Department of Computer Science and Engineering
The University of Texas at Arlington

Team Itus

NeverBLIND System

Team Members:
Austin Miller
Adrian Kotev
An Nguyen
Jacky Trevino
Mark Madolora

Late Updated: 2 October 2012 @ 2:03:00 PM
Table of Contents

List of Tables ........................................................................................................................................ iii
List of Figures ........................................................................................................................................ iv
Document Revision History ............................................................................................................... v

1 Introduction ...................................................................................................................................... 1
  1.1 Overview .................................................................................................................................... 1
  1.2 Product Overview ....................................................................................................................... 1
  1.3 Project Scope ............................................................................................................................. 2
  1.4 Definitions and Acronyms ........................................................................................................... 3

2 Meta-Architecture ............................................................................................................................ 5
  2.1 Architectural Vision ..................................................................................................................... 5
  2.2 Guiding Principles and Rules ..................................................................................................... 5
  2.3 Tradeoffs .................................................................................................................................... 6

3 Architecture Overview .................................................................................................................... 7
  3.1 Network Layer .......................................................................................................................... 8
  3.2 Sensing Layer ........................................................................................................................... 8
  3.3 I/O Layer .................................................................................................................................... 8
  3.4 Packet Layer ............................................................................................................................ 8
  3.5 CPU Layer .................................................................................................................................. 8

4 Network Layer .................................................................................................................................. 9
  4.1 Description .................................................................................................................................. 9
  4.2 Purpose ...................................................................................................................................... 9
  4.3 Function ..................................................................................................................................... 9
  4.4 Dependencies ............................................................................................................................. 9
  4.5 Interfaces ................................................................................................................................... 9
  4.6 Processing .................................................................................................................................. 10
  4.7 Data .......................................................................................................................................... 10
  4.8 Subsystems ............................................................................................................................... 10

5 Sensing Layer .................................................................................................................................. 12
  5.1 Description .................................................................................................................................. 12
  5.2 Purpose ...................................................................................................................................... 12
  5.3 Function ..................................................................................................................................... 12
  5.4 Dependencies ............................................................................................................................. 13
  5.5 Interfaces ................................................................................................................................... 13
  5.6 Processing ............................................................................................................................... 13
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Subsection</th>
<th>Description</th>
<th>Purpose</th>
<th>Function</th>
<th>Dependencies</th>
<th>Interfaces</th>
<th>Processing</th>
<th>Data</th>
<th>Subsystems</th>
</tr>
</thead>
<tbody>
<tr>
<td>6 I/O Layer</td>
<td>6.1</td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7 Packet Layer</td>
<td>7.1</td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8 CPU Layer</td>
<td>8.1</td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9 Operating System Dependencies</td>
<td>9.1</td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10 Relationship Mapping</td>
<td>10.1</td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11 Requirements Mapping</td>
<td>11.1</td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12 Testing Considerations</td>
<td>12.1</td>
<td>Description</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
List of Tables

Table 1-1 Definitions and Acronyms......................................................................................... 4
Table 4-1 Network Layer Interfaces .......................................................................................... 10
Table 4-2 Receiver Interfaces ...................................................................................................... 10
Table 4-3 Sender Interfaces ........................................................................................................ 11
Table 5-1 Sensing Layer Interfaces ............................................................................................. 13
Table 5-2 GPS Component Interfaces .......................................................................................... 14
Table 5-3 Compass Component Interfaces .................................................................................. 14
Table 6-1 I/O Layer Interfaces .................................................................................................... 16
Table 6-2 Output Component Interfaces ..................................................................................... 17
Table 6-3 Debug Log Component Interfaces ................................................................................. 17
Table 6-4 Input Component Interfaces ........................................................................................ 18
Table 7-1 Packet Layer Interfaces ............................................................................................... 20
Table 7-3 Packet Encoder Interfaces ........................................................................................... 21
Table 7-4 Packet Decoder Interfaces ............................................................................................ 21
Table 7-5 Packet Interpreter Interfaces ....................................................................................... 22
Table 8-1 CPU Layer Interfaces .................................................................................................. 24
Table 8-2 Data Processor Interfaces ............................................................................................ 25
Table 8-6 IFF Component Interfaces ........................................................................................... 26
Table 9-1 Data Elements ............................................................................................................. 30
Table 9-2 Producer-Consumer Relationship Matrix ................................................................... 30
Table 10-1 Requirements Mapping ............................................................................................. 31
List of Figures

Figure 1-1 NeverBLIND System Concept Drawing................................................................. 2
Figure 1-2 NeverBLIND System Technical Concept.................................................................. 3
Figure 3-1 Architecture Overview ............................................................................................ 7
Figure 4-1 Network Layer ........................................................................................................ 9
Figure 5-1 Sensing Layer ......................................................................................................... 12
Figure 6-1 I/O Layer ................................................................................................................. 15
Figure 7-1 Packet Layer .......................................................................................................... 19
Figure 8-1 CPU Layer ............................................................................................................. 23
Figure 9-1 Data Flow between Layers and Subsystems ............................................................. 28
Document Revision History

