**Project Deliverable J: User Manual**

GNG 2101 – Intro. to Product Dev. and Mgmt. for Engineers

Faculty of Engineering – University of Ottawa

**Onboard+ User Manual**

**Due Date:**

10 December 2020

**Submitted by:**

E32 - GNG2101

Jacob Kellett

Ranbir Brar

Khai Tran

Neal Anzengruber

**Abstract**

Onboard+ is an extension of Onboard (<https://launchpad.net/onboard>) which is an on-screen keyboard developed for a client and their patients at a hospital as a part of a project in GNG2101. This manual will cover the features, functions and capabilities of the Onboard+ along its development, instructions and troubleshooting. The main focus of Onboard+ is to improve user experience for patients with both physical and cognitive disabilities, this is done by implementing futures that focused on ease of use and functionality. The key aspect of this project was implementing macro menu functionality, which included easy creation of a macro that had the ability to access applications and websites in an instance of a click.

**Table of Contents**

[Abstract](https://docs.google.com/document/d/1e62V-ZlHwXt8mE3L1xl_KACA__7XDZoQ/edit#heading=h.4d34og8) …...……….………………………………………………………………………………….1

Table of Contents ……….………………………………………………………………………………2

Introduction…..…….…………………………………………………………………………………3

User Manual…….…………………………………………………………………………………..3-5

i. Features and Capabilities…….……………………………………………………………….....3

ii. Set-Up…………………………………………………………………………………………...4

iii. Code ………………………………………………………………………………………….5-6

iv Troubleshooting ………………………………………………………………………………6-7

v. Operation and Maintenance Guide……………………………………………………………….7

vi. Health and SafetyGuidelines……………………………………………………………………7

[Conclusions and Recommendations for Future Work](https://docs.google.com/document/d/1e62V-ZlHwXt8mE3L1xl_KACA__7XDZoQ/edit#heading=h.3rdcrjn) ……………………………………………………..8

Bibliography …………………………………………...……………………………………………….......8

Appendices …………………………………………………………………………………………………8

# 

**Introduction**

Many patients admitted to the hospital are required to stay for a couple of days, to possibly a couple of months. For a number of patients, their only access to online applications and communication platforms was through an outdated and difficult to use linux operating system; most of these patients have not even used a linux-based system before. In addition to this impractical system, a few of the patients have accessibility issues which include motor control, visual and/or other impairments. Their stay at the hospital is long and hard with this limited accessibility. Many of these patients are struggling to learn and use their technologies due to the keyboards and user interfaces being too complicated and not suited to their needs.

The main requirement for these patients with limited accessibility is technology that has an easy to use user interface along with functionality to facilitate their use of a new computer system they are not familiar with. Currently, there are no onscreen keyboards which are designed specifically for ease of access for patients with disabilities.

Onboard+’s design implements features and accessibility options to the existing Onboard keyboard including the addition of easily programmable macro buttons for both websites and applications while maintaining existing capabilities including hover click, word prediction, joystick support and customizable menus. Using a simple copy and paste format, our users are given the opportunity to keep all their favourite programs stored within the macro menu allowing for ease of access throughout their entire linux system.

**User Manual**

**Features and Capabilities:**

* Features a macro menu with 16 customizable buttons
  + each macro button can be programmed to open an application, website or snippet
  + additional buttons implemented onto macro menu creation menu to save as an application, website or snippet

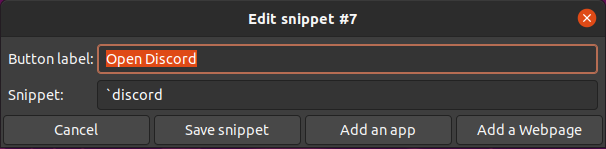


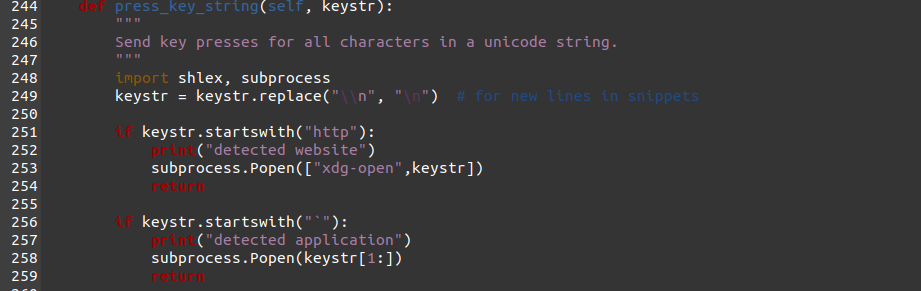
Figure 1. Extra Buttons to Save Macro as an Application or Webpage

* Controller and Joystick supported (along with other input devices support by Linux); features a customizable hover select option for input devices without an ‘enter’ button
* Includes standard Onboard on-screen keyboard features and functions
  + standard QWERTY keyboard layout with numpad and function keys
  + features middle, right, double, drag and hover click
  + features word prediction
* Includes standard Onboard menu features and functions
  + customizable keyboard appearance and size
  + customizable keyboard button behaviour

