ASEE 2022 ANNUAL CONFERENCE
Excellence Through Diversity
MINNEAPOLIS, MINNESOTA, JUNE 26TH-29TH, 2022 ASEE

Paper ID #37736

MATLAB Curriculum Based in Experimental Setups with Authentic Data Collection and Analysis Experiences

Brian Patrick O'connell (Associate Teaching Professor)

Brian O'Connell is an Associate teaching professor in the First-Year Engineering program at Northeastern University. His undergraduate degree in Mechanical Engineering came from the University of Massachusetts at Amherst in 2006. He then worked for Kollmorgen Electro/Optical as a mechanical engineer developing periscopes and optronic masts. In 2011, he returned to academia at Tufts University, earning his MS and Ph.D. in Mechanical Engineering for his work with low-cost educational technologies and his development and use of technologies to aid usage tracking in makerspaces to examine them as interactive learning environments. He joined Northeastern in 2017. As well as teaching first-year engineering courses, he continues to design new technologies and curricula for use in his own classroom as well as for K-12 engineering education outreach.

Jada Wong

A Mechanical Engineering student from Northeastern University with a minor in Architectural Engineering

MATLAB Curriculum Based in Experimental Setups with Authentic Data Collection and Analysis Experiences

ASEE 2022 Conference - DELOS Division – BYOE Session

Presenter information

Feel free to contact the presenter for help or ideas in applying these topics in your classroom or lab as well as more detailed drawings, models, code, and other resources to construct your own:

Brian O'Connell, Ph.D.
Associate Teaching Professor
Northeastern University – College of Engineering – First Year Engineering Program
b.oconnell@northeastern.edu

Summary

This demonstration features a suite of experimental setups designed and developed for use in a first-year engineering course as part of their MATLAB instruction. Each experiment involves a portable data collection unit and baseline data collection software to enable initial interaction with the hardware. They purposefully showcase experimental setup and data collection as part of their MATLAB curriculum to learn the software in a setting and context more in-line with how they will use it in future courses. The suite currently contains hardware for collecting UV levels and GPS data, listening for signal transmission through a simulated noisy environment, recording hand-eye reaction times based on randomized stimuli, and a desktop mechanical stress test.

The hardware enables students to collect the specific data they need for each experiment, containing all sensors and other necessary components in a rugged and compact self-contained unit. The size is such that they can easily transport to perform their experiments where convenient, like their dorm rooms. The cost of each averaged to less than $25 per student per group by taking advantage of low-cost systems like Arduino and rapid prototyping options typically available in most academic settings. Since the educational objective is to develop their skill and understanding in MATLAB, module assessment focuses on the quality of the scripts developed to analyze the collected data, not the quality of the experimentation or understanding of the underlying concepts. Those elements are heavily scaffolded though through provided software that:

- guides them through the experimental runs
- uploads their data point(s) to a database
- downloads the class dataset from that database

Students then apply their MATLAB lessons to organize and analyze the dataset by eliminating noise and extraneous data and visualization of findings.

Pedagogical Context

MATLAB is a matrix-based programming language and integrated development environment. The platform was designed specifically for engineers and scientists as a design and analysis tool. Their promotional material, examples, and tutorials focus on its use as a support tool in control systems, robotics, deep-learning, and more. Our institution utilizes it in a range of courses, all primarily based in engineering labs, used in conjunction with data collection in some experimental setup and then for the analysis and visualization of the collected data. Those courses rely heavily on our First-Year Program to introduce students to MATLAB, using us as a prerequisite for those labs with the expectation that students will effectively use MATLAB in those scenarios. Much of the introductory material for MATLAB focuses on simple implementations to learn programming concepts and arbitrary design challenges to practice those concepts, like building small games or implementing simple sorting tasks. They will use MATLAB in future lab courses to interact with external equipment or large datasets. Many students will not have experienced that context until then, requiring some cognitive shift in their association with MATLAB.

A primary goal in creating these experimental setups was to provide a real-world circumstance for students to engage with while developing their understanding and mastery of MATLAB. There has been growing support for the use of project-based learning for skills and, in particular, nesting those projects in real-world contexts[1, 2]. Situative learning practices are beneficial across many disciplines but of particular interest here by utilizing real-world datasets in programming courses [3-6]. This project looks to take that a step further by engaging the students in collecting those datasets through experimentation similar to that they will implement in future courses. To cite a popular film and growing cliché, it sets up a "Wax on, Wax off" experiential transfer of knowledge[7, 8]. The types of debugging and standard algorithms they will engage with under this context will be more applicable in future courses than the type of programming experience provided through game development or other non-related contexts.