<table>
<thead>
<tr>
<th>Revision #</th>
<th>Revision Date</th>
<th>Description</th>
<th>Rationale</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>09/04/2012</td>
<td>Initial Compilation</td>
<td>Initial Compilation</td>
</tr>
<tr>
<td>0.2</td>
<td>09/07/2012</td>
<td>First Internal Review Draft</td>
<td>Changes made according to group comments</td>
</tr>
<tr>
<td>0.3</td>
<td>09/08/2012</td>
<td>Second Internal Draft</td>
<td>Added Relationship Mapping and Requirements Mapping</td>
</tr>
<tr>
<td>0.4</td>
<td>9/12/2012</td>
<td>External Review Draft</td>
<td>Revisions made based on feedback from TAs, TGBL and group</td>
</tr>
<tr>
<td>1.0</td>
<td>9/12/2012</td>
<td>Gate Review</td>
<td>Gate Review Submission</td>
</tr>
<tr>
<td>2.0</td>
<td>9/22/2012</td>
<td>Gate Review Revision</td>
<td>Revisions made based on gate review comments</td>
</tr>
</tbody>
</table>
1 Introduction

1.1 Overview
This Architecture Design Specification document represents the architecture design for the NeverBLIND System. It will be a conceptual model of what is to be done to construct the NeverBLIND System.

1.2 Product Overview
The NeverBLIND System will be an easy to deploy, weapon mountable system that can identify friendly targets by using a transmitter, receiver and notification device. All friendly personnel will be required to carry a transmitter, receiver, and notification device with their weapon. The NeverBLIND System will be able to support up to four different teams. The system will have a switch where the user can select a team.

The NeverBLIND system will ensure that if a friendly target is in the LOS of the user’s weapon, the system will notify the user that a friendly is in the LOS until the target has moved out of the LOS of the weapon.

All subsystems, receiver, transmitter and notification device will operate independently of one another by processing information within each subsystem and creating data packets they can pass to other subsystems.

The system will consist of basic switches, buttons and lights to indicate when the product is on and when it is charging. There will also be a light that will indicate when the user is pointing the weapon at a friendly.
1.3 Project Scope

The final product will be kept to the paintball domain. Team Itus will work on a product that is safe and easy to deploy. The product will ensure that if a friendly target is in the line of sight of the user’s weapon, the system will notify the user that a friendly is in the line of sight until the target has moved out of the line of sight of the weapon. The NeverBLIND System is intended to use a combination of GPS and a 3-Axis compass in order to determine the positions of teammates and radio frequency to enable the use of multiple devices. The technical concept is shown in Figure 1-2.
1.4 Definitions and Acronyms

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>IFF</td>
<td>Identification Friend or Foe</td>
</tr>
<tr>
<td>LED</td>
<td>Light-Emitting Diode</td>
</tr>
<tr>
<td>IDE</td>
<td>Integrated Development Environment</td>
</tr>
<tr>
<td>LOS</td>
<td>Line of Sight</td>
</tr>
<tr>
<td>OS</td>
<td>Operating System</td>
</tr>
<tr>
<td>I/O</td>
<td>Input / Output</td>
</tr>
<tr>
<td>CPU</td>
<td>Central Processing Unit</td>
</tr>
<tr>
<td>x</td>
<td>Longitude</td>
</tr>
<tr>
<td>y</td>
<td>Latitude</td>
</tr>
<tr>
<td>z</td>
<td>Altitude</td>
</tr>
<tr>
<td>Symbol</td>
<td>Definition</td>
</tr>
<tr>
<td>--------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>o</td>
<td>Orientation of the user</td>
</tr>
<tr>
<td>h</td>
<td>Heading of the user</td>
</tr>
<tr>
<td>IR</td>
<td>Infrared</td>
</tr>
<tr>
<td>RF</td>
<td>Radio Frequency</td>
</tr>
</tbody>
</table>

**Table 1-1 Definitions and Acronyms**
2 Meta-Architecture

2.1 Architectural Vision

The NeverBLIND System will use a transmitter, receiver and notification device. The transmitter will transmit a friendly signal which will allow the receiver device to identify or not identify the target. The notification device will illuminate when a friendly target is detected.

The NeverBLIND System will consist of five layers: Network, Sensing, I/O, Packet and CPU layer. Each layer is guaranteed to be independent from each other and each layers’ functions will have little reliance on other layers. Designing independent layers minimizes the affect that changes made to one layer during implementation or construction has on other layers.

2.2 Guiding Principles and Rules

2.2.1 Extensibility

System is intended to be extensible in the future, so the architecture should be designed where modules can be added with minimal amount of effort.

2.2.2 Portability

System is intended to be mountable to a paintball gun, so it must attachable to the paintball gun. The size of the parts should be kept to a minimum, while the casing should be easy to handle.

2.2.3 Configurability

The system should have a layer designed in a way that the configurable parts of the user interface can be changed without effecting the operation of the rest of the system. System will allow users to configure the device through a configuration interface. Users shall be allowed to select teams and information which will be filtered to make sure team members get accurate information just from their teams.

2.2.4 Independence

Design must assure that every layer is independent from each other. This will allow changes in the implementation within one layer to be independent and to not affect other layers.

2.2.5 Integrity

Design must assure the correct accuracy, fastest speed and reliability when aiming at another user.
2.2.6 Durability

Design shall ensure that the hardware interface does not break. Team Itus will purchase sturdy components when possible.

2.3 Tradeoffs

The project was first intended to be in the military domain. However, after thorough research, Team Itus found that the system would have to meet certain specifications in order to be considered in the military domain which would greatly increase the complexity of the project. Team Itus decided to keep the NeverBLIND System to the paintball domain in order to keep the project feasible by avoiding complex military requirements. However, Team Itus will design the system architecture to be expandable by keeping major system layers independent for easy maintenance and future development.

