Work In Progress: Addressing the Great Debate on Best Control Platforms in Mechanical Engineering

Dr. Lawrence Funke, Ohio Northern University

Dr. Funke received his PhD in Mechanical Engineering from the University of Notre Dame in 2017. He is currently an assistant professor at Ohio Northern University.

Dr. Maria-Isabel Carnasciali, Merrimack College

Maria-Isabel Carnasciali is the new founding Associate Dean of the School of Engineering and Computational Sciences at Merrimack College (MA). Previously, she spent 13 years at the University of New Haven (CT) where her last role included four years as Assistant Provost. She is Professor of Mechanical Engineering and enjoys teaching thermo/fluids/energy and design related courses.

Work In Progress: Addressing the Great Debate on
Best Control Platforms in Mechanical Engineering

Abstract

Controlling and monitoring mechatronic systems has become increasingly important in mechanical engineering and, therefore, needs to be addressed in the mechanical engineering curriculum. The rise of open-source compact platforms such as Arduino and Raspberry Pi has led to easier access and potential confusion on when to use which system. Arguments can frequently be heard in faculty meetings: "Arduino is the best!" "No! PLCs (Programmable Logic Controllers) are always the best option." "Well, actually, from my experience, NI (National Instruments) equipment works best." The reality is that almost any project can likely be "completed" by any of these platforms. As with all things in engineering, there are tradeoffs, and we should teach students to seek an optimal solution. What that optimal solution is, what metrics should be considered, and how they are weighted depends on the application at hand. The point of this work is not to advocate that one is clearly the best but rather that it is necessary for engineers and educators to introduce engineering students to multiple platforms and help them navigate how to select a platform for a given application. This paper explores a variety of potential metrics and how each platform performs in each metric. Illustrative examples from mechanical engineering courses and capstone projects are used to provide additional context. Examples include platform selection for an introduction to mechatronics course and a controller for a safety system for the Ohio Northern University (ONU) dive team (capstone project).

These metrics are combined into an easy-to-use and adaptable decision matrix that can be applied in a variety of contexts. It is presented with illustrative examples, and feedback on the utility of such a method is sought. Additionally, supervising personnel will come with their own set of knowledge, experiences, and potential biases. These can play a significant role in the process and need to be addressed. Strategies for mitigating the negative effects of this and harnessing the benefits of experience are also presented. After incorporating feedback from this work, the matrix will be presented, used, and assessed in classes at both Ohio Northern University and Merrimack College.

1 Introduction and literature review

In the realm of engineering education, the selection of controllers emerges as a cornerstone, shaping the academic discourse and practical insights imparted to the next generation of engineers. Knowledge of controllers equips mechanical engineers with the skills to design and work with modern, sophisticated systems, keeping them relevant and effective in a rapidly evolving technological landscape [1]. Modern mechanical systems often integrate mechanical components with electronics and control systems. Understanding controllers allows mechanical engineers to effectively design, implement, and troubleshoot these integrated systems. As industries move towards automation to improve efficiency and safety, knowledge of controllers is essential for designing and maintaining robotic systems and automated machinery.

Controllers can add functionality to mechanical systems, solve complex problems, and innovate new products. Understanding their capabilities allows mechanical engineers to push the boundaries of what's possible in their designs. Controllers can optimize the performance of mechanical systems, reducing energy consumption and operational costs. Engineers capable of implementing and tuning control systems can significantly enhance the efficiency and sustainability of their projects.