Several other pedagogical considerations factored into the design, but all secondary to their use in a MATLAB curriculum. The cost was one of the primary considerations so that other institutions could implement designs with relative ease. A specific goal of a $25 maximum cost per unit per student was set and met, accounting for pairing and usage grouping. This is also why their construction uses many readily available COTS parts and components manufactured from common makerspace resources like 3d-printers and laser cutters. That served a secondary purpose of showcasing best design practices for those resources since the First-year engineering courses also include a design project that utilizes that equipment. Finally, we actively attempted to identify and develop experiments that would be applicable across a range of disciplines so that every student, no matter their engineering path, will have a chance at an at least partially familiar experiment in their future courses.

System Design

This MATLAB curriculum tool currently consists of 5 components; the four experimental setups and the central database. There also exists support software written in MATLAB; a class and live scripts for each experimental setup and lesson. The MATLAB class provides the functions necessary for interacting with the experimental setups and uploading data points to the database, and downloading a data set as part of the MATLAB assignments. The provided MATLAB live scripts guide students through the experimental process and collect some data to share with the database, providing some context and understanding of the data they will be working with. It also gives them instructions on accessing the larger data set they will use as the primary input for the MATLAB script they create as part of their course assignments.

To clarify the student learning experience, students engage with these experiments purely through the data collection process. The provided scripts have all the code, through a MATLAB class with objects and methods for each experiment, for data collection, upload to the database, and accessing a historical dataset from the database. The students get to use those as blackbox commands with guidance for what inputs they should provide and the outputs they should expect. The purpose of the first-year engineering MATLAB curriculum is for them to learn the basics of MATLAB and introductory programming concepts. Establishing and controlling the hardware interactions as well as implementing some of the collection algorithms is beyond that learning objective. Therefore, they are provided with the commands they'll need for those elements and scaffolds, in the form of flowcharts and pseudocodes, for some of the common analysis algorithms for these types of data. Through implementation of those in guided lessons and a little more open-endedness in the homework, they can learn and implement programming concepts within the context of these experimental setups that aim to help retention by teaching them in the same context they'll utilize and expand on them later in their undergraduate career.

Database

The database uses MySQL, an open-source relational database management system currently hosted on Amazon web services. This does have an initial cost, and other options or educational discounts are being pursued. Each experiment has a table associated with it, the content of which is noted under each experiment's description. They all capture the student email associated with any upload and the timestamp. Maintenance is currently manually implemented, but maintenance scripts are in development.

UV Level

The UV Level module (See Figure 1) enables the collection of UV light data from multiple locations around campus. It represents a simplification of some location-based data collection labs from the environmental and civil engineering curriculum. The students can use it at either a predefined location with its longitude and latitude or by using the geolocation services of MATLAB mobile. Students then upload that data to the database, each data point being the UV light level, timestamp, longitude, and latitude. Upon retrieval of the data set, students use that information to engage with MATLABs graphical representation tools, plotting UV light levels in various ways based on time of day, time of year, and location.

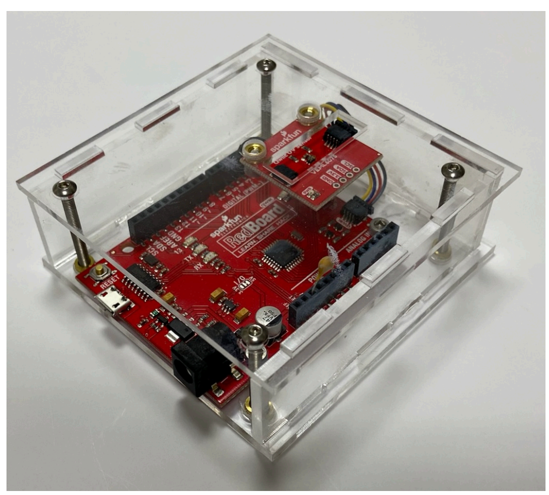

Figure 1. UV Module - Isometric view

It utilizes SparkFun's QWIIC UV Sensor and SparkFun's QWIIC RedBoard (See Figure 2). The QWIIC system simplifies the I2C communication between the components. The hardware itself is just those two components, connected and mounted in a protective enclosure that leaves the UV sensor exposed and access to the Arduino's USB port. A capacitor from reset to ground prohibits a student from overwriting the Arduino's script while still allowing for serial communication. The Arduino is programmed to wait for a request from a MATLAB script to take a sequence of UV light readings and return the average for a short period of time. If connected to MATLAB mobile on a student's phone, longitude and latitude can be collected or manually selected from predefined options. The latter option requires students to be at the predefined location.