Another tradeoff involved adding two new independent layers which increased complexity of the project but solved the RF’s directional problem and IR’s distance problem. At first, Team Itus decided to use infrared sensors, IR, to identify where users are pointing at to send and receive signals. However, the IR approach provides a short range distance (up to 0.1 meters) which does not meet the requirement for the paintball domain. On another hand, the radio frequency, RF, approach has an acceptable range of up to 300 meters which makes NeverBLIND system feasible for the paintball domain. However, the signal from RF propagates everywhere giving a directional problem because we do not know where we actually send and receive signals from. Therefore, Team Itus decided to develop two new independent layers: the position layer and the network layer to solve both distance and directional problem and keep the project expandable in the future.

A position layer was added which contains two sub-systems: a GPS device and a 3-Axis compass. The system gets coordinate information from the GPS and directional information from the 3-Axis compass before sending them to the CPU layer. The CPU layer will apply algorithms to efficiently estimate where the users are located and where they are heading. The position of the user will be collected to be sent to the network.

The network layer uses the Xbee radio frequency network to get data from the CPU as packets to send to other NeverBLIND Systems. At the same time, it receives the information from other users to process and trigger the system as needed.
3 Architecture Overview

The NeverBLIND System will be divided into five main layers: Network, Sensing, I/O, Packet and CPU. Figure 3-1 represents the architecture overview with the dataflow.

![Figure 3-1 Architecture Overview](image-url)
3.1 Network Layer

The network layer acts as an intermediary between other NeverBLIND devices and its own CPU layer. The network layer receives packets from other NeverBLIND devices containing their positions and transports them to its own CPU layer to be decoded. The network layer also receives encoded position packets and sends them the network layer of other NeverBLIND devices.

3.2 Sensing Layer

The sensing layer will gather the position and orientation of the NeverBLIND device by using a combination of GPS and compass components. The sensing layer will transform the raw data from the GPS and compass components into software objects to be transported to the CPU layer.

3.3 I/O Layer

The I/O layer will provide an interface between the CPU layer and the user of the NeverBLIND system. The I/O layer will consist of two main components: input and output. Input will take in discrete values from the NeverBLIND user and convert them into software objects to be transported to the data processor contained in the CPU layer. Output will take software objects from the CPU layer and transform them into discrete analog outputs.

3.4 Packet Layer

The packet layer is responsible for packet handling within the NeverBLIND system. The packet layer will receive information from the data processor and encode them before being transported to the network layer. The packet layer will also receive packets from the network layer, filter them based on team ID and interpret them before transporting them to the CPU layer.

3.5 CPU Layer

The CPU layer is responsible for handling the necessary computations for the NeverBLIND System. The CPU layer will retrieve the user’s position, orientation and heading from the sensing layer, find an optimized average and transport them to the packet layer. In addition, the CPU layer will receive interpreted packets from the packet layer in the IFF component to determine if the teammate is within the line of sight before transporting discrete output to the I/O layer.
4 Network Layer

4.1 Description
The network layer communicates internally with the packet layer and communicates externally with other NeverBLIND devices. The network layer has two subsystems: a sender and a receiver.

4.2 Purpose
The purpose of the network layer is to transport packets between other NeverBLIND devices and its own packet layer.

4.3 Function
The network layer receives packets from other NeverBLIND devices and sends these packets to the packet layer. It also transmits packets received from packet layer to other NeverBLIND devices. The network layer does not process any data; it is simply an intermediary layer.

4.4 Dependencies
The Network layer is dependent upon the packet layer to provide encoded packets to send to other NeverBLIND devices.

4.5 Interfaces

<table>
<thead>
<tr>
<th>Type</th>
<th>Source</th>
<th>Destination</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal</td>
<td>Packet encoder</td>
<td>Sender</td>
<td>Packets encoded by the packet layer to send to other NeverBLIND device(s)</td>
</tr>
</tbody>
</table>
### Table 4-1 Network Layer Interfaces

<table>
<thead>
<tr>
<th>Source</th>
<th>Data</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>External Sender</td>
<td>Other NeverBLIND device(s)</td>
<td>Packets to be decoded by the packet layer of destination device</td>
</tr>
<tr>
<td>External Other NeverBLIND device(s) Receiver</td>
<td>Packets being sent from other NeverBLIND devices</td>
<td></td>
</tr>
<tr>
<td>Internal Receiver</td>
<td>Packet decoder</td>
<td>Packets received from other NeverBLIND devices sent to the packet layer to be decoded</td>
</tr>
</tbody>
</table>

#### 4.6 Processing

None

#### 4.7 Data

The network layer sends and receives network packets which contain the following information about the sending device: GPS coordinates of the sending device and the team identifier of the sending device.

#### 4.8 Subsystems

##### 4.8.1 Receiver

#### 4.8.1.1 Purpose

The purpose of the receiver is to receive network packets from other NeverBLIND devices.

#### 4.8.1.2 Function

When the receiver picks up a packet from another NeverBLIND device it sends the data to the packet layer to be decoded.

