The Smart Skeleton: an open-source, interactive tool for teaching muscle actions and joint movements

John M. Pattillo
Department of Natural Sciences, Middle Georgia State University, Macon, Georgia

Abstract
This paper describes the design, construction, and use of an open-source hardware and software tool intended to help Anatomy and Physiology students test their knowledge of muscle actions and joint movements. Orientation sensors are attached to a model skeleton to turn the skeleton into an interactive, physical model for teaching limb movements. A detailed description of the construction of the tool is provided, as well as the configuration and use of companion software.

INTRODUCTION

Over the past decade, the open-source hardware movement has allowed nonexperts to create complex electronics and computing projects. Generally, an open-source hardware project combines ready-made microcontroller boards, sensors, actuators, and software with collaborative online support. These resources allow designers to combine hardware and software for a variety of purposes, including the creation of educational tools (1–3). This paper describes an open-source tool intended to help Anatomy and Physiology students test their knowledge of muscle actions and joint movements. These topics can be difficult, as they require students to visualize motion in three dimensions. There are a number of computer applications with animations to help students study muscle actions, but these are still essentially two-dimensional representations of three-dimensional concepts. A more valuable approach would closely mimic the real-world results of muscle contraction. Commonly available plastic teaching skeletons are useful for this purpose, as they can be manipulated to mimic natural joint movements. However, students must still imagine the presence of contracting muscles. The tool described here, named the “Smart Skeleton,” combines the manipulative aspects of a physical skeleton model with the flexibility and visual feedback of a computer application.

To construct the Smart Skeleton, a life-sized laboratory teaching skeleton is fitted with a set of orientation sensors. These provide a companion computer application with measurements of the three-dimensional orientations of the limb bones. The application prompts students to move the model so as to demonstrate their knowledge of joint movements or muscle actions. Data from the orientation sensors are then used to determine the correctness of motions and provide visual feedback to the student. An instructor may configure the application to drill students on any angular motion affecting the shoulder, elbow, wrist, hip, knee or ankle. Queries may include questions about motion terminology or the actions of 73 muscles or groups of muscles.

This project has been designed to be low-cost (approximately US$330 at the time of this writing) and relatively easy to build, configure, and modify. Only basic, through-hole soldering skills are required to construct the simplest configuration of the Smart Skeleton, and assembly of the project should take 3 to 6 h. This paper gives an overall view of the hardware and describes the use of the companion computer application. Detailed instructions for constructing the Smart Skeleton are provided in Supplemental Data S1 (see https://doi.org/10.6084/m9.figshare.14068673). A Github repository (Supplemental Data S2; see https://github.com/jmpattillo/smart-skeleton) contains all schematics, application files, and source code. In addition to the supplementary written instructions, a series of videos demonstrating the construction of the project is available (Supplemental Data S3; see https://youtube.com/playlist?list=PL05kkbUcbLboK99Vn05cJ2aeBMIQRTm3b). The author has endeavored to make construction of the Smart Skeleton accessible to non-experts. However, instructors who do not wish to undertake building the project themselves may wish to collaborate with colleagues or students in an engineering or computer science department.

HARDWARE OVERVIEW

The Smart Skeleton relies on seven orientation sensors to measure the three-dimensional orientation of the limbs and torso (4). The sensors are linked together in a short-range “i2c” serial bus (5, 6). USB cables are used to connect sensors and to create an extended serial network (7), distributed by several serial switches (8). The entire network is in turn...
managed by a microcontroller with built-in WiFi capability (9). Thus orientation data from each sensor can be sent wirelessly to a computer hosting the companion software. The microcontroller also acts as a WiFi host. This improves the reliability of orientation sensing, and may also prevent the host computer from wirelessly accessing the internet while the skeleton is in use. Figure 1 shows a simplified electrical schematic of the Smart Skeleton system.