Choosing the *correct* type of system depends on factors like the complexity of the task, the environment in which it will operate, the need for scalability or future changes, and cost considerations. Controllers take many types, including data acquisition systems leveraging software such as NI DAQ leveraging LabVIEW, open-source prototyping microcontrollers such as Arduinos and Raspberry Pis, and programmable logic controllers (PLCs). Microcontrollers are often preferred for educational purposes, prototyping, or small-scale applications due to their versatility, low cost, and ease of use. The advent of open-source microcontrollers, like Arduino, has marked a revolutionary shift in engineering education and hobbyist projects alike. Figure 1 provides an overview of the prevalence of various controllers over the years based on published papers through ASEE. From 2015 to 2021, we see a large increase in the prevalence of Arduino and Arduino-like controllers. These accessible, versatile platforms have democratized the field of embedded systems, enabling students, educators, and DIY enthusiasts to bring interactive projects to life with relative ease. A search through the ASEE publications (peer.asee.org) reveals over 1000 papers involving Arduino kits published between 2002 and 2023 – with higher numbers published by the following divisions: Electrical and Computer Engineering (94), Engineering Technology (82), First-Year Programs (75), Manufacturing (67), Mechanical Engineering (46), and Pre-College Engineering Education (38). Within the papers published through Mechanical Engineering, authors highlight the advantages of the low-cost microcontrollers in integrating their use early on and throughout the four-year undergraduate curriculum (examples: [2-7]).

It should be noted that much of the literature reviewed refers to controllers/microcontrollers as tools or components in relation to other efforts (i.e. implementation of lab projects, capstone projects, etc). Few focus explicitly on the selection process or the training for students on effective selection given rapidly changing and available options. The paper by D'Souza et al. [8] provides guidance for selecting microcontrollers for undergraduate engineering capstone projects, with the goal of helping students minimize the time spent on this controller selection. A prior paper by one of the authors [5] described an effort to embed the usage of Arduino microcontrollers in four mechanical engineering courses. The integration aimed at providing students with a consistent and progressive learning experience regarding microcontrollers and their applications in engineering. This paper ventures into the intricate decisions of controller selection within educational frameworks, spotlighting its implications on the pedagogical strategies and learning outcomes in engineering curricula. Through some examples, we dissect the nuances and pedagogical considerations crucial in equipping future engineers with the knowledge and adaptability to excel in the ever-evolving technological landscape, with a special focus on mechanical engineering courses.

