touchpad Code | | | | | | | | |
| RT01 | Test if initializing Touchpad and RF success | 1. Wire 3 SPI pins between 1 MSP430G2553 and nRF24L01 2. Wire 2 I2C pins between MSP430G2553 and Touchpad TM5957 3. Using test code to check initialization RF – send command 4. Using test code to check initialization Touchpad – send command | 1. Success communication to RF – status has been sent back 2. Success communication to Touchpad – status has been sent back | TC01 and CP03 | Passed | 14/11/13 | CuongNC | | BachLT |
| RT02 | Test data send from touchpad, through microchip’s process, through nRF24L01’s RF to receiver | 1. Wire 3 SPI pins between 1 MSP430F6659 and nRF24L01 2. Wire 3 SPI pins between 1 MSP430G2553 and nRF24L01 3. Wire 2 I2C pins between MSP430G2553 and Touchpad TM5957 4. Touch or slide on touchpad to create test data | 1. Success initialization and communication between MSP430F6659 and nRF24L01 2. Success initialization and communication between MSP430G2553 and nRF24L01 3. Data send from touchpad can be recognize at receiver | RT01 | Passed | 14/11/13 | CuongNC | | BachLT |
|  | Receiver Module | | | | | | | | |
|  | * Communication Protocol | | | | | | | | |
| CP11 | Test SPI’s initialization on MSP430F6659 by wiring MOSI and MISO | 1. Wire MISO and MOSI 2. Using SPI initialization code and write a test code to send data through MOSI and receive sent data through MISO | 1. Success initializing SPI pins 2. Able to send data through MOSI and receive data through MISO |  | Passed | 14/11/13 | CuongNC | | CuongNC |
| CP12 | Test SPI’s communication by using MSP430F6659 and MSP430G2553 chip | 1. Wire one chip’s MISO to another one’s MOSI and the same to 2 pins MOSI and MISO, wire 2 SLK pins 2. Using code test – making 1 chip is master and the other one is slave, send data from F6659 master chip and G2553 slave chip echo it back | 1. Success initializing SPI protocol 2. Check if master chip has been echoed back sent data | CP11 | Passed | 14/11/13 | CuongNC | | CuongNC |
| CP13 | Test SPI protocol between MSP430F6659 and RF | 1. Wire 3 SPI pins between F6659 chip and nRF24L01 2. Using code test – Initialize SPI from F6659 chip and send command to RF | 1. Success in initializing SPI protocol 2. nRF24L01 acknowledge command sent from F6659 and echo back status (0x0E is success in communicate expected status) | CP12 | Passed | 14/11/13 | CuongNC | | BachLT |
| CP14 | Test RF communication between transmitter and receiver | 1. Wire 3 SPI pins between 1 MSP430F6659 and nRF24L01 2. Wire 3 SPI pins between 1 MSP430G2553 and nRF24L01 3. Using test code to send data back and forward between 2 microchip through nRF24L01 | 1. Success initialization and communication between MSP430F6659 and nRF24L01 2. Success initialization and communication between MSP430G2553 and nRF24L01 3. Data send from MSP430G2553 can be received at MSP430F6659 through the same channel and addresses | CP12 and CP02 | Passed | 14/11/13 | CuongNC | | BachLT |
|  | * USB HID (Basic report sending) | | | | | | | | |
| UH01 | Test device can be able to acknowledge an USB device has been plugged in – a mouse device | 1. Connect device to a PC which runs windows OS or MAC OS | 1. PC able to acknowledge a new device has been plugged in 2. Check in device manager of PC, it’s a new pointer device has been plugged in |  | Passed | 14/11/13 | CuongNC | | CuongNC |
| UH02 | Test if device can acknowledge report (a set of points) has been sent back from MSP430F6659 | 1. Connect Launchpad to PC 2. Using test code – send a built report (a set of points) from MSP430F6659 to PC | 1. PC able to acknowledge a new pointer device has been plugged in 2. Pointer moves around on the screen as a circle | UH01 | Passed | 14/11/13 | CuongNC | | CuongNC |
| UH03 | Test if device can acknowledge report (a set of button clicks) has been sent back from MSP430F6659 | 1. Connect Launchpad to PC 2. Using test code – send a built report (a set of button clicks) from MSP430F6659 to PC | 1. PC able to acknowledge a new pointer device has been plugged in 2. Pointer starts to left click, then right click | UH01 | Passed | 14/11/13 | CuongNC | | CuongNC |
| UH04 | Test if device can acknowledge report (a set of vertical scroll) has been sent back from MSP430F6659 | 1. Connect Launchpad to PC 2. Using test code – send a built report (a set of vertical scroll) from MSP430F6659 to PC | 1. PC able to acknowledge a new pointer device has been plugged in 2. Pointer starts scroll up and down | UH01 | Passed | 14/11/13 | CuongNC | | CuongNC |
| UH05 | Test if device can acknowledge report (a set of horizontal scroll) has been sent back from MSP430F6659 | 1. Connect Launchpad to PC 2. Using test code – send a built report (a set of horizontal scroll) from MSP430F6659 to PC | 1. PC able to acknowledge a new pointer device has been plugged in 2. Pointer starts scroll left and right | UH01 | Failed | 14/11/13 | CuongNC | | CuongNC |
|  | * Integration RF and Receive Code | | | | | | | | |
| RR01 | Test if initializing USB HID and RF success | 1. Connect Launchpad to PC 2. Using test code to check initializing RF – send command | 1. PC able to acknowledge a new pointer device has been plugged in 2. Success communicating with RF – status has been sent back |  | Passed | 14/11/13 | CuongNC | | CuongNC |
| RR02 | Test set of data send from transmitter to receiver can be recognized at PC device via USB HID | 1. Wire 3 SPI pins between 1 MSP430F6659 and nRF24L01 2. Wire 3 SPI pins between 1 MSP430G2553 and nRF24L01 3. Using test code to send data (set of coordinates) from MSP430G2553 to MSP430F6659 | 1. Success initialization and communication between MSP430F6659 and nRF24L01 2. Success initialization and communication between MSP430G2553 and nRF24L01 3. Pointer start to move around on the screen as a circle | CP14 and RR01 | Passed | 14/11/13 | CuongNC | | CuongNC |