#### 4.8.1.3 Dependencies

None

#### 4.8.1.4 Interfaces

<table>
<thead>
<tr>
<th>Source</th>
<th>Data</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Receiver</td>
<td>Packet</td>
<td>Packet decoder</td>
</tr>
<tr>
<td>Other NeverBLIND Device</td>
<td>Packet</td>
<td>Receiver</td>
</tr>
</tbody>
</table>

#### 4.8.1.5 Processing

None
4.8.1.6 Data
Network packets

4.8.2 Sender

4.8.2.1 Purpose
The purpose of the sender is to send network packets to other NeverBLIND devices.

4.8.2.2 Function
Receives encoded network packets from the packet layer and sends them to other NeverBLIND devices.

4.8.2.3 Dependencies
Packet layer

4.8.2.4 Interfaces

<table>
<thead>
<tr>
<th>Source</th>
<th>Data</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sender</td>
<td>Packet</td>
<td>Other NeverBLIND device(s)</td>
</tr>
<tr>
<td>Packet Encoder</td>
<td>Packet</td>
<td>Sender</td>
</tr>
</tbody>
</table>

Table 4-3 Sender Interfaces

4.8.2.5 Processing
None

4.8.2.6 Data
Network packets
5 Sensing Layer

5.1 Description
The sensing layer is responsible for gathering position, orientation and heading data for the NeverBLIND System. This layer consists of two subsystems: a GPS component and a 3-Axis compass component.

5.2 Purpose
The purpose of the sensing layer is to gather the longitude, latitude and altitude, referred to as x, y and z respectively from the GPS hardware component and raw orientation and heading data referred to as o and h from the 3-Axis compass hardware component and transform the raw data into software objects in the form of Position(x, y, z) and Compass(o, h). The sensing layer will then pass the software objects to the CPU layer.

5.3 Function
The sensing layer interfaces with GPS and 3-Axis compass hardware components and computes pin inputs to logical data. The sensing layer is specific to its hardware interface. Its function contains pin input to one layer and frees other layers to use internal software objects.
5.4 Dependencies
The sensing layer is dependent on GPS component and 3-Axis compass component hardware. Its converter logic will be specific to hardware (GPS hardware module) and (compass hardware module).

5.5 Interfaces

<table>
<thead>
<tr>
<th>Type</th>
<th>Source</th>
<th>Destination</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal</td>
<td>GPS</td>
<td>Data processor</td>
<td>GPS location in the form of Position (x, y, z)</td>
</tr>
<tr>
<td>Internal</td>
<td>3-Axis Compass</td>
<td>Data processor</td>
<td>Compass data in the form of Compass(o, h)</td>
</tr>
<tr>
<td>External</td>
<td>GPS Satellite</td>
<td>GPS</td>
<td>Raw data in the form of (x, y, z)</td>
</tr>
<tr>
<td>External</td>
<td>Magnetic North</td>
<td>3-Axis Compass</td>
<td>Raw data in the form of (o, h)</td>
</tr>
</tbody>
</table>

Table 5-1 Sensing Layer Interfaces

5.6 Processing
The sensing layer will process digital pin input from both the GPS component and 3-Axis compass component hardware, convert them to software objects and send the converted software objects to the CPU layer.

5.7 Data
The sensing layer will process raw digital pin input and output software objects to the data processor within the CPU layer.

5.8 Subsystems

5.8.1 GPS Component

5.8.1.1 Purpose
The purpose of the GPS component is to gather the location of the NeverBLIND device in the form of x, y and z.

5.8.1.2 Function
Gather digital pin input from GPS component hardware and convert to a GPS position software object.

5.8.1.3 Dependencies
GPS component hardware [specific module here]

5.8.1.4 Interfaces

<table>
<thead>
<tr>
<th>Source</th>
<th>Data</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPS</td>
<td>Position(x, y, z)</td>
<td>Data processor</td>
</tr>
<tr>
<td>Source</td>
<td>Data</td>
<td>Destination</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------</td>
<td>------------------------------------</td>
</tr>
<tr>
<td>GPS Satellite</td>
<td>Raw position data (x, y, z)</td>
<td>GPS</td>
</tr>
</tbody>
</table>

Table 5-2 GPS Component Interfaces

5.8.1.5 Processing
Process digital pin input and convert to software object

5.8.1.6 Data
GPS data containing x, y, and z position

5.8.2 3-Axis Compass Component

5.8.2.1 Purpose
The purpose of the compass component is to gather the orientation of the NeverBLIND device.

5.8.2.2 Function
Gather digital pin input from compass component hardware and convert to orientation software object.

5.8.2.3 Dependencies
Compass component hardware [specific module here]

5.8.2.4 Interfaces

<table>
<thead>
<tr>
<th>Source</th>
<th>Data</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-Axis Compass</td>
<td>Compass(o, h)</td>
<td>Data processor</td>
</tr>
<tr>
<td>Magnetic North</td>
<td>Raw compass data (o, h)</td>
<td>3-Axis Compass</td>
</tr>
</tbody>
</table>

Table 5-3 Compass Component Interfaces

5.8.2.5 Processing
Process digital pin input and convert to software object.

5.8.2.6 Data
Orientation and heading
# 6 I/O Layer

![I/O Layer Diagram]

6.1 Description

The I/O (Input / Output) layer is responsible for handling user input and system output of the NeverBLIND System. The I/O layer consists of three subsystems: an input component, an output component, and a debug log component.
6.2 Purpose
The I/O layer will provide the bridge between the CPU and packet layers and the user. It will interface internal data to external data and vice versa.

6.3 Function
The I/O layer will provide and generate debug logs, IFF notifications out, and LED light switches in its output subsystem. The I/O layer will interface the team select switch as well as the on/off switch. The I/O layer will be responsible for converting data between input and output formats.