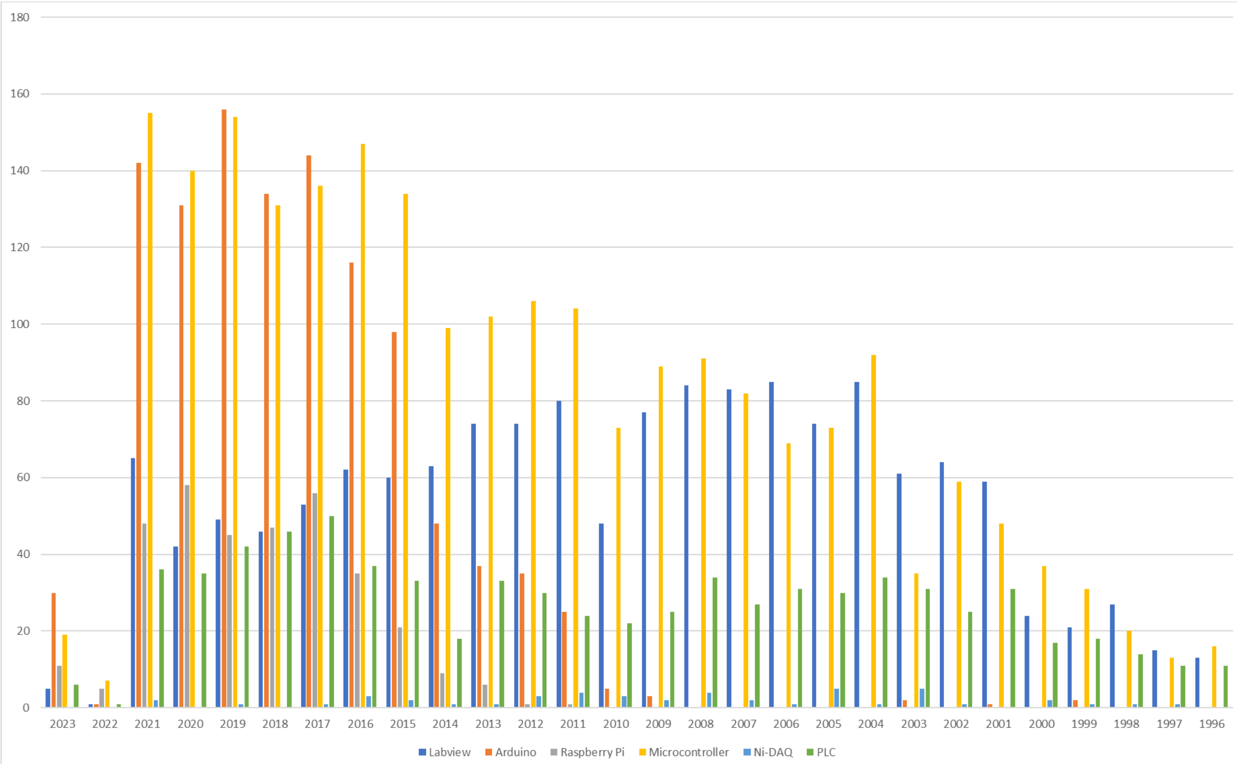

Figure 1: Count of papers published through ASEE involving the different controllers. Source: http://Peer.ASEE.Org compiled Fall 2023.

2 Introducing the Platforms

Adequately covering every possible data acquisition and control platform would require a textbook. Therefore, this section will focus on three categories of platforms, providing brief descriptions of each to orient the reader.

2.1 Arduino and other open-source prototyping tools

Arduino is probably the most well-known and used open-source microcontroller on the market. They are used throughout education from K-8 STEM classes through graduate engineering work and even in industry. The fact that they are open source means that there are several versions at both different price and quality points. Arduino is a microcontroller that can simply execute code to read inputs and produce outputs. Frequently its low price point and plethora of freely available user support are cited as its best qualities. On the other hand, its relatively low capabilities and frequently lower-quality construction are cited as shortcomings. It should be noted that Raspberry Pi and similar micro-computers alleviate some of these shortcomings, while maintaining a low price point, but are not different enough from Arduino to warrant their own category in this paper, and again, discussing all in-betweens would result in an intractably large number to consider. It is a good point to keep in mind that in-between options do frequently exist and can be explored after an initial assessment.

2.2 Data Acquisition (DAQ) platforms such as NI DAQ modules

Dedicated DAQ modules can be found in experimentation classes and industry laboratories around the world. These tend to be specialized for inputs receiving data from one or more sensors and would often have extra hardware incorporated into them to amplify, filter, convert to digital, etc., the signals coming in. National Instruments (NI) is a very common one that has a wide range of USB DAQ modules that can be easily integrated with PCs, especially if using their LabView program. NI also makes products focused on control, allowing for output, and many of their DAQ modules do have output capabilities, but frequently fewer than their inputs. Since these modules are specifically designed for sensor input, they tend to get very high quality data, and many companies, such as NI, have dedicated support teams to assist with technical issues. While a great benefit this also leads to the biggest con of NI that the equipment tends to be expensive especially when you also need to buy software licenses, and frequently have to pay for support packages if you want to take advantage of their experts.

2.3 Programmable Logic Controllers (PLC)

PLCs are the standard for process control and automation in the manufacturing industry. They are compatible with a wide variety of modules that can allow for inputs, outputs, communication protocols, etc. Frequently, they come on racks in cabinets where a new card needs to be purchased for each type of function required (analog input, analog output, digital input, digital output), however more recently, IO link has emerged where the same module can be used for both inputs and outputs, and multiple of these cards can be linked together. This seems to be quickly becoming a new standard and helps alleviate one of the largest negatives, which is that expanding a PLC's capabilities is generally expensive. One major advantage of PLCs is that they are designed for manufacturing environments, so they tend to be more robust and can come with different IP ratings to match the intended work zone. This makes them highly flexible, but also highly expensive. They frequently require their own software that is some combination of expensive and possibly requiring a learning curve.

