ISBN: 978-3-95450-221-9 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2021-THPV004

OPEN-HARDWARE KNOB SYSTEM FOR ACCELERATION CONTROL OPERATIONS

E. Munaron, M. Montis, L. Pranovi, INFN-LNL, Legnaro, Italy

Abstract

Nowadays technologies in LINAc facilities brought the common Human-Machine Interfaces (HMIs) to be more aligned to the standards coming from the information technology (IT) and the operators started to interact to the apparatus with the common computers' instruments: mouse and keyboard. This approach has both pros and cons. In order to minimize the cons and with the idea of providing an alternative to interact with HMIs, we tried to design and realize an open-hardware knob system solution.

INTRODUCTION

In the market, in addition to the standard devices (such as keyboard, mouse, and so on), there are several HMI controllers used to interact with personal computers. This kind of hardware is based on encoders and buttons and it needs dedicated drivers for specific operating systems (OSs) or software to work.

Sometimes software tools and applications adopted by the control system environment can introduce limitations and constraints in the usage of these commercial devices.

To overcome this limitation (in terms of drivers and OS interface), the need for a new technical solution would be suitable.

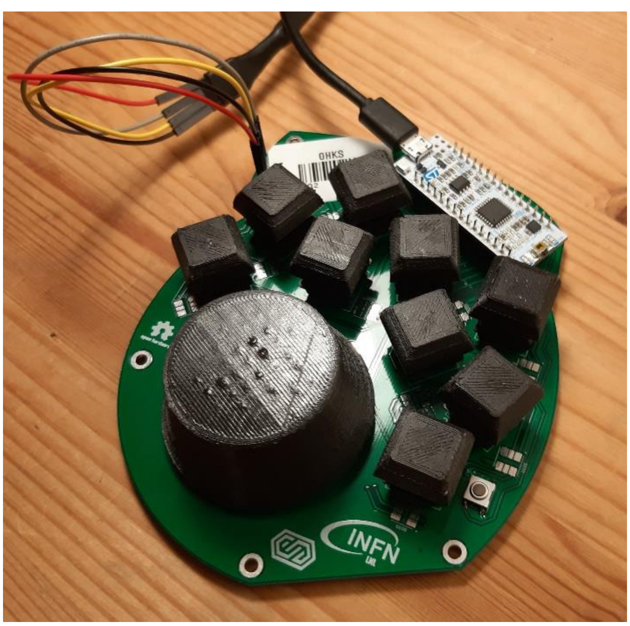

Figure 1: The open-hardware knob controller prototype.

THE OPEN-HARDWARE KNOB CONTROLLER

Because of the heterogeneous situation in terms of hardware and software solutions adopted for the different functional systems composing the facilities at INFN National Laboratories in Legnaro, the new controller should guarantee to be easily interfaced with several kinds of instruments and devices. According to this assumption and based on the talk by Kerry Scharfglass during KiCon 2019 [1], we started to study and design a new knob controller equipped with an encoder and buttons which has to achieve the following goals:

- Easy to configure
- Multi-platform
- Cost-optimized
- Adopt public licenses for both hardware and software

Unlike many proprietary solutions, this hardware interface can be recognized as a USB keyboard from any kind of device and is virtually compatible with any modern operating system. Key buttons and encoder can be programmable and dynamically reconfigurable by the user if needed.

In addition to the technical assumptions previously mentioned, we've chosen to release the project under Open Hardware License (OHL) [2] and GNU General Public License version 3 (GPLv3), so any developer or user interested in this work will be free to reproduce, develop, customize and share his product.

ELECTRONICS

The hardware (Figure 1) has been structured on a Printed Circuit Board (PCB) where the following parts are mounted:

- 9 Cherry-MX type keys
- 1 mechanical encoder
- 1 microcontroller

The keys are used to simulate, at the firmware level, the pressure of one keyboard button or a combination of them. Each key is equipped with anti-bounce circuitry and adequate ESD protection (Human-Body Model - HBM) to prevent a malfunction or breakdown of the electronics.