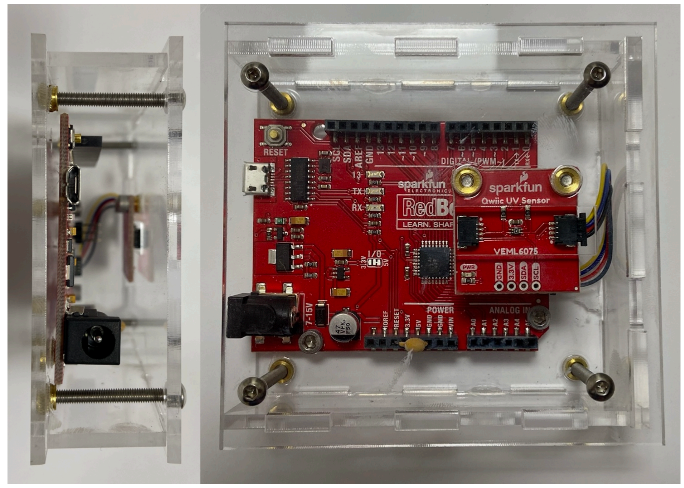

Figure 2. UV Module - Access and Sensor view

This module also specifically used a single add-on board and simplified enclosure design as a proof-of-concept for possible expansions. The SparkFun QUIIC system offers a range of data collection options, including other environmental sensors, gas sensors, accelerometers, and more. It is the simplest of the modules. Its design is easily modified to accept new sensor breakouts, making new modules limited by the cost and availability of those breakouts and the applicability of the data sources they open up access to.

Human Reaction Time

The human reaction time module (See Figure 3). is a modernization of the Fitts/Peterson 1964 motor response experiments[9], referenced in some of the university's bio-engineering courses. It tests the reaction time by having them touch a target and then, based on a visual stimulus, move to another target and return, recording the times involved. The hardware involves an Arduino Uno with an Adafruit capacitive touch sensor breakout board connected to 13 copper pads and 5