### System Testing

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Test case description | Test case procedure | Expected result | Inter-test case dependence | Result | Test date | Creator | Tester |
| FT01 | Test Move Pointer Action  Slide on touchpad | 1. Wire 3 SPI pins between 1 MSP430F6659 and nRF24L01 2. Wire 3 SPI pins between 1 MSP430G2553 and nRF24L01 3. Wire 2 I2C pins between MSP430G2553 and Touchpad 4. Connect transmitter to a power 5. Connect receiver to a PC 6. Slide on touchpad | 1. Success initialization and communication between MSP430F6659 and nRF24L01 2. Success initialization and communication between MSP430G2553 and nRF24L01 3. Success initialization and communication between MSP430G2553 and Touchpad TM5957 4. Pointer moves as tester slides on touchpad | RR02 | Passed | 15/11/13 | CuongNC | CuongNC  BachLT |
| FT02 | Test Click Action  Tap on touchpad – Anywhere but Top-Left Corner, Bottom Side and Right Side | 1. Wire 3 SPI pins between 1 MSP430F6659 and nRF24L01 2. Wire 3 SPI pins between 1 MSP430G2553 and nRF24L01 3. Wire 2 I2C pins between MSP430G2553 and Touchpad 4. Connect transmitter to a power 5. Connect receiver to a PC 6. Tap on touchpad (Anywhere but Top-Left Corner, Bottom Side and Right Side) | 1. Success initialization and communication between MSP430F6659 and nRF24L01 2. Success initialization and communication between MSP430G2553 and nRF24L01 3. Success initialization and communication between MSP430G2553 and Touchpad TM5957 4. Pointer has left-click action as tester tap on touchpad | RR02 | Passed | 15/11/13 | CuongNC | CuongNC  BachLT |
| FT03 | Test Double-Click Action  Double tap on touchpad - Anywhere but Top-Left Corner, Bottom Side and Right Side | 1. Wire 3 SPI pins between 1 MSP430F6659 and nRF24L01 2. Wire 3 SPI pins between 1 MSP430G2553 and nRF24L01 3. Wire 2 I2C pins between MSP430G2553 and Touchpad 4. Connect transmitter to a power 5. Connect receiver to a PC 6. Double tap on touchpad (Anywhere but Top-Left Corner, Bottom Side and Right Side) | 1. Success initialization and communication between MSP430F6659 and nRF24L01 2. Success initialization and communication between MSP430G2553 and nRF24L01 3. Success initialization and communication between MSP430G2553 and Touchpad TM5957 4. Pointer has double left-click action as tester double tap on touchpad | RR02 | Passed | 15/11/13 | CuongNC | CuongNC  BachLT |
| FT04 | Test Right Click Action  Tap on Top-Left Corner of Touchpad | 1. Wire 3 SPI pins between 1 MSP430F6659 and nRF24L01 2. Wire 3 SPI pins between 1 MSP430G2553 and nRF24L01 3. Wire 2 I2C pins between MSP430G2553 and Touchpad 4. Connect transmitter to a power 5. Connect receiver to a PC 6. Tap on Top-Left Corner of Touchpad | 1. Success initialization and communication between MSP430F6659 and nRF24L01 2. Success initialization and communication between MSP430G2553 and nRF24L01 3. Success initialization and communication between MSP430G2553 and Touchpad TM5957 4. Pointer has right-click action as tester tap on Top-Left Corner of touchpad | RR02 | Passed | 15/11/13 | CuongNC | CuongNC  BachLT |
| FT05 | Test Vertical Scroll  Slide on the right side of touchpad | 1. Wire 3 SPI pins between 1 MSP430F6659 and nRF24L01 2. Wire 3 SPI pins between 1 MSP430G2553 and nRF24L01 3. Wire 2 I2C pins between MSP430G2553 and Touchpad 4. Connect transmitter to a power 5. Connect receiver to a PC 6. Slide on the right side of touchpad | 1. Success initialization and communication between MSP430F6659 and nRF24L01 2. Success initialization and communication between MSP430G2553 and nRF24L01 3. Success initialization and communication between MSP430G2553 and Touchpad TM5957 4. Pointer has vertical scroll action as tester slide on the right side of touchpad | RR02 | Passed | 15/11/13 | CuongNC | CuongNC  BachLT |