The orientation sensors and serial switches are joined to custom printed circuit boards (PCBs) to create seven sensor modules. In addition to linking the orientation sensors to the serial bus, the custom PCBs also allow for power management, setting serial bus addresses, and connecting USB cables. Custom PCBs may be ordered online (10), or the PCB design files in the Github repository may be used to order PCBs from another supplier. The PCB files may also be modified using Autodesk Eagle, which is freely available to education customers (11). As described below, the Smart Skeleton software uses color-coded spheres as feedback to show students when joints have been moved beyond software-defined thresholds. During in-class testing, students were distracted by having to constantly look back and forth between the skeleton and the computer screen to determine when they had moved the skeleton sufficiently for the software to register their intent. Optional LED boards were added to the sensor modules address this problem. The LED boards are more difficult to construct, as they require surface-mount soldering. However, they are not needed for the sensor modules to function, and they may easily be added to the modules at any time after construction.

All modules are enclosed in custom, three-dimensional (3-D)-printed cases (Fig. 2, inset). The cases were designed in Autodesk Fusion 360, a cross-platform design suite freely available to education customers (11). The cases may be printed as is or customized if desired. Printable 3-D files (STL format) and Fusion 360 files for all 3-D-printed parts are available on the Github repository.

Six of the completed sensor modules are attached to the anterior of the left limbs of a skeleton (Fig. 2). The cases are mostly simple boxes that are easily affixed to the bones with cable ties. However, the case for the leg module is specially designed to rest over the anterior tibial crest, and the case for the foot module is designed to fixed to the metatarsals with a 3-D-printed, angled bracket. A seventh module is attached to the torso with Velcro (Fig. 2). The torso module is used as a reference so that overall motion of the skeleton (such as moving on casters) is not interpreted as movement of the limbs. As described below, the Velcro is needed to easily remove the torso module for calibration. All sensor modules are in turn connected to a main module consisting of a custom PCB joining the microcontroller, a serial switch, and power management components. The system is powered by a 5-volt DC supply. Any 5-volt supply should work, but it is convenient to use a common a rechargeable cell phone charger.

**SOFTWARE OVERVIEW**

The companion software prompts student to use the skeleton to perform predetermined joint actions and muscle movements. The application provides not only textual feedback but also uses detailed skeletal models to show a virtual representation of the movements of the physical skeleton (Fig. 3). These models were derived from those published by the National Bioscience Database Center of Japan (12). Compiled versions of the companion application for both

---

**Figure 1.** Simplified schematic showing the connection of different sensor modules and the topology of the i2c serial network. Power distribution on individual printed circuit boards (PCBs) and cable shielding connections have been omitted for clarity.
MacOS and Windows are available on the Github repository. The application was written using the Unity game development environment (13). Unity seemed an ideal choice because it has extensive graphical capabilities, has a robust support community, and is available at no cost for educational use. The software for the microcontroller was developed using the open-source Arduino development environment (14, 15). The Github repository contains all Unity and Arduino source code, and all Unity project files for any users who wish to modify the software.

**USING THE SMART SKELETON**

**Instructor Configuration**

Instructors configure the companion application by editing a comma separated value file ("questions.csv", included with the Github repository). This file contains fields to set the text of question prompts, correct answers, and feedback for correct and incorrect responses. Points awarded for correct responses can also be determined. The instructor may also specify that models of specific muscles or groups of muscles appear on the right limbs of the virtual skeleton (Fig. 3D). The muscles may be configured to appear immediately or to appear as a hint after a specified number of incorrect answers. A separate companion spreadsheet file (also included in the Github repository) contains a list of 73 numerical codes for specifying individual muscles or muscle groups.

During use, the Smart Skeleton software interprets the intent the user by setting flags when joint movements cross threshold angles. In this way, the software can represent a variety of joint positions as a single integer. When a student submits an answer, the integer representation of the skeleton’s position is compared with an integer representing the correct answer. The spreadsheet allows the instructor to easily calculate the “correct” integers representing desired positions. For example, to set the function the biceps femoris, a “1” would be entered in the appropriate cell for flexion of the knee and another “1” would be entered into the cell for hyperextension of the hip. All other cells would remain as “0”. The excel spreadsheet calculates the resulting integer value, which can then be entered into the answer cell in the "question.csv" file. In some cases, it may be useful to have more than one right answer to a muscle action prompt. For example, the instructor may wish to focus on the shoulder flexion and/or extension actions of the pectoralis major and ignore rotation. However, an enterprising student may be aware that the pectoralis major also acts to medially rotate the shoulder. By entering an alternate answer, the instructor may allow correct answers with and without shoulder rotation. Individual feedback may be specified for both primary and alternate correct answers. After the instructor has edited "questions.csv", the file must be saved in the same directory as the companion application.