The microcontroller is devoted to managing the USB HID communication to the target system (such as a personal computer or a control device). To have multiple options in terms of hardware solutions for the microcontroller, the PCB has been realized to be compatible with both ARM STM32 microcontroller [3] and Teensy LC module [4]. This degree of freedom is very useful because it let developers select the preferred software toolchain for firmware implementation: for example, the Teensy LC module

ISBN: 978-3-95450-221-9 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2021-THPV004

is compatible with Arduino Integrated Development Environment (IDE) [5], the famous opensource solution used to write and upload programs to Arduino compatible boards or, with the help of third-party cores, other vendor development boards.

Following the open-source philosophy adopted for the entire project, the PCB schema has been built using KiCad open-source software [6].

MECHANICAL COMPONENTS

Under the mechanic aspect, the main goal the project wanted to achieve was to provide a complete system buildable from scratch, minimizing the cost in terms of components. For this reason, different parts have been designed to be realized through 3D printers and Fused Deposition Modeling (FDM) technology. Among the different 3D printing solutions, FDM technology has been considered the best choice because it is suitable for creating robust, durable, and dimensionally stable components, with the highest degree of accuracy and repeatability with low costs. As a matter of fact, 3D printers are quite diffused in several professional heterogeneous environments and among makers because of the commercial costs for devices, tools and raw materials.

Based on this choice, key covers and encored knob have been designed to be produced using FDM technology as visible in Figure 2, while the entire device cover has not been finalized due to the fact that additional ergonomics studies on the entire controller are ongoing.

The entire set of 3D models have been done with FreeCAD [7] (Figure 3), according to the philosophy adopted by the project.

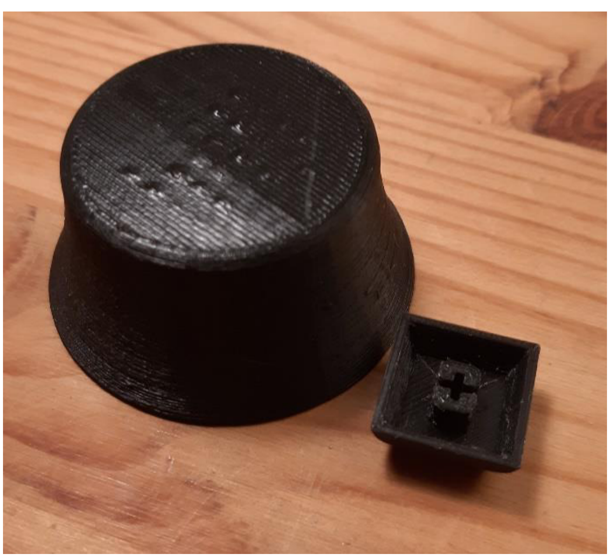

Figure 2: Mechanical parts realized with FDM technology.

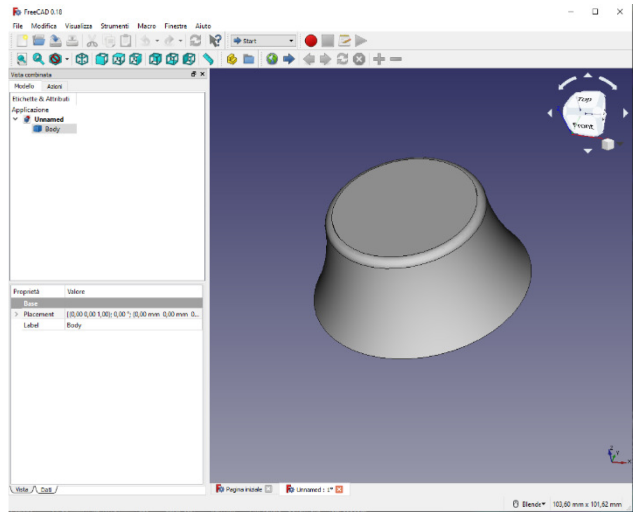

Figure 3: FreeCAD knob schematics.

FIRMWARE

Based on the microcontroller used, the preliminary firmware version has been defined using Arduino IDE and Teensyduino add-on [8] as software toolchain (see Figure 4).