| FT06 | Test Horizontal Scroll  Slide on the bottom side of touchpad | 1. Wire 3 SPI pins between 1 MSP430F6659 and nRF24L01 2. Wire 3 SPI pins between 1 MSP430G2553 and nRF24L01 3. Wire 2 I2C pins between MSP430G2553 and Touchpad 4. Connect transmitter to a power 5. Connect receiver to a PC 6. Slide on the bottom side of touchpad | 1. Success initialization and communication between MSP430F6659 and nRF24L01 2. Success initialization and communication between MSP430G2553 and nRF24L01 3. Success initialization and communication between MSP430G2553 and Touchpad TM5957 4. Pointer has horizontal scroll action as tester slide on the bottom side of touchpad | RR02 | Failed | 15/11/13 | CuongNC | CuongNC  BachLT |
| FT07 | Test Vertical Scroll  Touch on the right side of touchpad then move (can be able to move out of Right-Side range) and System still understand it’s a vertical scroll | 1. Wire 3 SPI pins between 1 MSP430F6659 and nRF24L01 2. Wire 3 SPI pins between 1 MSP430G2553 and nRF24L01 3. Wire 2 I2C pins between MSP430G2553 and Touchpad 4. Connect transmitter to a power 5. Connect receiver to a PC 6. Touch on the right side of touchpad then move (can be able to move out of Right-Side range) | 1. Success initialization and communication between MSP430F6659 and nRF24L01 2. Success initialization and communication between MSP430G2553 and nRF24L01 3. Success initialization and communication between MSP430G2553 and Touchpad TM5957 4. Pointer has vertical scroll action even though tester moves out of Left-Side range. The point of this action is user only need to keep in touch with the touchpad from the first touch at the right side and then move, the system still understand it as a vertical scroll | RR02 | Passed | 15/11/13 | CuongNC | CuongNC  BachLT |
| FT08 | Test Horizontal Scroll  Touch on the bottom side of touchpad then move (can be able to move out of Bottom-Side range) and System still understand it’s a horizontal scroll | 1. Wire 3 SPI pins between 1 MSP430F6659 and nRF24L01 2. Wire 3 SPI pins between 1 MSP430G2553 and nRF24L01 3. Wire 2 I2C pins between MSP430G2553 and Touchpad 4. Connect transmitter to a power 5. Connect receiver to a PC 6. Touch on the bottom side of touchpad then move (can be able to move out of Bottom-Side range) | 1. Success initialization and communication between MSP430F6659 and nRF24L01 2. Success initialization and communication between MSP430G2553 and nRF24L01 3. Success initialization and communication between MSP430G2553 and Touchpad TM5957 4. Pointer has horizontal scroll action even though tester moves out of Bottom-Side range. The point of this action is user only need to keep in touch with the touchpad from the first touch at the bottom side and then move, the system still understand it as a horizontal scroll | RR02 | Failed | 15/11/13 | CuongNC | CuongNC  BachLT |
| FT09 | Test Drag action  Tap on touchpad then tap again without release finger and move – | 1. Wire 3 SPI pins between 1 MSP430F6659 and nRF24L01 2. Wire 3 SPI pins between 1 MSP430G2553 and nRF24L01 3. Wire 2 I2C pins between MSP430G2553 and Touchpad 4. Connect transmitter to a power 5. Connect receiver to a PC 6. Tap on touchpad then tap again without release finger and move | 1. Success initialization and communication between MSP430F6659 and nRF24L01 2. Success initialization and communication between MSP430G2553 and nRF24L01 3. Success initialization and communication between MSP430G2553 and Touchpad TM5957 4. Pointer has drag action as tester taps on touchpad then taps again without release finger and move | RR02 | Passed | 15/11/13 | CuongNC | CuongNC  BachLT |