3 Introducing the Decision Matrix

In the process of selecting an optimal control system for engineering applications, a comprehensive decision matrix could play an essential role in weighing various metrics against project-specific requirements and constraints. The proposed matrix includes factors such as expense, availability, resolution, speed, robustness, experience with the platform, compatibility, the necessity of additional cards/adapters, and software requirements. Each of these metrics plays a pivotal role in determining the suitability of systems ranging from cost-effective microcontrollers like Arduino to more robust and sophisticated options like PLCs or NI platforms. Understanding the nuanced impact of these factors, coupled with the subjective influence of personal experience and familiarity with certain platforms, is crucial in making an informed, balanced, and unbiased choice, as detailed in Table 1 and further expounded through practical examples in Section 4. The matrix and these issues will be presented to and discussed with capstone students at ONU during the 2024-2025 school year.

The following metrics are proposed to be included in the decision matrix:

- Expense - This can vary widely for the different systems and sometimes can make the choice in and of itself. For example, if the budget is a strict $100 total, then Arduino is the

only option. If $5,000 plus then there really is no reason not to get the more robust, widely known PLC or NI platforms that are more costly.
    - Expense should include not only the nominal platform, but also any extra cards/adapter required, and software.

- Availability - Fortunately, this seems to be less and less of an issue, but chip and supply shortages may make an option less or not available at all.

- Resolution - The resolution, especially of analog-to-digital converters (ADC) will vary widely, so the required resolution of the application should be considered.

- Speed - Similar to resolution, the speed, especially sample rate, processing time, and writing frequency, can have a significant impact on the decision, especially if the application involves real-time control.

- Robustness - In general, microcontroller platforms (ie arduino) are cheaper both monetarily and in quality, while PLC and similar are both more expensive and more reliable. Therefore, the required robustness (week long proof-of-concept vs needs to work in a remote harsh environment for 10 years, as extremes) may play a large role in the decision.

- Experience with platform (learning curve) - The experience and knowledge of both the people making the design decision and the stakeholders likely to use and maintain the system should be taken into consideration. How much time will it take them to become familiar with the platform? Will an additional expert need to be hired? There are several questions related to this that should be considered.

- Details on things showing up elsewhere.
    - Compatibility - This can refer to both ensuring that the platform is compatible electrically and mechanically with the inputs and outputs of other components in the system, as well as with the company, where certain companies have strict hardware requirements.
        - Influences expense and learning curve.
    - Software requirements - Some equipment requires specialized software that can be both expensive and difficult to learn, whereas others might be open source.
        - Impacts expense, and learning curve.

Clearly, the importance of these metrics and even whether they all should be considered or not is going to be highly dependent on the specific application. So, Table 1 combines them into a decision matrix, that is then used in examples from the authors' experience in Section 4 to illustrate how weighting, and if they are considered at all, changes application to application. This list and the examples could be presented to the students to show them how to use it before they are asked to use it in their capstone projects.

Table 1. Template for determining what platform is best for a given project.

			Scoring/Weighted Scoring					
Metric	*Quantity (units)*	*Weighting*	*Arduino***		*PLC****		*NI****	
Expense	USD $			0		0		0
Availability	Lead time (days)			0		0		0
Resolution	bits in ADC*			0		0		0
Speed	sample frequency *			0		0		0
Robustness	IP rating			0		0		0
Personal Experience	Time required to feel comfortable programming			0		0		0
	Total:	**0**		**0**		**0**		**0**

* quantity could be different for different applications, ie if doing many calculation may need processing time instead of sample frequency, or if using encoders, may need bits in counter instead of bits in ADC

** could be other microcontroller platforms such as raspberry pi, also Arduino nano vs mega vs uno could be split out

*** similarly PLC and NI could be hundreds of dollars (Automation Direct Click PLC or Allen-Bradley Micro820 or NI USB-6001) or thousands of dollars (Allen-Bradley Compactlogix PLC or NI USB-6211) so a specific model may need to be selected and added to the table

4 Applying the decision matrix to real-life case studies

This section looks at how this proposed framework could be used in multiple real-life case studies. The project is described, weights are assigned and defended, and then scores are given based on the described project. Some concluding discussion for each project and how the specific application influenced the results are presented.