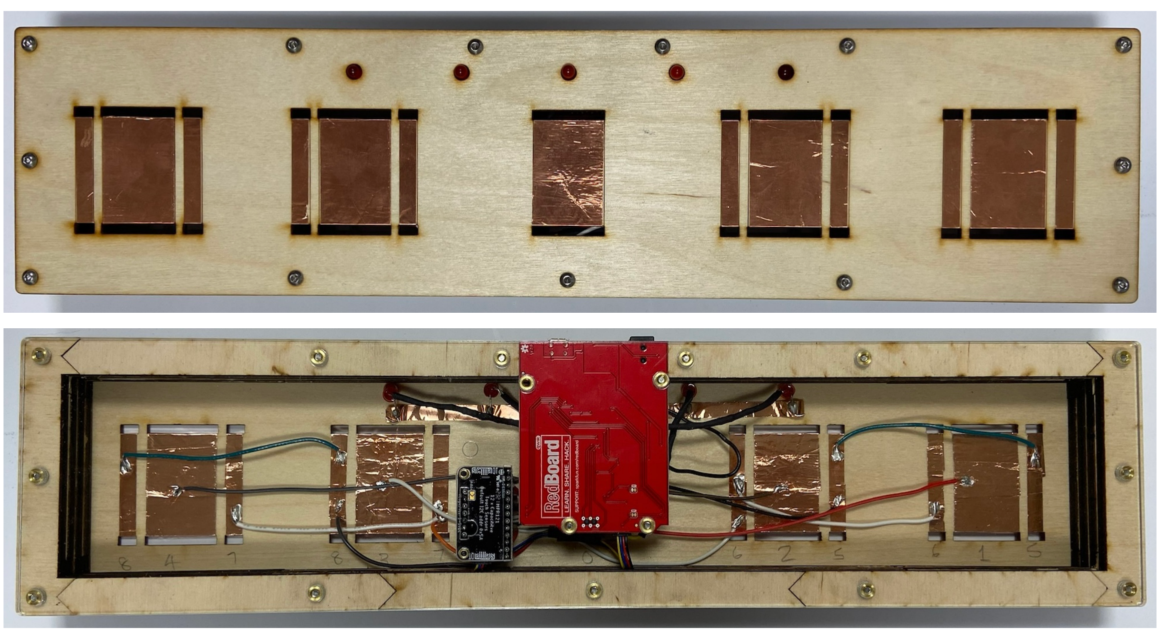

Figure 3. Reaction Time Module - Top and Rear View

LEDs. There is a central touchpad and two others on either side, with smaller pads on either side of each, spaced 3 inches apart each. The test subject puts their finger on the central pad while the center LED is on. Once it turns off, a short delay of random length will occur, and one of the other LEDs, randomly selected, will turn on. The subject is then to lift their finger from the central pad, touch the pad indicated by that LED, and return their finger to the central pad again. The Arduino captures three times of interest; the reaction time - how long it takes to lift a finger once the second light turns on, the movement time - how long it takes to touch one of the pads, and the return time - how long it takes to touch the central pad again. It also records the pad indicated and which was touched.

The MATLAB script triggers the start of an experiment by sending a request to the Arduino. When powered, the Arduino is actively listening to its serial connection. Once it receives that request, it controls much of the experiment process and then returns the results, writing them to the serial port connected to MATLAB. The collected data and some general subject data get uploaded to the database. The full dataset students work with for their assignment includes the experiment data and anonymized subject data for all entries collected by that class and the previous one. They then use MATLAB to organize and analyze that data, using organizational and statistical tools. Unknown to the students, the database contains a generated set of impossibly fast or slow data points and incomplete data points that they will need to filter out.

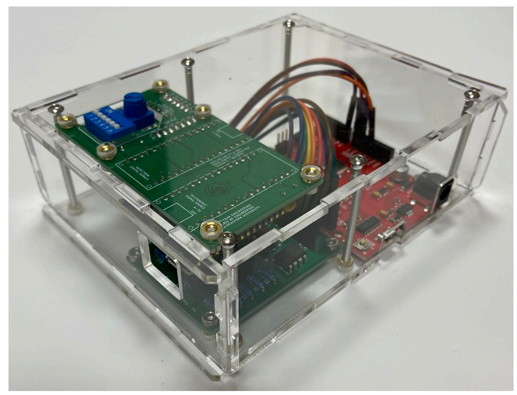

Figure 4. Isometric View of Signal Processing & Decryption Module

Signal Processing & Decryption

The signal processing & decryption module (See Figure 4) is a scaffolded variation on an electrical/systems engineering lab that involves the identification and decryption of a phantom signal. In the original lab, students receive an RF signal sending an encoded message, garbled due to analog interference from city infrastructure and digital interference from other signals in the area. Our module simplifies this for an introductory MATLAB experience by containing all of the signals into a single complex circuit that generates the signal, passes it

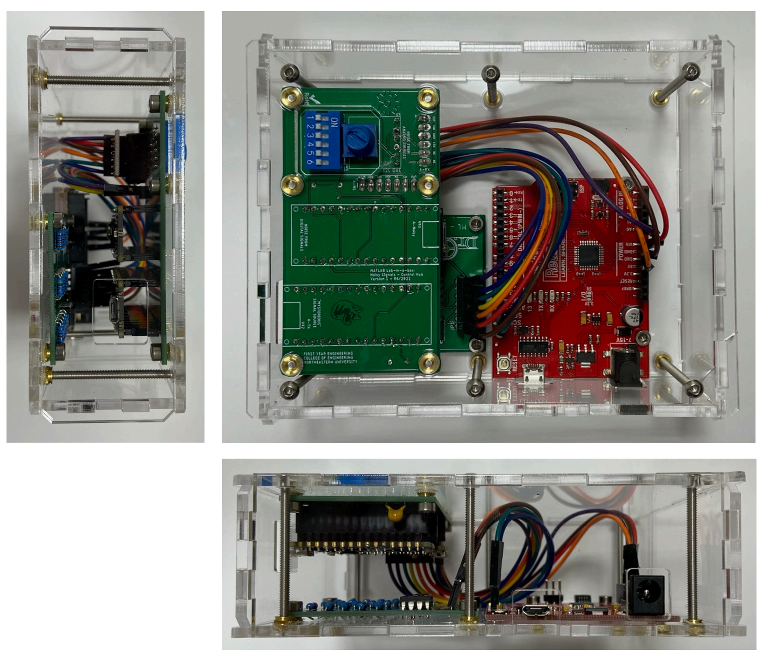

Figure 5. Signal Processing & Decryption Module: Side, Top, & Front View

through a noisy circuit, and provides an access point for the student to connect to through MATLAB and record the signal.