This simple application is composed of the main Arduino sketches file (Arduino program language based on C/C++) and an external C header file. In particular:

- The main sketches file is devoted to establish the serial port communication and manage the events;
- The C library defines the maps for the key buttons and the knob controller. In this place it is possible, for example, to customize the response in case of buttons pressed combinations;
- Additional standard Arduino libraries.

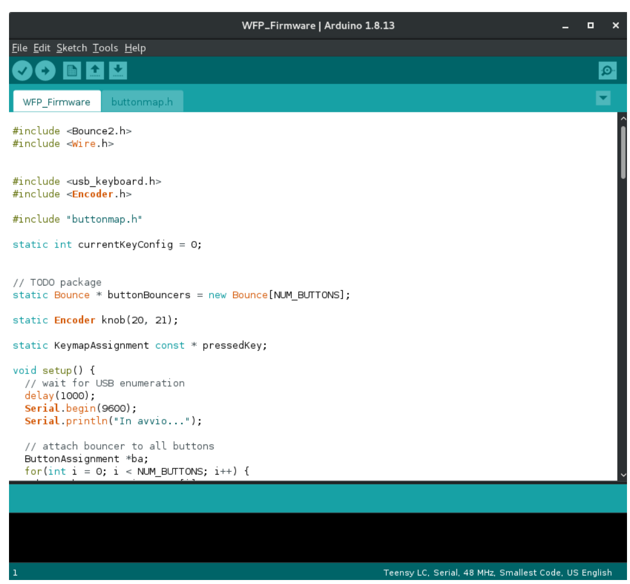

Figure 4: Firmware code realized with Arduino IDE and Teensyduino packages.

ISBN: 978-3-95450-221-9 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2021-THPV004

FIRST TESTS

Preliminary tests on different operative systems let us verify the correct functionality in terms of electronics and firmware: in every setup, the device has been automatically recognized as a USB keyboard. Trying to use a heterogeneous set of applications (word processors, draw applications, etc.), the device substituted the normal keyboard without any issue (according to the buttons map pre-loaded in the Firmware).

FURTHER DEVELOPMENTS

After the preliminary R&D performed and the test results obtained, further developments will be focused on two different areas of interest:

- Optimize the electronics and the mechanical parts to simplify schematics for PCB production and 3D printing processes
- Further investigations in ergonomics of the device, taking into account the feedback coming back from the beta testers, to realize a comfortable controller for end-users experience

In addition to that, a long-term goal the project wants to achieve is an advanced integration with the programs and software used for the control system; as consequence, deeper studies and tests with control system HMI applications will be performed, in order to adopt this hardware solution as a standard for INFN-LNL linear accelerators facility.

CONCLUSION

The need for a simpler and more intuitive solution for beam transport and experimental control has been the starting point to design and develop a new controller for linear accelerators.

The prototype realized is a multi-platform, easy to configure solution where most of the costs have been minimized through the usage of fast prototyping technologies, such as 3D printing. In addition to that, the product follows the open-source philosophy, and all the design parts are freely available under OHL and GPLv3 licenses.

First feedbacks coming from the test are very promising and let us confirm the technology assumptions made during the design stage; at the same time, further development will be done to optimize different aspects (electronics, mechanics, performances) and realize a more efficient and comfortable control device.

REFERENCES

[1] Kerry Scharfglass "Alternative User Interfaces for KiCad" KiCon 2019, `https://2019.kicad-kicon.com/talks/`

[2] OHL, `https://ohwr.org/cernohl`

[3] STM32 microcontrollers, `https://www.st.com/en/microcontrollers-microprocessors/stm32-32-bit-arm-cortex-mcus.html`

[4] Teensy LC, `https://www.pjrc.com/teensy/teensyLC.html`

[5] Arduino IDE, `https://www.arduino.cc/en/software`

[6] KiCad EDA, `https://www.kicad.org/`

[7] *FreeCAD*, `https://www.freecadweb.org/`

[8] Teensyduino package, `https://www.pjrc.com/teensy/teensyduino.html`