6.4 Dependencies
The I/O layer will not be dependent on data from the data processor and input from the hardware input.

6.5 Interfaces

<table>
<thead>
<tr>
<th>Type</th>
<th>Source</th>
<th>Destination</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal</td>
<td>Input</td>
<td>Packet encoder</td>
<td>Discrete values from input components to be converted to software objects</td>
</tr>
<tr>
<td>Internal</td>
<td>Input</td>
<td>Packet decoder</td>
<td>Discrete values from input components to be converted to software objects</td>
</tr>
<tr>
<td>Internal</td>
<td>IFF component</td>
<td>Output</td>
<td>Software objects to be converted into discrete analog outputs</td>
</tr>
<tr>
<td>Internal</td>
<td>IFF component</td>
<td>Debug log</td>
<td>Digital output to be used for debugging purpose during development</td>
</tr>
</tbody>
</table>

Table 6-1 I/O Layer Interfaces

6.6 Processing
The I/O layer will process discrete values from input components and convert them to software object values. It will also convert software object values to discrete analog outputs and digital output will be used for debugging during development.

6.7 Data
Data will consist of discrete inputs, discrete outputs, and digital outputs.

6.8 Subsystems

6.8.1 Output Component

6.8.1.1 Purpose
The purpose of the output component is to output internal data to a human readable output format of the NeverBLIND System.
6.8.1.2 Function
Convert internal software objects to discrete outputs.

6.8.1.3 Dependencies
Software object data structures and hardware interfaces

6.8.1.4 Interfaces

<table>
<thead>
<tr>
<th>Source</th>
<th>Data</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>IFF component</td>
<td>Discrete output</td>
<td>Output</td>
</tr>
</tbody>
</table>

Table 6-2 Output Component Interfaces

6.8.1.5 Processing
Process software object data structures into discrete or file output.

6.8.1.6 Data
Software objects, digital signals and file output

6.8.2 Debug Log Component

6.8.2.1 Purpose
The purpose of the debug log component is to output digital data to a human readable output format to be used for debugging during the development process of the NeverBLIND System.

6.8.2.2 Function
Display digital output to be used for debugging during development.

6.8.2.3 Dependencies
Software object data structures and hardware interfaces

6.8.2.4 Interfaces

<table>
<thead>
<tr>
<th>Source</th>
<th>Data</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>IFF component</td>
<td>Digital output</td>
<td>Debug log</td>
</tr>
</tbody>
</table>

Table 6-3 Debug Log Component Interfaces

6.8.3 Input Component

6.8.3.1 Purpose
The purpose of the input component is to provide an interface for user input into the NeverBLIND System.
6.8.3.2 Function
Convert discrete digital hardware input into software objects.

6.8.3.3 Dependencies
Hardware components and software object data structures

6.8.3.4 Interfaces

<table>
<thead>
<tr>
<th>Source</th>
<th>Data</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>Discrete input</td>
<td>Packet encoder</td>
</tr>
<tr>
<td>Input</td>
<td>Discrete input</td>
<td>Packet decoder</td>
</tr>
</tbody>
</table>

Table 6-4 Input Component Interfaces

6.8.3.5 Processing
Process discrete digital hardware input and convert to software object.

6.8.3.6 Data
Discrete input
7 Packet Layer

7.1 Description
The packet layer is responsible for handling packets within the NeverBLIND system. The packet layer has three subsystems: packet decoder, packet interpreter and packet encoder.

7.2 Purpose
The packet layer receives data to be encoded from the CPU layer and packets to be decoded and interpreted from the network layer. It filters the received packets based on the team ID before sending the interpreted packets to the CPU layer.

7.3 Function
The primary function of the packet layer is to handle the transportation of packets between the network and CPU layer. When the packet layer receives processed data from the CPU layer, it will encode the data into a packet before sending it to the network layer. When a packet is received from the network layer, the packet decoder will filter the packets based on the team ID received from the input layer and interpret the packets into a computer readable format before transporting the packets to the CPU layer.
7.4 Dependencies
The packet layer requires accurate data received from the CPU layer and network.

7.5 Interfaces

<table>
<thead>
<tr>
<th>Type</th>
<th>Source</th>
<th>Destination</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal</td>
<td>Receiver</td>
<td>Packet decoder</td>
<td>Packet</td>
</tr>
<tr>
<td>Internal</td>
<td>Input</td>
<td>Packet decoder</td>
<td>Discrete input</td>
</tr>
<tr>
<td>Internal</td>
<td>Packet decoder</td>
<td>Packet decoder</td>
<td>Filtered packet</td>
</tr>
<tr>
<td>Internal</td>
<td>Packet interpreter</td>
<td>IFF component</td>
<td>Interpreted packet</td>
</tr>
<tr>
<td>Internal</td>
<td>Packet encoder</td>
<td>Sender</td>
<td>Packet</td>
</tr>
<tr>
<td>Internal</td>
<td>Data processor</td>
<td>Packet encoder</td>
<td>Processed data</td>
</tr>
<tr>
<td>Internal</td>
<td>Input</td>
<td>Packet encoder</td>
<td>Discrete input</td>
</tr>
</tbody>
</table>

Table 7-1 Packet Layer Interfaces

7.6 Processing
The packet layer will process packets received from the network layer by decoding, filtering and interpreting them. The packet layer will also take processed data from the CPU layer and encode them for transport by the network layer.

7.7 Data
Packets, filtered packets, interpreted packets, processed data and discrete input

7.8 Subsystems

7.8.1 Packet Encoder

7.8.1.1 Purpose
The purpose of the packet encoder is to encode the position of the user into a packet to be transported by the network layer.