The hardware starts with an Arduino Nano Every generating a repeating signal, a phrase encoded into binary. It first travels through the noisy circuit. There, it interacts with a resistor array to simulate signal obstruction and degradation and a randomized switching element for small refractory effects. In addition, a second Arduino Nano Every generates randomized noise and, through a digital to analog converter, injects that into the resistor array. Finally, a third Arduino, a SparkFun RedBoard, receives the resulting signal and serves as the access point for the students to listen upon (See Figure 5). Due to the complexity of the circuits, custom PCBs help organize these components and simplify the manufacture of multiple units.

The MATLAB script enables the student to listen for that signal, setting a sample rate and the number of samples collected. The noise generating element includes a dip-switch and potentiometer that control some additional signals in the system, allowing students to experiment with additional known signals. The data point uploaded is the sample rate, experiment settings, and the collected sample of the signal. The MATLAB challenge is to use the available samples to decode the message. They will know the maximum message length and that a specific character combination always precedes the message. With that information, they can develop an algorithm to determine the start of the message within each sample and then reorganize and combine them to have a recognizable signal decode.

Mechanical Strength

The Mechanical Strength Testing module (See Figure 6) performs a common mechanical test utilized by many mechanical and civil engineering labs, a 3-point bending test. It allows for some stock material to be supported at its ends, and then a point source load is applied directly in the middle of those supports, bending the material until failure. Like the others, this is a

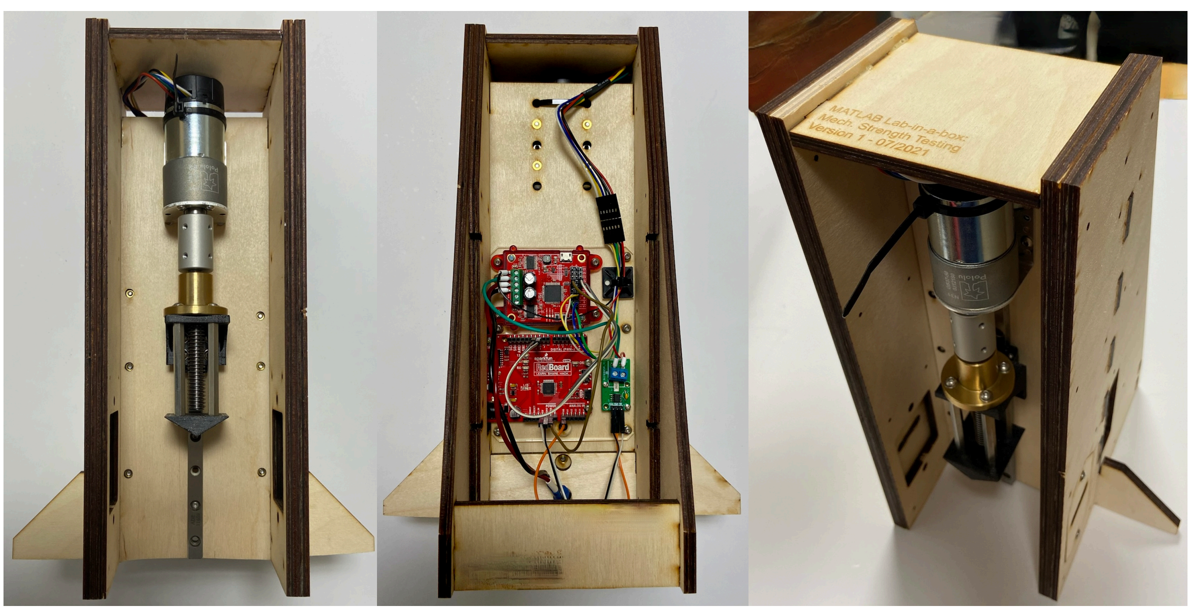

Figure 6. Mechanical Strength Testing Module: Front, Rear (with protective plate removed) and general view

simplified version of the traditional and more robust lab setup. It records the distance traveled by the point source via the encoder on the motor. Typical setups use a load cell on that point source to measure the applied force. However, this setup utilizes a current sensor on the motor, sensing the increase in current required by the motor as the material resists. This setup uses a commonly available gearmotor with an encoder and an intelligent motor controller to simplify setup and control independent of the student interactions. The use of current sensing required a precision current sense amplifier and the surface mount packaging options for the readily available devices necessitated a custom PCB to enable the interface.