**Startup Procedure**

To begin using the skeleton, the host computer is first connected to the Smart Skeleton WiFi access point, and the application is launched. When prompted by the application, the instructor or student follows onscreen instructions to move the skeleton, briefly holding the left limbs at several different, orthogonal positions (approximately two seconds per position). This procedure allows the orientation sensors to self-calibrate. The torso module must also be briefly removed from its Velcro mount and moved to several different positions. Once the calibration movements are complete, the skeleton is moved to anatomical position and joint...
Tracking is started by pressing the "R" key. If at any point joint tracking stalls during use, pressing the "R" key will restart tracking. The sensor modules have proven to be robust during in-class use. However, individual orientation sensors will occasionally fail to initialize properly after the power supply is connected. After power is applied, the microcontroller runs a self-test routine and the LED modules (if present) of any malfunctioning sensors will flash red. This situation can be corrected by forcing the sensors to reset by pressing the right Shift and "X" keys simultaneously. The sensors may also be reset by disconnecting and reconnecting the power supply. However, using this method will require the host computer to reconnect to the Smart Skeleton WiFi access point.

Student Use of the Smart Skeleton

As a student moves the skeleton, translucent spheres appear around joints when motions have crossed threshold angles (Fig. 3, A, C, and D). These spheres change color depending on the number of thresholds that have been crossed. For example, if the elbow joint is extended and supinated (as in anatomical position) then no sphere appears. If the elbow is flexed, then a blue sphere will appear around the joint to indicate that a single threshold has been crossed. If the elbow is both flexed and pronated, then the sphere will turn yellow to indicate that two thresholds have been crossed. In the rare case that a hip or shoulder motion would cross three thresholds, then the sphere around these joints will change to magenta. If LED modules are present, these will be illuminated so as to match the presence and color of the spheres around the joints (Fig. 3).

During use, the user's view of the virtual skeleton can be rotated using the left and right arrow keys. The up and down arrow keys pan the view vertically, so as to encompass the upper or lower portions of the body. The right and left bracket keys are used to zoom in and out, respectively.

Figure 3. The companion application in use. A: the software prompts the user to demonstrate a specific muscle action. B: the user responds by moving the skeleton so as to mimic the requested muscle action. In this case, the user has flexed both the shoulder and the elbow to mimic the action of the biceps brachii. A and B: colored spheres appear (A) and LED boards in the sensor modules illuminate (B) when preprogrammed movement thresholds are passed. In this case, both the spheres and the LEDs are blue to indicate that only 1 axis of motion has been used. Note that the user has configured the software to ignore movements of the wrist. C: the user has pressed the foot pedal to enter their answer and the software has indicated a correct answer. D: the user has entered an incorrect answer by only flexing the elbow. The instructor has configured the software to show the muscle on the right limb after a single wrong answer.
Changes of the user’s view of the virtual skeleton are especially useful for getting a satisfactory view of the lower limb and in closely examining muscle attachment points if the instructor has configured the software to show muscles.

Movements of some joints may cause other joints may be moved inadvertently. During in-class testing of the skeleton, students found this frustrating as they would have to hold several joints still while moving the intended joint. To eliminate this problem, a feature was added to allow specific joints to be ignored during movement. Ignoring the motion of the wrist, elbow, ankle, or knee can be toggled with specific keystrokes listed in the application window (Fig. 3A).