**Set-Up:**

1. Open the Terminal
2. Type in “sudo apt-get install onboard” in the Terminal
   1. after entering your password, the installation should start automatically
3. Open Onboard to ensure it is properly installed
   1. open the Terminal
   2. type in “onboard”
   3. Onboard should launch, close the terminal to close Onboard
4. Locate the Onboard files
   1. typically, Onboard is located in your system’s distribution packages
   2. Computer\lib\python3\dist-packages\Onboard
5. Open the Terminal in the Onboard file
   1. it is also possible to locate Onboard manually through the Terminal
6. Type in “sudo chmod 777 \*.py” in the Terminal directing to Onboard
7. Locate the Onboard folder and drag and drop the files from Onboard+ directly into the Onboard directory
   1. accept to replace the old files with the new files with the same name
8. Open Onboard/Onboard+ by the Terminal
   1. Onboard and Onboard+ has successfully merged, and Onboard will now have Onboard+’s features

**Code:**

****Figure 2. Application and Website Macro Code in Python located in Keyboard.py

Onboard+ uses subprocess (contained in Python’s library) to open the Linux Terminal. In a Linux environment, users are able to open applications through the terminal by entering the name of the application’s executable file.

In Figure 2, line 256 will check for the keyword `, which is a flag to the program that represents that the macro is saved as an application. Onboard+ will then open the Linux Terminal, and input the desired program name and execute it.

In Figure 2, line 251 will check for the keyword “http” which signifies a website. Onboard will then execute the subprocess command and open the Linux to then launch the saved URL through the user’s default browser.

The implementation of these features appear easy; however, please consult the required documentation before committing any changes to anything within the Onboard file as these changes will directly affect Onboard, which is located in the user’s system’s distribution packages and could possibly cause fatal errors to Onboard.

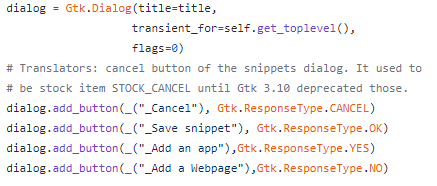


Figure 3. Creation of new buttons for a dialog in Python located in KeyboardWidget.py



Figure 4. Creation of Application and Website Macro Code in Python located in KeyboardWidget.py

The code shown in Figure 3 is the code responsible for the creation of the “Add an app” and “Add a webpage” buttons shown in Figure 1. These buttons will automatically save the user’s input as an application or website by applying their keywords. This feature has been implemented with error detection and will not add the application and website keyword if the keyword is already contained within the input already.

**Keywords for Creating Macros:**

` signifies an application

http signifies a webpage

**Troubleshooting:**

There are currently no known bugs for Onboard+, any bugs and problems the user may occur will be due to the standard Onboard software. In this case, please refer to <https://launchpad.net/onboard> for more information about the standard Onboard. It is possible to have errors while setting up Onboard+, the trouble shooting guide below will discuss the problems and solutions.

1. Macro: website not opening
   1. ensure a proper URL is being saved
   2. URLs typically begin with “https:”
   3. 

Figure 5. Functioning Example of a Website Macro

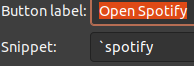
1. Macro: application not opening
   1. ensure the proper application name is being saved and the snippet starts with a `
   2. application names in Linux are typically all lowercase
   3. 

Figure 6. Functioning Example of an Application Macro

1. Input Device not recognized
   1. ensure the input device is compatible with the Linux OS and your system’s hardware - if the input device is compatible with Linux, it is with Onboard+
   2. ensure your input device’s firmware is up-to-date
2. Onboard is not opening
   1. Onboard is limited to one instance being open, to fix this, close all instances of onboard
      1. open terminal
      2. type “killall onboard”
      3. open onboard again
   2. The problem may stem from the system, do a system restart, or worst case, a clean reinstall of Onboard and Onboard+

**Operation and Maintenance Guide:**

Onboard is open-source and can be edited by any programmer. There are official updates that include improvements and bug fixes which can be found on <https://launchpad.net/onboard>. To update your Onboard, please consult LaunchPad or use Linxus’ “sudo apt-get upgrade Onboard” commands in the Linux Terminal.

**Health and Safety Guidelines:**

Onboard+ will not have any catastrophic failures and will not pose any risk to its users and anyone else. If a fatal error were to occur, it will cause Onboard+ to crash, and the host system will not be affected.

[**Conclusions and Recommendations for Future Work**](https://docs.google.com/document/d/1e62V-ZlHwXt8mE3L1xl_KACA__7XDZoQ/edit#heading=h.3rdcrjn)**:**

In conclusion, Onboard+ is a cost effective and easy to maintain solution to the problem that patients at the Saint-Vincent hospital face. While developing our prototypes for this product, we realized quickly that working with a Linux OS (such as Ubuntu) would be another learning curve. However, after learning the commands used in the Linux terminal, they were used to develop the prototypes and were definitely the most productive ways to implement functionality in the keyboard. Onboard+ has many functionality features, however there could definitely be more added accessibility features - a suggestion for future groups would be to focus on the accessibility of the software since the functionality features were heavily focused on by our team. We recommend that future developers become familiar with using python scripts in a Linux environment.

**Bibliography:**

**Appendices:**

Final Prototype’s BOM

There were no purchased materials for the final prototype of the OnBoard+ product. Controllers were used instead of joysticks - to test the software. These controllers can essentially mimic exactly what a joystick would do for a client. Finally the program will require a Linux operating system in order to be programmed. The cost for a Linux computer can be circumnavigated by downloading a virtual environment on all of the groups computers where a Linux OS can be utilized.