The MATLAB script introduces the students to graphical user interfaces and allows for the experiment's control. The students input the material type, configuration, and cross-sectional area, then insert it into the module and start the experiment. While it runs, the module records position and current measurements for the full travel length of the point source. The MATLAB script uploads inputted material information and the collected measurements to the database. They then analyze the class measurements from the dataset, determine typical performance for the various materials, and determine appropriate filtering techniques to account for the noise from current sensing. This setup complements another project in the course by enabling them to collect data on uncommon building materials; various types and combinations of pasta for their pasta bridge design challenge. This experiment uses an appropriately scaled apparatus and a comparative material in MATLAB to influence their design decisions and model the likely performance of their bridge.

This module is the largest and most expensive of the 4. It is not intended to be as portable as the others due to the nature of the experiment experience it intends to simulate. While others are shared among 2-3 students and designed to be portable, this module is set up in an accessible location, specifically our first-year engineering student learning center, to be accessible and shared by many students.

Future Work/Concerns

As this is the first version of this suite, there are several intended improvements and concerns. The most significant improvement will be the expansion to utilize experiments from all the major disciplines. Northeastern University's First-Year Engineering Program combines students of all engineering majors. The intention is to eventually have an experimental option for each lesson so that students can pick which they'd like to engage with but achieve the same MATLAB learning goals even if they selected a civil engineering option or a bio-engineering option. This will be dependent on both examining the effectiveness of this initial suite regarding the MATLAB learning objectives and further research, something being examined through the inaugural implementation of this suite. Collaboration with the other departments will be used to determine useful experiments to emulate that fit well within the purposes of the suite. Particular focus will be put on '1 sensor' experiments like the UV Level experiment which specifically only utilizes a COTS sensor module and an Arduino to create a simple, small, and relatively inexpensive option within the suite. If other experiments that utilize just one sensor module can be identified, that creates some easy to implement options to expand the suite.

There also exist some concerns for the first implementation that will be paid particular close attention too. If these become issues or prove to be unfounded will determine how much they will influence future work. The first is with running these devices. There are some configuration issues that need to be addressed. The initial assignment includes instructions for a wide range of operating systems but, even when it was just for doing a baseline install of MATLAB, there are always a few issues. Feedback, requests for screen recording problems, and other assessment methods have been integrated into the assignment to identify and troubleshoot any issues that may arise and the instruction will be updated accordingly. The other concern is ruggedness of design. Durability and transportability was designed into the units but reality is reality. Each unit has been serialized and photographed. Assignment instruction includes how to handle and report damage, clarifying that it will not affect their grade but will just be used to update and redesign the units. All other major concerns are curricular and will be discussed in other publications more concerned with the pedagogical effectiveness of this approach.

Bill of Materials

UV Level Module

Part	Description	Source	P/N	Value	Qty	Cost
QWIIC RedBoard	Controller and MATLAB Connection	Sparkfun	15123	$16.95	1.0	$16.95
QWIIC UV Sensor	Senses Ultraviolet Light levels	Sparkfun	15089	$7.65	1.0	$7.65
QWIIC Cable	Connects QWIIC Enabled Components	Sparkfun	17260	$0.86	1.0	$0.86
3mm Acrylic	Case Material	Delvies Plastics	clear_acrylic_sheet	$4.46	0.5	$2.23
M2 Inserts	Heat set threaded insert	McMaster	94180A307	$0.18	6.0	$0.94
M3 Inserts	Heat set threaded insert	McMaster	95180A331	$0.16	4.0	$0.57
M2 Bolt - 8 mm	Electronics mounting	McMaster	92095A454	$0.26	6.0	$1.53
M3 Bolt - 15 mm	Enclosure Fastners	McMaster	92095A119	$0.17	4.0	$0.70
Threadlocker	Hardware Retention				A/R	$0.25
SuperGlue	Case Construction				A/R	$0.25
					Total:	$31.93

Human Reaction Time Module

Part	Description	Source	P/N	Value	Qty	Cost
QWIIC RedBoard	Controller and MATLAB Connection	Sparkfun	15123	$16.95	1.0	$16.95
Capacitive Touch Breakout	Enables capacitive touch sensing	Adafruit	1982	$7.95	1.0	$7.95
QWIIC Cable	Connects QWIIC Enabled Components	Sparkfun	17260	$0.86	1.0	$0.86
1" Copper Tape	Touch sensor pads				A/R	$0.50
1/4" Copper Tape	Over/undershoot pads				A/R	$0.50
M2 Inserts	Heat set threaded insert	McMaster	94180A307	$0.16	6.0	$0.94
M3 Inserts	Heat set threaded insert	McMaster	95180A331	$0.14	13.0	$1.86