When a student has moved the skeleton so as to answer the question prompt, they hold the relevant limb still and tap a USB foot switch plugged into the host computer. The foot switch should be programmed to enter a lowercase “b” (answers may also be entered by pressing the “B” key on the keyboard). A student working alone would have their hands full with the limb of the skeleton, and so a foot switch offers a convenient method of entering an answer. If the answer is correct the question text turns green and the prespecified number of points are awarded. The application also displays the feedback text specified by the instructor. The student is then prompted to move the skeleton back to anatomical position and press the “N” key (Fig. 3C). This calibrates the skeleton on the screen to anatomical position, and advances to the next question. The skeleton may also be calibrated at any time by moving the limbs to anatomical position and pressing the “C” key. The orientation sensors will drift slightly over time, and so calibrating to anatomical position after each question helps keep motion tracking accurate. Drifting does not normally pose a problem for using the device, as the internal calibration of the orientation sensors improves as they are used. If an incorrect answer is given, the prespecified feedback appears, and the student is allowed to try again (Fig. 3D). The student may also use keyboard commands listed on the screen to skip questions or return to previous questions. However, once a question has been answered correctly the first time, no additional points will be awarded for correct answers. The text of correctly answered prompts remains green to remind students that they have already answered a question. If another student wishes to use the software, the instructor may reset the questions by pressing the left Shift and “R” keys simultaneously.

**Limitations**

In its current configuration, the Smart Skeleton can only be used for teaching any angular motions of the shoulder, elbow, wrist, hip, knee, and ankle. The muscle models the software can be configured to display are limited to those that cross these joints. The orientation sensors are unable to detect linear motions such as elevation of the shoulder. However, these motions are rarely possible with teaching skeletons. Movements of the digits are also not detected. Nevertheless, the software can currently be configured to teach most actions of 73 individual muscles and groups of muscles, as well as motion terminology for the larger joints of the limbs. The project could be expanded in the future to allow tracking of some other angular motions. For example, pelvis-mounted skeletons with flexible spines are commercially available, and so tracking of angular torso and neck motions could be added. Elevation and depression of the mandible could also be added. Indeed, the main module includes an extra sensor port (labeled “Head” in Fig. 1), which is intended for future expansion. Adding the joints of the digits could also be accomplished, although this would require developing a smaller set of sensors and modifying the joints to use springs to hold the bones together.

With the current version of the software, the Smart Skeleton is designed demonstrate movements beginning from anatomical position. Therefore, some muscle actions such as adduction of the limbs or supination of the forearm will not require a student to move the skeleton to demonstrate the correct answer. Students should be made aware of this possibility before using the skeleton. The instructor may also wish to edit the question configuration file to reinforce knowledge of these actions using feedback in the application. For example, the feedback shown in Fig. 3C confirms that the student has demonstrated all the actions of the biceps brachii, including supination. The software could be modified in the future to test from positions other than anatomical position.

**Choice of Model Skeleton**

In-class testing illuminated important considerations as to the choice of a model skeleton to use with the Smart Skeleton hardware. Skeletons mounted hanging from the skull are not ideal, as students are distracted by the task of keeping the skeleton from swaying during use. A skeleton mounted at the pelvis with locking casters is much more stable. While most teaching skeletons have articulations that allow realistic movement of most limb joints, many are limited in their ability to rotate the hip and shoulder from anatomical position. The author has found that it is usually possible to loosen the metal fixtures holding the hip and shoulder joints on inexpensive teaching skeletons so as to allow sufficient rotation of the shoulder and hip from anatomical position. If this modification is insufficient, plastic skeletons can be modified to use elastic cords at the shoulder and hip (suggested modifications are included in Supplemental Data S1; see https://doi.org/10.6084/m9.figshare.14068673). Most inexpensive teaching skeletons use wires to hold the bones and the feet together, and so inversion and eversion are usually possible. However, the ankle may lack the ability to be plantar flexed. Others have the ankle hanging normally in a plantar flexed position. The latter case is preferable, as ankle movement can be enhanced by holding the joint in anatomical position with an elastic band. This will allow a student to move the ankle in two directions in the sagittal plane so as to simulate both dorsiflexion and plantar flexion.

**CONCLUSIONS**

The Smart Skeleton provides a novel and exciting method for students to master their knowledge of joint movements and the muscle actions. Instructors may configure the project software to give prompts and feedback using whatever terminology or difficulty level they choose. Although requiring some skill in soldering, the hardware for the project is
relatively inexpensive to construct. All project files are open-source and may be freely used or modified. Although the project is currently limited to angular motions of the larger joints of the limbs, the author intends to continue to develop the Smart Skeleton to add additional features. Any software updates will be made available on the project Github repository.