7.8.1.2 Function
Gather all the information to packet, encode them and send them to the network layer.

7.8.1.3 Dependencies
Processed data from data processor
7.8.1.4 Interfaces

<table>
<thead>
<tr>
<th>Source</th>
<th>Data</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packet encoder</td>
<td>Packet</td>
<td>Sender</td>
</tr>
<tr>
<td>Data processor</td>
<td>Processed data</td>
<td>Packet encoder</td>
</tr>
</tbody>
</table>

Table 7-2 Packet Encoder Interfaces

7.8.1.5 Processing
Encode the position of the user for transport to the sender component of the network layer.

7.8.1.6 Data
Packets and processed data

7.8.2 Packet Decoder

7.8.2.1 Purpose
The purpose of the packet decoder is to decode the packets received from the receiver component of the network layer and filter them based on team ID received from the input component then transport the processed information to the packet interpreter.

7.8.2.2 Function
Receive packet from receiver component of the network layer, decode and send the filtered packets to the packet interpreter.

7.8.2.3 Dependencies
Processed data from the network layer

7.8.2.4 Interfaces

<table>
<thead>
<tr>
<th>Source</th>
<th>Data</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packet decoder</td>
<td>Filtered packet</td>
<td>Packet interpreter</td>
</tr>
<tr>
<td>Receiver</td>
<td>Packet</td>
<td>Packet decoder</td>
</tr>
</tbody>
</table>

Table 7-3 Packet Decoder Interfaces

7.8.2.5 Processing
Decode the packet from the receiver for transport to the data processor.

7.8.2.6 Data
Packet and filtered packets
7.8.3 Packet Interpreter

7.8.3.1 Purpose
The purpose of packet interpreter is to transform the decoded data from the packet decoder into computer readable data before sending it to IFF component to determine if the teammate is within the line of sight of the user.

7.8.3.2 Function
The packet interpreter will get decoded data from the packet decoder as a stream of bytes and transform it into computer readable.

7.8.3.3 Dependencies
Data from the packet decoder

7.8.3.4 Interfaces

<table>
<thead>
<tr>
<th>Source</th>
<th>Data</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packet decoder</td>
<td>Filtered packet</td>
<td>Packet interpreter</td>
</tr>
<tr>
<td>Packet interpreter</td>
<td>Interpreted data</td>
<td>IFF component</td>
</tr>
</tbody>
</table>

Table 7-4 Packet Interpreter Interfaces

7.8.3.5 Processing
Process position information from packet decoder

7.8.3.6 Data
GPS information of other NeverBLIND devices
8 CPU Layer

8.1 Description
The central processing unit (CPU) layer is the core of NeverBLIND System where all of the information is gathered and processed before sending the data to the packet layer or I/O layer for further computations. The CPU layer consists of 2 subsystems: a data processor and an IFF component.

8.2 Purpose
The CPU layer contains 2 subsystems which are tied together to ensure information is computed correctly so that other layers can work properly using optimized information.

8.3 Function
The CPU layer acts as the central computing center for the NeverBLIND System. The user’s GPS location is collected from the sensing layer, computed before being sent to the packet layer. The CPU layer will also receive interpreted packets from the packet layer to be processed by the IFF component which transports that information to the I/O layer.
8.4 Dependencies

The CPU layer requires data received from the sensing layer and packet layer to be accurate. The internal data protocol will be uploaded using an USB interface and the external data will be sent and received through the Xbee network.

8.5 Interfaces

<table>
<thead>
<tr>
<th>Type</th>
<th>Source</th>
<th>Destination</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal</td>
<td>GPS</td>
<td>Data processor</td>
<td>GPS location in the form of Position (x, y, z)</td>
</tr>
<tr>
<td>Internal</td>
<td>3-Axis compass</td>
<td>IFF component</td>
<td>3-Axis compass data in the form of Compass(o, h)</td>
</tr>
<tr>
<td>Internal</td>
<td>Packet interpreter</td>
<td>IFF component</td>
<td>Interpreted packet</td>
</tr>
<tr>
<td>Internal</td>
<td>Data processor</td>
<td>IFF component</td>
<td>Processed data</td>
</tr>
<tr>
<td>Internal</td>
<td>Data processor</td>
<td>Packet encoder</td>
<td>Processed data</td>
</tr>
<tr>
<td>Internal</td>
<td>IFF component</td>
<td>Debug log</td>
<td>Digital output</td>
</tr>
<tr>
<td>Internal</td>
<td>IFF component</td>
<td>Output</td>
<td>Discrete output</td>
</tr>
</tbody>
</table>

Table 8-1 CPU Layer Interfaces

8.6 Processing

The CPU layer will do all of the necessary calculations as the NeverBLIND System device sends and receives data. The CPU layer will get latitude and longitude positions from the GPS component in the position layer, convert them into x, y and z coordinates, optimize them using different moving estimation algorithms and send them to the packet encoder subsystem. At the same time, the CPU layer will combine the GPS data with the information received from the 3-axis compass component and use trigonometry to calculate the distance between the user and their target and the angles needed for further computation. The processed information will be distributed to both the IFF component and the packet encoder to send to the network layer for distribution to other NeverBLIND System devices and the I/O layer to notify the user of the NeverBLIND System.

8.7 Data

Data will consist of GPS location software object, compass software object, processed packets, interpreted packets and filtered packets.