Part	Description	Source	P/N	Value	Qty	Cost
M2 Bolt - 8 mm	Electronics mounting	McMaster	92095A454	$0.26	6.0	$1.53
M3 Bolt - 30 mm	Enclosure Fastners	McMaster	92095A119	$0.05	13.0	$0.64
3mm Acrylic	Rear panel	Delvies Plastics	clear_acrylic_sheet	$9.50	0.5	$4.75
1/4" plywood	Top Panel	Woodpecker Crafts	PLY-12-20-14	$4.75	0.3	$1.57
1/8" plywood	Side walls	Woodpecker Crafts	PLY-12-20-18	$3.75	1.0	$3.75
RED 5mm LED	Indicator Lights				5.0	$2.00
330 Ohm resistors	LED Ballast Resistors				5.0	$0.50
20 gauge wire	Sensor connections				A/R	$1.00
Shrink Tubing	Wiring Protection/Isolation				A/R	$0.50
Threadlocker	Hardware Retention				A/R	$0.25
SuperGlue	Case Construction				A/R	$0.25
					Total:	$46.30

Signal Processing & Decryption Module

Part	Description	Source	P/N	Value	Qty	Cost
QWIIC RedBoard	Controller and MATLAB Connection	Sparkfun	15123	$16.95	1.0	$16.95
Arduino Every Nano	Signal Source / Noise Source	Arduino	ABX00028	$12.90	2.0	$25.80
Custom PCB	Resistor Array	JLC PCB	N/A	$1.00	1.0	$1.00
Custom PCB	UI / Arduino Interface	JLC PCB	N/A	$1.50	1.0	$1.50
LM385n op-amp	Randomizing Signal Path	Amazon	B077BR9KT2	$0.50	1.0	$0.50
SPDT switch LCC110	Randomizing Signal State	Digikey	849-LCC110	$3.19	1.0	$3.19
Digital-to-Analog Converter	Converting Analog Noise	Sparkfun	12918	$4.73	1.0	$4.73
2k Ohm	Signal Interference				3.0	$0.20
3k Ohm	Signal Interference				1.0	$0.05
4k Ohm	Signal Interference				5.0	$0.25
6k Ohm	Signal Interference				3.0	$0.15
10k Ohm	Signal Interference				3.0	$0.15
10k potentiometer	Signal Control	Sparkfun	9806	$1.00	1.0	$1.00
6-pin dipswitch	Signal Control	Digikey	2223-DS01C-254-L-06BE-ND	$0.60	1.0	$0.60
12V Power Supply	Power	Mouser	562-QFWB-18-12-US01	$6.64	1.0	$6.64
3mm Acrylic	Rear panel	Delvies Plastics	clear_acrylic_sheet	$4.46	1.0	$4.46

Part	Description	Source	P/N	Value	Qty	Cost
M2 Inserts	Heat set threaded insert	McMaster	94180A307	$0.16	15.0	$2.36
M3 Inserts	Heat set threaded insert	McMaster	95180A331	$0.14	8.0	$1.14
M2 Bolt - 8 mm	Electronics mounting	McMaster	92095A454	$0.26	15.0	$3.83
M3 Bolt - 30 mm	Enclosure Fasteners	McMaster	92095A119	$0.05	6.0	$0.30
20 gauge wire	Sensor connections				A/R	$2.00
Threadlocker	Hardware Retention				A/R	$0.25
SuperGlue	Case Construction				A/R	$0.25
					Total:	$77.29