4.1 Mechatronics course:

Author Funke developed an intro to Mechatronics course at ONU from scratch. The intention was that it be a hands-on, project-based course that introduced the students to the components that make mechatronic systems, focusing on sensors and actuators. Students needed to understand the basics of how these components work, how to determine which is best to use, and then combine and control them to achieve a stated goal. The instructor wanted it to be an experience-driven course where the students were encouraged to learn through failure and discover for themselves. The first lab was simply to make a DC motor spin in both directions at different speeds. No further guidance was given, simply the components they would need. The course culminated in the students designing and controlling a small conveyor system that incorporated three actuators and four or more sensors to move and sort wooden blocks based on what letter was on top of them. There was a $10,000 budget to buy all the equipment needed for up to 24 students to take the class. Kits that contained all of the hardware needed to do these projects were priced out and purchased. Arduino megas were chosen as the platform to control and interact with the sensors and actuators. This case study is now presented to determine if the proposed framework concurs with this decision.

The first step is to determine appropriate weights for each of the categories, remembering that the total weights should add up to 100:

- Expense - with less than $1000 for each kit and wanting to buy multiple motors and actuators for each of these, expense is critical and therefore weighted highest at 30.

- Availability - parts had to be ordered and on hand in about 60 days, so availability was important but not critical. On the other hand, as discussed in robustness, lab uptime is critically important, and a non-zero amount of equipment is expected to be destroyed each semester. Therefore, being able to replace equipment quickly is very important, resulting in a weight of 30.

- Resolution - Since this is for education, resolution is actually argued to be less critical. This is because it needs to be high enough to run reasonably, but issues with resolution, what they look like, and how to fix them are topics that can be beneficial to cover in a course like this. Therefore, if the device does not end up having a high enough resolution, this is seen less as a problem and more of a learning opportunity, therefore, it is weighted a 5.

- Speed - Similar to resolution, it needs to be fast enough to be able to control something, but if it is less than ideal, it is not a deal breaker, so it also receives a 5.

- Robustness - This is a tricky one for this application. The intention was to give students equipment and let them wire and learn how to use it on their own. This means it is likely that something will be wired incorrectly during the course. So here, robustness would mean that it either needs to be robust enough to survive being short circuited or that the entire system is robust to parts being damaged and can be easily replaced. Since short-circuit-proof equipment is not yet available, it was decided that replaceability was actually more important than traditional robustness, so this was weighted as a 10 and explains the higher rating on availability.

- Experience with platform (learning curve) - The instructor's experience was less important since they had time to develop it, and learning new platforms and programming languages tends to be easier for faculty. The real stakeholders here were the students. Since the point of the course was selecting and using hardware, not programming, a platform that more students were likely to have interacted with before the class was important. That said, the instructor could always provide skeleton codes and the internet has no shortage of examples for just about any platform. Therefore, this is important but not critical, so it is weighted 20.

Table 2. Pugh's method for determining the best platform for an Introduction to Mechatronics Class.