8.8 Subsystems

8.8.1 Data Processor

8.8.1.1 Purpose

The purpose of the data processor is to get positional information of the user for calculation and optimization.
8.8.1.2 Function
The data processor will get information from the GPS component, convert them into appropriate units, optimize them using moving estimation algorithms and calculate the distance between the user and their target as well as directional angles needed to decide whether or not to trigger the system. The data processor will then send the processed information to the IFF component and packet encoder.

8.8.1.3 Dependencies
Data from GPS, data transfer rate and accuracy of estimation algorithm

8.8.1.4 Interfaces

<table>
<thead>
<tr>
<th>Source</th>
<th>Data</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPS</td>
<td>Position (x, y, z)</td>
<td>Data processor</td>
</tr>
<tr>
<td>Data processor</td>
<td>Processed data</td>
<td>Packet encoder</td>
</tr>
<tr>
<td>Data processor</td>
<td>Processed data</td>
<td>IFF component</td>
</tr>
</tbody>
</table>

Table 8-2 Data Processor Interfaces

8.8.1.5 Processing
Process all position information from GPS.

8.8.1.6 Data
GPS information

8.8.1.7 Processing
Encode the position of the user for transport to the sender component of the network layer.

8.8.1.8 Data
Packets and processed data

8.8.2 IFF Component

8.8.2.1 Purpose
The purpose of the IFF component take the optimized position of the user from the data processor, extrapolate the user’s line of sight based on the information gathered from the 3-axis compass and to send discrete output to the output component if a teammate falls within that line of sight. The IFF component will also output digital output to be used for debugging during development.
8.8.2.2 Function
Sends a discrete output to the output layer if a user’s teammate falls within the user’s line of sight

8.8.2.3 Dependencies
Processed data from the network layer and data processor

8.8.2.4 Interfaces

<table>
<thead>
<tr>
<th>Source</th>
<th>Data</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>IFF component</td>
<td>Discrete output</td>
<td>Output</td>
</tr>
<tr>
<td>IFF component</td>
<td>Digital output</td>
<td>Debug log</td>
</tr>
<tr>
<td>Data processor</td>
<td>Processed data</td>
<td>IFF component</td>
</tr>
<tr>
<td>3-axis compass</td>
<td>Compass(o, h)</td>
<td>IFF component</td>
</tr>
</tbody>
</table>

Table 8-3 IFF Component Interfaces

8.8.2.5 Processing
Extrapolates the user’s line of sight based on their position and information gathered from the 3-axis compass. Will then calculate if a teammate falls within that line of sight.

8.8.2.6 Data
Processed data, interpreted packets, Compass(o, h) and discrete and digital output
9 Operating System Dependencies

The NeverBLIND System is designed with a microcontroller, the Arduino, as a CPU that will do all the computation as well as send and receive data. The software will be developed on a computer and be compiled to executable files before sending them to the serial port of the microcontroller. Although the NeverBLIND System depends on computers and operating systems to compile and embed the code, the IDE for the Arduino is available for several operating systems, including: Windows, Mac and Linux, all of which will give Team Itus the same output as needed. In addition, the embedded code is sent directly to the microcontroller through the USB connection without any additional programmer kits which optimize compatibility problems.
10 Relationship Mapping

Figure 10-1 Data Flow between Layers and Subsystems
### 10.1 Overview

Relationship mapping describes the data flow between the layers and subsystems of the NeverBLIND System.

### 10.2 Producer-Consumer Relationship

The data elements flowing between the different layers and subsystems are defined in Table 9-1. The producer-consumer relationship is described in Table 9-2. The producer is the source of the data element, while the consumer is the destination of the data element.

<table>
<thead>
<tr>
<th>Data Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Data packet containing the ID and position of another NeverBLIND device.</td>
</tr>
<tr>
<td>2</td>
<td>Data packet containing the ID and position of the user NeverBLIND device sent to the network layer of another NeverBLIND device.</td>
</tr>
<tr>
<td>3</td>
<td>Data packet containing the ID and position of another NeverBLIND device to be decoded by the packet decoder in the CPU layer.</td>
</tr>
<tr>
<td>4</td>
<td>Encoded data packet containing the ID and position of the user NeverBLIND device transported to the sender in the network layer.</td>
</tr>
<tr>
<td>5</td>
<td>Decoded packet containing the ID and position of another NeverBLIND device to be interpreted by the packet interpreter.</td>
</tr>
<tr>
<td>6</td>
<td>Discrete input containing the team ID of the NeverBLIND device to be encoded by the packet encoder.</td>
</tr>
<tr>
<td>7</td>
<td>Discrete input containing the team ID of the NeverBLIND device used to filter the packets received from the receiver.</td>
</tr>
<tr>
<td>8</td>
<td>Interpreted packet containing ID and position in a computer readable format.</td>
</tr>
<tr>
<td>9</td>
<td>Processed data containing the position of the user NeverBLIND device to be encoded by the packet encoder to send to another NeverBLIND device.</td>
</tr>
<tr>
<td>10</td>
<td>Discrete output produced by the IFF component to be sent to the output component of the I/O layer.</td>
</tr>
<tr>
<td>11</td>
<td>Processed data that contains the optimized position of the user to be used by the IFF component.</td>
</tr>
<tr>
<td>12</td>
<td>Digital output produced by the IFF component to be used for debugging during the development process.</td>
</tr>
<tr>
<td>13</td>
<td>Software object produced by the GPS component in the sensing layer in the form of Position(x, y, z) to be processed by the data processor in the CPU layer.</td>
</tr>
<tr>
<td>14</td>
<td>Software object produced by the 3-axis compass component in the sensing layer in the form of Compass(o, h) to be processed by the IFF component in the CPU layer.</td>
</tr>
<tr>
<td>15</td>
<td>Raw data gathered from GPS satellites by the GPS component containing the longitude, latitude and altitude of the NeverBLIND device in the form of x, y and z.</td>
</tr>
</tbody>
</table>
Raw data gathered by the 3-axis compass component based on magnetic North containing the orientation and heading of the NeverBLIND device referred to as o and h.