Mechanical Strength Module

Part	Description	Source	P/N	Value	Qty	Cost
QWIIC RedBoard	Controller and MATLAB Connection	Sparkfun	15123	$16.95	1.0	$16.95
50:1 Gearmotor w/ Encoder	Brushed DC Motor, 12V, 5.5A stall, 0.26 kg-cm	DigiKey	2183-4753-ND	$39.95	1.0	$39.95
2x7A Motor Driver	Serial Control Intelligent Motor Driver	BasicMicro	3284	$79.95	1.0	$79.95
12V Power Supply	Power	Mouser	562-QFWB-18-12-US01	$6.64	1.0	$6.64
Aluminum L-Bracket	Motor Mounting	DigiKey	2183-1084-ND	$3.98	1.0	$3.98
Linear Guide Rail	Guide rail for linear guide assembly	MISUMI	SHS15-180L(GK)	$88.00	1.0	$88.00
Linear Guide Block	Block with ball bearings for linear guide assembly	MISUMI	SRS15MUU(GK)	$62.00	1.0	$62.00
Lead Screw	2mm pitch, 89mm lg, screw with shaft on end	MISUMI	MTSRA10-89-S12-Q6-FC2-FW7-FY0-5	$26.27	1.0	$26.27
Lead Screw Nut	Round flanged brass nut, 2mm pitch, 10mm Dia.	MISUMI	MTSFR10	$17.79	1.0	$17.79
Shaft Couple	Rigid shaft couple for D10-D10 shafts, w set screws	MISUMI	CPRS20-6-6	$18.38	1.0	$18.38
Hex Spacer	Spacer to connect wedge to flanged nut	McMaster	94868A713	$4.03	4.0	$16.12
Custom PCB	Case Construction	JLCPCB	NA	$3.18	1.0	$3.18
Current Sense Amplifier	Amplifying current to measurable levels	DigiKey	296-48837-ND	$4.07	1.0	$4.07
50 mOhm Resistor	Current Sense Resistor	DigiKey	FCSL64R050FER-ND	$1.16	1.0	$1.16

Part	Description	Source	P/N	Value	Qty	Cost
100nF Capacitor	Stabilizer	DigiKey	56-K104K15X7RF53H5G-ND	$0.45	1.0	$0.45
3 Pos header	Interboard connection	DigiKey	SAM1029-03-ND	$0.41	1.0	$0.41
2 pos screw terminal	Motor Connection	DigiKey	A122765-ND	$1.04	1.0	$1.04
Panel Mount Barrel Jack	Power	DigiKey	2368-57-PJ80-ND	$1.75	1.0	$1.75
3mm Acrylic	Electronics Mounting	Delvies Plastics	clear_acrylic_sheet	$9.50	1.0	$9.50
1/4" plywood	Frame Components	Woodpecker Crafts	PLY-12-20-14	$4.75	2.0	$9.50
M2 Inserts	Heat set threaded insert	McMaster	94180A307	$0.16	6.0	$0.94
M3 Inserts	Captive threaded insert	McMaster	95180A331	$0.14	16.0	$2.28
M2 Bolt - 8 mm	Electronics mounting	McMaster	92095A454	$0.26	6.0	$1.53
M3 Bolt - 6 mm	Various Locations	McMaster	92095A179	$0.05	12.0	$0.61
M3 Bolt - 12 mm	Various Locations	McMaster	92095A119	$0.05	4.0	$0.20
20 gauge wire	Various connections				A/R	$1.00
Threadlocker	Hardware Retention				A/R	$0.25
Wood Glue	Frame Construction				A/R	$0.25
					Total:	$414.15

Works Cited

[1] K. Cennamo and D. Kalk, *Real world instructional design: An iterative approach to designing learning experiences*. Routledge, 2019.

[2] S. Avargil, O. Herscovitz, and Y. J. Dori, "Teaching thinking skills in context-based learning: Teachers' challenges and assessment knowledge," *Journal of science education and technology,* vol. 21, no. 2, pp. 207-225, 2012.

[3] R. E. Anderson, M. D. Ernst, R. Ordóñez, P. Pham, and S. A. Wolfman, "Introductory programming meets the real world: Using real problems and data in CS1," in *Proceedings of the 45th ACM technical symposium on Computer science education*, 2014, pp. 465-466.

[4] J.-I. Choi and M. Hannafin, "Situated cognition and learning environments: Roles, structures, and implications for design," *Educational technology research and development,* vol. 43, no. 2, pp. 53-69, 1995.

[5] A. C. Bart, R. Whitcomb, D. Kafura, C. A. Shaffer, and E. Tilevich, "Computing with corgis: Diverse, real-world datasets for introductory computing," *ACM Inroads,* vol. 8, no. 2, pp. 66-72, 2017.

[6] N. Nevaranta, P. Jaatinen, K. Gräsbeck, and O. Pyrhönen, "Interactive learning material for control engineering education using MATLAB live scripts," in *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, 2019, vol. 1: IEEE, pp. 1150-1154.

[7] F. Fonseca and L. Spence, "The karate kid method of problem based learning," in *Innovative practices in teaching information sciences and technology*: Springer, 2014, pp. 9-17.
[8] S. C. Leman and L. House, "Wax On, Wax Off: Relearning How to Teach Data Analysis. An Interactive Visual Approach."
[9] P. M. Fitts and J. R. Peterson, "Information capacity of discrete motor responses," *Journal of experimental psychology,* vol. 67, no. 2, p. 103, 1964.