**ACKNOWLEDGMENTS**

 Photographs in Supplemental Data S1 (Supplemental Figs. S1A, S1B, S2A, S2B, S3A, and S3B) are modified from originals by Adafruit Industries, LLC and are used by permission of Creative Commons license CC BY-SA 3.0 (https://creativecommons.org/licenses/by-sa/3.0/).

**GRANTS**

 This work was partially supported by a grant from the Middle Georgia State University Foundation.

**DISCLOSURES**

 No conflict of interest is declared by the author.

**AUTHOR CONTRIBUTIONS**

 J.M.P. conceived and designed research; J.M.P. performed experiments; J.M.P. analyzed data; J.M.P. interpreted results of experiments; J.M.P. prepared figures; J.M.P. drafted manuscript; J.M.P. edited and revised manuscript; J.M.P. approved final version of manuscript.

**REFERENCES**

1. Ganesh G, Mahalingam S, Annamalai G, Damodharan U. Seeing is believing: a demonstration of critical fusion frequency and its multidimensional nature. Adv Physiol Educ 41: 315–319, 2017. doi:10.1152/advan.00170.2016.
2. Heradio R, Chacón J, Vargas H, Galan D, Saenz J, de la Torre Cubillo L, Dormido S. Open-source hardware in education: a systematic mapping study. IEEE Access 6: 72094–72103, 2018. doi:10.1109/ACCESS.2018.2881929.
3. Landas-Jiménez MA, González-Gaspar P, Pérez-Estudillo C, López-Meraz ML, Morgado-Valle C, Beltran-Parrazal L. Open-box muscle-computer interface: introduction to human-computer interactions in bioengineering, physiology, and neuroscience courses. Adv Physiol Educ 40: 119–122, 2016. doi:10.1152/advan.00009.2015.
4. Adafruit Industries, LLC. Adafruit BNO055 Absolute Orientation Sensor (Online). https://learn.adafruit.com/adafruit-bno055-absolute-orientation-sensor. [17 February 2021].
5. Arduino, LLC. Wire Library (Online). https://www.arduino.cc/en/reference/wire. [17 February 2021].
6. SparkFun Electronics, Inc. I2C Tutorial (Online). https://learn.sparkfun.com/tutorials/i2c/all. [17 February 2021].
7. Vasquez J. Taking the Leap Off Board: an Introduction to i2c over Long Wires (Online). https://hackaday.com/2017/02/08/taking-the-leap-off-board-an-introduction-to-i2c-over-long-wires/. [17 February 2021].
8. Adafruit Industries, LLC. Adafruit TCA9548A 1-to-8 I2C Multiplexer Breakout (Online). https://learn.adafruit.com/adafruit-tca9548a-1-to-8-i2c-multiplexer-breakout. [17 February 2021].
9. Adafruit Industries, LLC. Adafruit Feather HUZZAH ESP8266 (Online). https://learn.adafruit.com/adafruit-feather-huzzah-esp8266. [17 February 2021].
10. Oshpark, LLC. Profile for smartskeleton (Online). https://oshpark.com/profiles/smartskeleton. [17 February 2021].
11. Autodesk, Inc. Unlock Educational Access to Autodesk Products (Online) https://www.autodesk.com/education/edu-software/overview [17 February 2021].
12. Mitsuhashi N, Fujiwara K, Tamura T, Kawamoto S, Takagi T, Okubo K. BodyParts3D: 3D structure database for anatomical concepts. Nucleic Acids Res 37: D782–D785, 2009. doi:10.1093/nar/gkn613.
13. Unity Technologies ApS. Unity for Education (Online). https://unity.com/learn/education. [17 February 2021].
14. Arduino, LLC. Download the Arduino IDE (Online). https://www.arduino.cc/en/Main/Software. [17 February 2021].
15. Arduino, LLC. Installing Additional Arduino Libraries (Online). https://www.arduino.cc/en/guide/libraries. [17 February 2021].