# Implementing NB-IoT: Communication with a Loading Cell

Johannes Almroth

3rd September 2019

## Abstract

The purpose of this project is to establish a line of communication between a loading cell and the internet. This will be done through the NB-IoT technology, and the data being sent is the one being produced by the loading cell. Using a development board from PyCom and a SIM-card from Telia as the network service provider, [...]

# Contents

# Chapter 1

# Introduction

Naming identifiers in a effective and efficient way is important for code comprehension. Variables such as $x$ and $y$ are not enough to convey context as well as meaning, and to write code that encapsulates complex mental models of thinking that need to be understood by multiple people in a development team, appropriate identifier names need to be chosen[4; 3; 7]. Good identifiers increases code readability, and that in turn is critical for the abstraction of concepts, collaboration and code maintainability[1; 7]. This is put into perspective when reading a paper by Deissenboeck *et al.* [5] which states that 70% of the source code for a given release of the popular Java IDE program *Eclipse* consisted solely of identifiers (measured in characters). Given the importance of identifiers, the question remains of *how* they can be written to further aid in code comprehension.

The two main styles of writing identifiers today are camelCasing (*e.g.,* coolBeans) and underscore casing (*e.g.,* cool_beans). Deciding which one to use is often a matter of the given convention within the programming language that the code is written in. As noted in a previous study on the subject [8], early programming languages such as Basic, Cobol, Fortran and Ada were case-insensitive, and thus encouraged the use of either the underscore or hyphens to write variables. When languages such as C and Java were introduced, camel casing became more common, and the argument can be made that it requires less effort to type, and thus improves writing speed. A paper by Epelboim *et al.* [6] found that un-spaced text (*i.e.,* hellothere) lead to a 10–20% slower reading speed in subjects, and would thus suggest that camel cased identifiers should be harder to read than snake cased identifiers.

As of today there's been two studies aimed at answering the question which one of camel casing or underscore casing affects reading comprehension the most. These are papers by Binkley *et al.* [2] and Bonita *et al.* [8] conducted in 2009 and 2010 respectively. The findings of these studies will be presented in [...]

## 1.1 Purpose and Goals

The study will primarily replicate the 2010 paper. This study aims to provide additional data to the questions posed in the 2009 and 2010 papers, *e.g.,* how does camelCasing and snake_casing affect code comprehension. The data gathering equipment used will be in the form of an eye-tracker. Another goal of the study is to achieve a solid statistical significance through a high enough subject count. Additionally, a few extra variables about the subjects will be gathered pertaining to previous programming experiences as well as their identifier style of choice, as to expand the

given research and to contribute additional data to the [...] (research area?)

## 1.2 Delimitations

This study will only use the same test format as presented in the 2010 paper.

# Chapter 2

# Background

So far there's been two studies aiming to determine if there is a significant difference between camel casing and underscore casing. The first study conducted by Binkley *et al.*in 2009 [2] with 135 subjects concluded that the camel casing style leads to a better all round performance, at least when the subject is trained on the style, despite taking on average 0.42 seconds longer to read. The second study conducted by Sharif *et al.*in 2010 [8] with the help of eye-tracking equipment found that camel cased words took on average 0.932 seconds longer to read, and concludes that the under score style leads to an improvement in both reading time and lessens the amount of visual effort required by the subjects. Visual effort is denoted by how long a subject stares at a given part of the screen, and how often this gaze wanders and switches focus. Even though the two studies differ in their conclusions, the eye-tracking study suffers from a small sample size, with a meagre 15 subjects compared to the 135 subjects in the 2009 study. However, the eye-tracking equipment used in the 2010 study lends a lot of credibility to the data presented, in contrast to the 2009 study where some subjects participated via an online interface in a non-controlled environment that could potentially obfuscate data.

# Chapter 3

# Methodology

# Chapter 4

# Results

4.1 Correctness and Find Time

4.2 Visual Effort

4.3 Similarities and Differences

# Chapter 5

# Discussion

Chapter 6

# Threats to Validity

# Chapter 7

# Conclusions & Future Work

## 7.1 Future Work

# Bibliography


[1] K.K. Aggarwal, Y. Singh, and J.K Chhabra. An integrated measure of software maintainability. IEEE, August 2002. doi: 10.1109/RAMS.2002.981648.

[2] Dave Binkley, Marcia Davis, Dawn Lawrie, and Christopher Morrel. To camelcase or under_score. IEEE, May 2009. doi: 10.1109/ICPC.2009.5090039.

[3] Raymond P.L. Buse and Westley R. Weimer. A metric for software readability. ACM, July 2008. doi: 10.1145/1390630.1390647.

[4] Simon Butler, Michel Wermelinger, Yijun Yu, and Helen Sharp. Relating identifier naming flaws and code quality: An empirical study. IEEE, October 2009. doi: 10.1109/WCRE.2009.50.

[5] Florian Deissenboeck and Markus Pizka. Concise and consistent naming. *Software Qual Journal*, 14(3):261–282, 2006. doi: 10.1007/s11219-006-9219-1. eclipse.org.

[6] Julie Epelboim, James R. Booth, Rebecca Ashkenazy, Arash Taleghani, and Robert M. Steinman. Fillers and spaces in text: The importance of word recognition during reading. *Vision Research*, 37(20):2899–2914, 1997. doi: 10.1016/S0042-6989(97)00095-3.

[7] D. Lawrie, H. Feild C. Morrell, and D. Binkley. What's in a name? a study of identifiers. IEEE, June 2006. doi: 10.1109/ICPC.2006.51.

[8] Bonita Sharif and Jonathan I. Maletic. An eye tracking study on camelcase and under_score identifier styles. IEEE, June-July 2010. doi: 10.1109/ICPC.2010.41.