# SOFTWARE USER’S MANUAL

## Package Contents

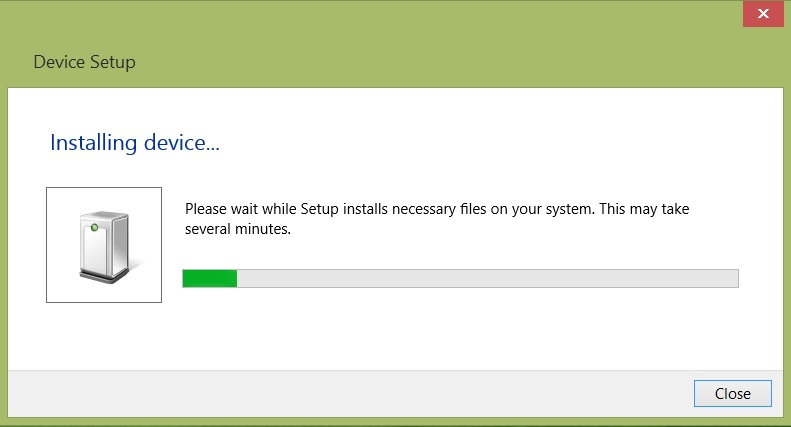
* Touchpad (include a board which has Touchpad TM5957, MSP430G2553 chip and nRF24L01)
* Receiver (include a board which has MSP430F6659, USB plug, nRF24L01)
* User’s Manual