### Table 10-1 Data Elements

<table>
<thead>
<tr>
<th>Producer</th>
<th>Consumer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Other NeverBLIND Device</td>
<td>Receiver</td>
</tr>
<tr>
<td>Receiver</td>
<td><em>sender</em></td>
</tr>
<tr>
<td>Sender</td>
<td>2</td>
</tr>
</tbody>
</table>

### Table 10-2 Producer-Consumer Relationship Matrix
11 Requirements Mapping

11.1 Purpose
Team Itus utilizes requirements mapping in order to verify that the architectural design satisfies all high level requirements defined in the system requirements specification. Table 10-1 defines which layers fulfill the high level requirements. High level requirements concerning packaging and deliverables have been omitted since they are primarily functional requirements.

<table>
<thead>
<tr>
<th>Number</th>
<th>Requirement</th>
<th>Network</th>
<th>Packet</th>
<th>CPU</th>
<th>I/O</th>
<th>Sensing</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Signal Strength</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.3</td>
<td>Transmitter</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.4</td>
<td>Receiver Device</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.5</td>
<td>Notification Device</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.1</td>
<td>Range</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>5.2</td>
<td>Response Time</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.1</td>
<td>On/Off Switch</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>6.2</td>
<td>On/Off Indicator</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>8.1</td>
<td>Team Identification Switch</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

Table 11-1 Requirements Mapping
12 Testing Considerations

12.1 Overview
The system architecture design will be tested by Team Itus to verify that the NeverBLIND System fulfills the requirements specified in the System Requirements Specification, Architectural Design Specification and Detailed Design Specification. Each layer and subsystem component has been designed to be independent. The independence of each layer and subsystem will allow Team Itus to test them individually before implementation.

12.2 Testing Approach
Team Itus will test each subsystem component independently to verify that it performs its purpose correctly before implementing it into its respective layer.

After each subsystem has been installed into a layer, the layer will then be tested as a whole to verify that it performs its purpose correctly before integration with the other layers.

Team Itus will perform unit, component, integration and system testing to ensure that the NeverBLIND System performs as expected.

Team Itus will check to make sure all inputs and outputs are valid and that the data flows as expected between the various subsystems, layers and other NeverBLIND System devices.

12.3 Network Layer
The network layer must be able to transport packets between other NeverBLIND devices and its own CPU layer. In order to test that the network layer is functioning properly, Team Itus will ensure that the network layer is able to receive and send test data.

12.3.1 Receiver
Verify that the receiver is able to receive packets from other NeverBLIND devices and transport them to the packet decoder.

12.3.2 Sender
Verify that the sender is able to receive packets from the packet encoder and transport them to other NeverBLIND devices.

12.4 Sensing Layer
The sensing layer must be able to gather information from the GPS and 3-axis compass components and calculate the correct position, orientation and heading of the NeverBLIND
device. The raw data must then be transformed into software objects before being transported to the data processor.

12.4.1 GPS Component
Verify that the GPS component is able to gather the correct location of the NeverBLIND device in comparison to a known working GPS device.

12.4.2 3-Axis Compass Component
Verify that the 3-axis compass component provides the correction orientation and heading of the NeverBLIND device in comparison to a known working compass.

12.5 I/O Layer
The I/O layer must be able to convert discrete values from the input component into software objects to be transported to the data processor. The I/O layer must also be able to receive software objects from the IFF component and transform them into discrete analog outputs and digital output.

12.5.1 Output Component
Verify that the output component converts software objects and outputs them correctly by sending the output component test data.

12.5.2 Input Component
Verify that the input component receives discrete values from the user and converts them into software objects correctly by sending test data to the input component.

12.5.3 Debug Log Component
Verify that the debug log component converts software objects and outputs them correctly by sending debug log output component test data.

12.6 Packet Layer
The packet layer must be able to receive processed data from the data processor and encode them before sending them to the sender component. It must also be able to receive packets from the receiver in the network layer, decode and filter them based on team ID and interpret the packets into a computer readable format.

12.6.1 Packet Encoder
Verify that the packet encoder receives processed data from the data processor and the team ID from the input component and encode the packet correctly before transporting to the sender.
12.6.2 Packet Decoder
Verify that the packet decoder receives the packet from the receiver, decodes the packet and filters them based on team ID before transporting the filtered packets to the packet interpreter.

12.6.3 Packet Interpreter
Verify that the packet interpreter receives filtered packets from the packet decoder and transforms the packets into a computer readable format before transporting the interpreted packets to the data processor.

12.7 CPU Layer
The CPU layer must be able to handle all necessary computations for the NeverBLIND System given information from the sensing and packet layer. In order to test that the CPU layer is functioning properly, test data will be sent into the CPU layer and checked for proper output.

12.7.1 Data Processor
Verify that the data processor receives position information from sensing layer and processes data correctly before sending the data to the packet encoder and IFF component.

12.7.2 IFF Component
Verify that the IFF component receives information from the data processor and 3-axis component and calculates whether or not a teammate falls within the user’s line of sight correctly.