			Scoring/Weighted Scoring					
Metric	*Quantity (units)*	*Weighting*	*Arduino*		*PLC*		*NI*	
Expense	USD $	30	0	0	-1	-30	-2	-60
Availability	Lead time (days)	30	0	0	-1	-30	-1	-30
Resolution	bits in ADC	5	0	0	1	5	1	5
Speed	sample frequency	5	0	0	1	5	1	5
Robustness	IP rating	10	0	0	1	10	1	10
Personal Experience	Time required to feel comfortable programming	20	0	0	-1	-20	-1	-20
	Total:	**100**		**0**		**-60**		**-90**

Pugh's method was applied here where one option is picked as a baseline and given all zeros; the other options are then given a -2, -1, 0, 1, 2 in each metric where these are much worse, worse, same, better, much better, respectively. These are then totaled and the highest score is considered the winner. For this particular application, inexpensive and readily available were the most important aspects since many were needed, they were not all expected to last the entire semester, and extras needed to either be on hand or quickly available as backups. Despite decent quality PLCs being less than $200, the $40 price tag on an Arduino mega plus the availability of open-source software and code, made them the clear winner. This is born out in the numbers in Table 2. Granted both NI and PLC equipment would be better on the actual performance (resolution and speed) and robustness side of things; however, since this is meant to be a figure out how to make it work and when you fail, learn from it type class, these benefits were outweighed by the Arduino platform.

4.2 Diver Safety System for capstone project

Author Funke at ONU advised a capstone team that was tasked with designing a safety system for the school's dive team. When divers are learning new elements they often end up hitting the water hard and at weird angles. This can result in serious injury, especially from higher diving boards. Essentially, the team needed to design a system that could be added and removed from the pool that would release compressed air for a set period of time. The air bubbles to the surface breaking the water tension and essentially cushioning the diver. Both diver testimony and experimental data showed that the impact force was greatly reduced with this device. They needed to be able to trigger one of two bubbler systems on demand with a push of a button, hence a control system was needed. This was to be used by the school's dive team for as long as possible. The team originally planned to use Arduino for cost savings, but ended up using a click PLC for the increased robustness. This case study investigates if the team made the best decision.

Once again, appropriate weights for each of the categories were determined, summing to 100:

- Expense - with a total budget of $5,000 expense was important, but did not need to be minimized, so it was weighted as 20.

- Availability - while this product does deal with safety and excessive downtime is not desired, the team successfully competed for decades without it, so availability is not critically important, as long as it is not likely a part would become impossible to replace. Therefore, a weight of 10 is reasonable.

- Resolution - the only inputs are binary push buttons, so resolution does not apply and is removed from the table.

- Speed - the system does need to respond to inputs rapidly, but the design simply required a valve to be opened or closed based on the button input. The control system has no impact on how long it takes the valve to open or close, so the only speed here is how long in between the button being pressed and the signal to open or close the valve being sent. Since this is going to be on the order of a millisecond for all options, this metric is also removed from the table.

- Robustness - This is the most important aspect by far. The plan was for the control system to live in a storage closet on the pool deck and last several years without intervention. This means that it is in a relatively harsh environment with water and chlorine, will be moved around, and since the team is graduating needs little-to-no maintenance since the original designers will not be around to service it. This importance warrants a weighting of 50.

- Experience with platform (learning curve) - This could be looked at in two ways. Either the experience of the capstone team in coding it, or the experience of the people likely to service it. Since servicing it would likely fall to the faculty adviser after the students leave, the analysis will focus on the students since the faculty are familiar with all options. The intention is that it does not need serviced, but something always happens (the team's adviser actually did need to service it the following year), so this aspect is given a middle weighting of 20.

Table 3. Pugh's method for determining the best control platform for a diver safety system.

			Scoring/Weighted Scoring					
Metric	*Quantity (units)*	*Weighting*	*Arduino*		*PLC*		*NI*	
Expense	USD $	20	0	0	-1	-20	-2	-40
Availability	Lead time (days)	10	0	0	0	0	-1	-10
Robustness	IP rating	50	0	0	1	50	1	50
Personal Experience	Time required to feel comfortable programming	20	0	0	-1	-20	-1	-20
	Total:	**100**		**0**		**10**		**-20**

The same methodology was used as in the previous case study. PLC narrowly wins, and this also illustrates why the students initially thought they would use an Arduino. They had assumed that the PLC would be hundreds of dollars versus tens. However, the clickPLC price point is just under $100. Additionally, since the PLC can deal with 24VDC directly, there is no need for additional relays or power supplies whereas with an Arduino, a step down to 5VDC would have been needed, as well as a relay to drive the valves. Therefore, it could be argued that the expense for the PLC should be 0 once supporting equipment is considered. The robustness gains of minimizing the number of pieces of hardware needed and having the more secure, easier to protect connections of the PLC make it the clear winner. So, in the end, the students made the correct decision.