## Installation Guide

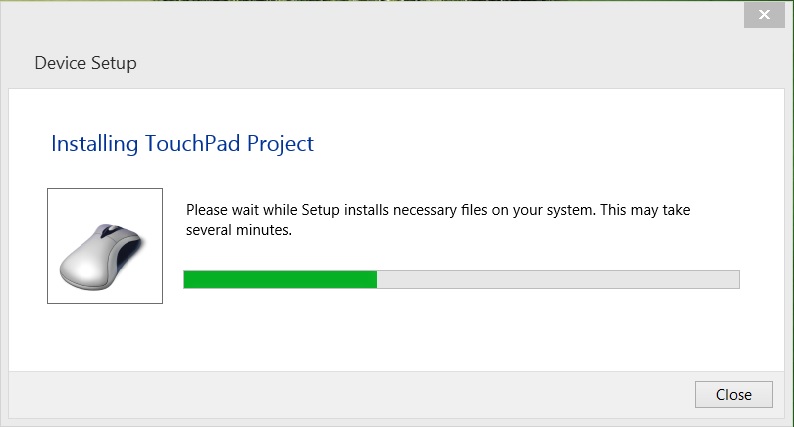
**Step 1**: Plug receiver into PC

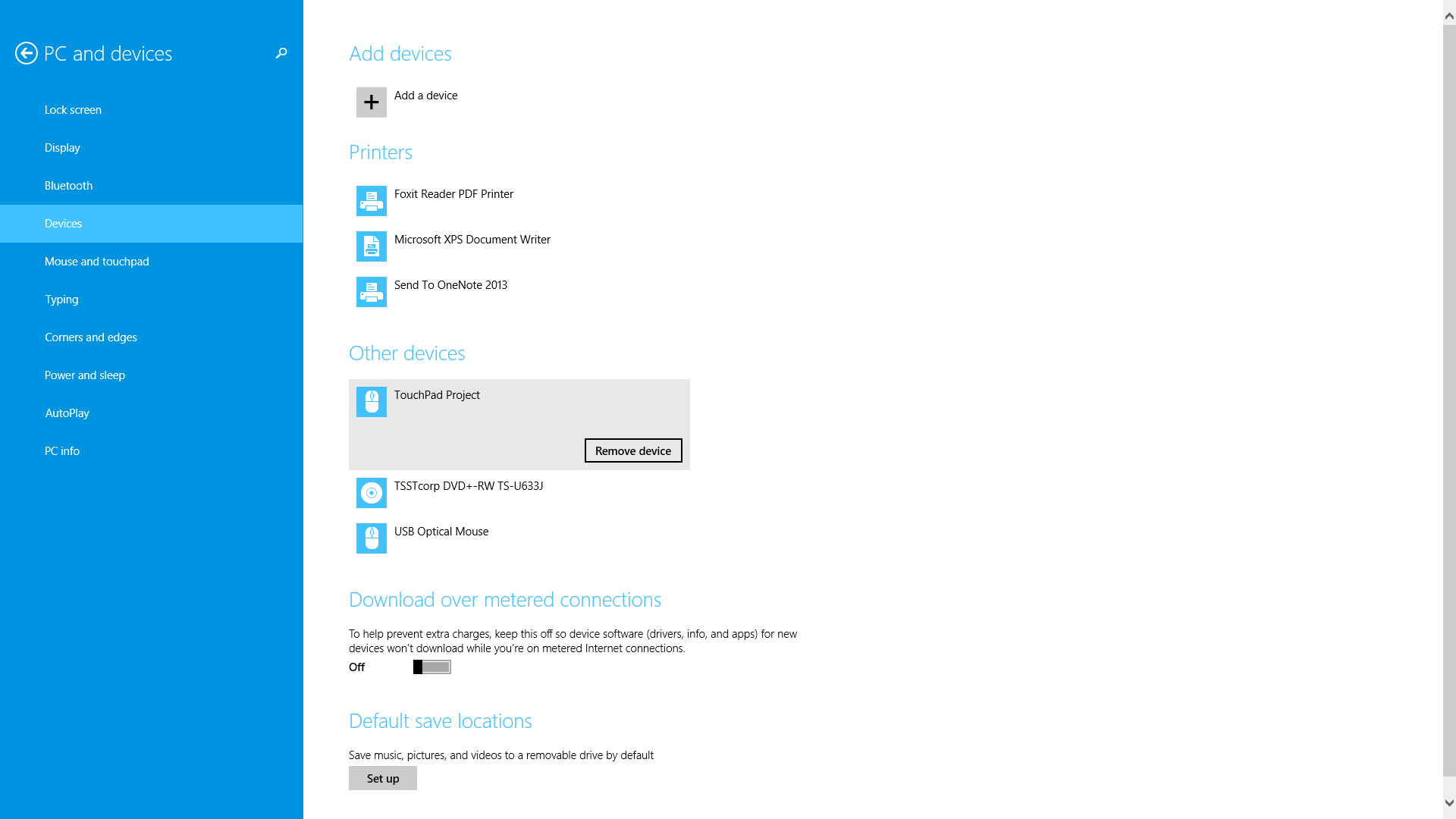


**Step 2**: Waiting for OS install the device



Success:





**Step 3**: Turn on the touchpad device



## User Guide

After completing these simple steps, user can be able to use the Wireless Touchpad

Basic Features:

* Tap on Touchpad for simple Left-Click Action
* Slide on Touchpad for simple moving Action

Advanced Features:

* Tap on the Top-Left of Touchpad for Right-Click Action
* To have a Vertical Scroll Action, user can tap on the Left-Side Corner and moving Up-Down to scroll
* To Drag, user tap on touchpad and release then re-touch and keep touching along with moving will be acknowledged as a Drag action

# REFERENCES

* Receiver Hardware:

Reuse Receiver PCB Layout from Keyboard HeroES team

* MSP430F6659:

<http://www.ti.com/lit/ds/symlink/msp430f6659.pdf>

* MSP430G2553:

<http://www.ti.com/lit/gpn/msp430g2553>

* nRF24L01:

<https://www.sparkfun.com/datasheets/Components/SMD/nRF24L01Pluss_Preliminary_Product_Specification_v1_0.pdf>

* Touchpad TM5957:

Since this touchpad’s documents are confidential and we have to ask for their confirmation, we don’t have any links to reference to. Below are official page of TM5957:

<http://www.cirque.com/solutions/oemtouchpads/tm5957.aspx>

* TI forum:

<http://e2e.ti.com/>

* Google:

<https://www.google.com/>