5 Conclusion and Future Work

This paper sought to develop a framework to assist engineers, students, faculty, and professionals determining the platform best for a specific application. The framework encourages the user to consider multiple aspects of the project at hand and to quantify both the importance of each aspect and the potential platform's ability to perform that aspect. This also allows for the consideration of biases that prior experience may introduce into the selection. This prior experience with a particular platform may be a benefit, especially if it is in a rapid prototyping phase and the device needs to be programmed as quickly as possible. On the other hand, if this is something that needs to last long term and will have multiple people working on it over the device's lifetime, then ease of use for all potential users needs to be considered, not just for the engineering doing the original programming. This work was largely motivated by the authors seeing students and colleagues struggle to determine when Arduinos are warranted and when they are not suitable, and sometimes not even being aware of the capabilities or existence of the different options. Therefore, the authors have presented this framework with some case studies to solicit feedback from the broader community. The plan is to take this feedback and present a modified version (both of the elements of the matrix and the examples of it being used) to the Senior capstone students at ONU to be used in the 2024-2025 school year to gain further data on its utility and further refine. Assessment will be carried out to determine the effectiveness of this approach. One method being considered is to administer pre- and post-surveys on controller selection knowledge and confidence. Additionally, the devices used in previous capstone projects will be compared to those projects that used this framework to see what, if any, trends can be found.

References

[1] D. Vyas, (2010), *Microcontrollers: options and trends in today's market*. ACM Proceedings International Conference and Workshop on Emerging Trends in Technology, Mumbai, India.

[2] Vasquez, H., & Fuentes, A. A. (2013, June), *Integration of Sensors and Low-Cost Microcontrollers into the Undergraduate Mechanical Engineering Design Sequence* Paper presented at 2013 ASEE Annual Conference & Exposition, Atlanta, Georgia. 10.18260/1-2—19803

[3] Dillon, H., & Schmedake, N., & Eifler, K. E., & Doughty, T. A., & Lulay, K. (2016, June), *Design of a Curriculum-Spanning Mechanical Engineering Laboratory Experiment* Paper presented at 2016 ASEE Annual Conference & Exposition, New Orleans, Louisiana. 10.18260/p.26679

[4] Dickrell, P. L. (2017, June), *Five-Minute Demonstrations: Minimal Faculty Investment for Maximum Learning Impact* Paper presented at 2017 ASEE Annual Conference & Exposition, Columbus, Ohio. 10.18260/1-2--28366

[5] Funke, L., & Hylton, J. B., & Sawyers, D. (2019, June), *Work in Progress: Incorporating Microprocessors across the Mechanical Engineering Curriculum* Paper presented at 2019 ASEE Annual Conference & Exposition, Tampa, Florida. 10.18260/1-2—33630

[6] Harib, K. H., & Sivaloganathan, S., & Hamza, R. K. M., & Aziz, M. A. (2020, June), *Foundation Mechatronics Laboratory Course for Mechanical Engineering Students* Paper presented at 2020 ASEE Virtual Annual Conference Content Access, Virtual On line. 10.18260/1-2--34687

[7] McPheron, B. D., & Parson, K. M., & Stein, M. (2021, April), *Development of a Mechatronics System Design Course* Paper presented at 2021 Illinois-Indiana Regional Conference, Virtual. 10.18260/1-2—38263

[8] D’Sauza, J.F., Reed, A. D., & Adams, C. K. (2014), *Selecting Microcontrollers and Development Tools for Undergraduate Engineering Capstone Projects.* Computers in Education Journal, vol. 5, pp54-66.