Sociedade Brasileira de Automática (SBA)
XXIV Congresso Brasileiro de Automática - CBA 2022, 16 a 19 de outubro de 2022

A Proposal for Expanding Temperature Control Lab Educational Kit to ESP32 Boards

Gustavo Pereira da Fonseca * **Josenalde Barbosa de Oliveira** *

* *Agricultural School of Jundiaí, Federal University of Rio Grande do Norte, RN 160 Km 03, 59280-000, RN, Brazil (e-mail: gustavo.fonseca.703@ufrn.edu.br, josenalde.oliveira@ufrn.br).*

AbstractThis paper presents a proposal to expand the Temperature Control Lab (TCLab) educational tool for the ESP32 embedded platform, with a software interface adapted to the Firmata protocol and Sysex commands. This migration adds ESP32 features as DAC, bluetooth, WiFi and RTOS compatibility with its dual core architecture. Besides validating the usual PID control experiments, a brain emotional learning algorithm (BELBIC) was simulated and deployed, aggretating new educational skills and perspective to TCLab continuous development.

Keywords: Reinforcement learning; BELBIC; Firmata; Intelligent Control; TCLab; Control education.

1. INTRODUCTION

The Temperature Control Lab (TCLab) is an open project with both hardware and software available (Rossiter et al., 2019), for control education purpose. Originated from demands for the process control teaching, it initially was based on the concept of take-home kit which had particular interest during remote classes, but it is equally applied for in-person laboratory experiments conducted by students due to its low cost (Oliveira and Hedengren, 2019; de Moura Oliveira et al., 2020) and ease of programming in Matlab/Simulink (Hacioglu et al., 2020) or Python, as an suitable alternative to commercial solutions. Its Arduino-based hardware with a printed circuit board of a simple thermal plant as compatible shield, brings a pocket size real system to the teaching-learning process (Figure 1), satisfying student demand for practical applications beyond theory which thus may evolve a set of competencies (Tupac-Yupanqui et al., 2022).

Figure 1. TCLab components: Arduino Leonardo, Thermal system shield, USB cable, External power source

With a transistor temperature as the process variable and a PWM base input as the control signal (Figure 2), a diversity of control related subjects can be explored, as modeling and identification, parameter optimization (Oliveira et al., 2020), linear and nonlinear control, machine learning, among other benchmark problems (Park et al., 2020). TCLab is commercially available and has already been successfully used by several universities throughout the USA, and some institutions in Europe (UK, Portugal), South Africa (Pretoria) and South America, where Herrera et al. (2020) deployed a dynamic sliding mode control for long delay systems. Indeed, it is already available a plenty of educational resources, such as video classes and tutorials as well as source code. However it is still an unknown teaching tool outside its origin universities and co-workers. This paper not only disseminates it for a broader audience, but proposes its software integration to a well known targeted Internet of Things (IoT) device ESP32 (ESPRESSIF, 2021), see Figure 3. Due to its higher availability and flexibility of ADC and GPIO pins, dual core, Wi-Fi and bluetooth native support (Junior et al., 2020; Misal et al., 2020) it has been increasingly chosen as a preferable platform. TCLab is based on a Arduino Leonardo board with 8-bit, 16 MHz ATmega32u4 microcontroller, which doubles the 10-bit ADC pins of the basic version Arduino UNO. The ESP32 version used in this paper is based on the XTensa® dual core 32-bit LX6, up to 240 MHz each core, 34 GPIO with up to 18 pins (12-bit ADC) and 2 8-bit channels DAC. This DAC is particularly interesting for control systems when a PWM input signal is not suitable for a specific plant.

Besides a first PID control validation replacing the processing board from Arduino to ESP32, a Brain Emotional Learning Based Intelligent Controller (BELBIC) (Lucas et al., 2004) was due simulated on the plant model based on the TCLab shield and then deployed to ESP32, expanding the existing control challenges and topics. These

ISSN: 2525-8311 DOI: 10.20906/CBA2022/3739

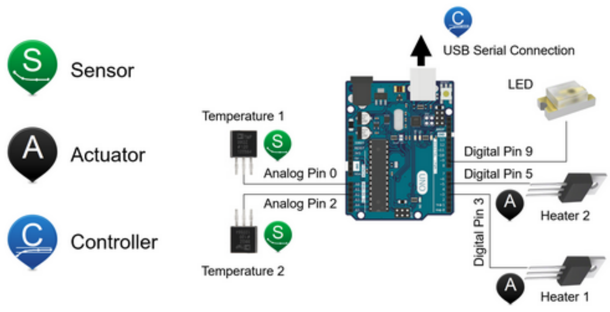

Figure 2. TCLab project structure

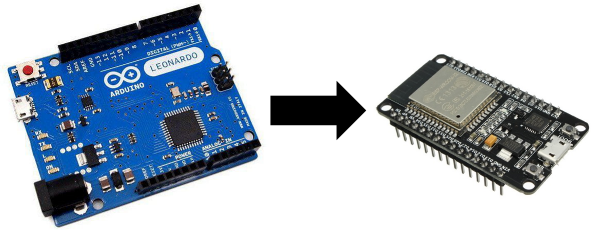

Figure 3. Replacing Arduino Leonardo by ESP32 platform

developments were carried out as final project by students of embedded systems at the undergratuate course on Analysis and Systems Development of Federal University of Rio Grande do Norte, Brazil. All tutorials already available at the TCLab repositories for the original Arduino Leonardo board and software may be evaluated on ESP32 with the open contributions of this paper.

2. ADAPTING AND RUNNING TCLAB TUTORIALS

2.1 Running on ESP32

TCLab core hardware is composed by two transistors TIP31C acting as heaters, and two temperature sensors TMP36. The aim of control is to maintain a desired temperature at set point by adjustment of heater power output. As stated by Herrera et al. (2020), the thermal energy from heater is transferred by conduction, convection, and radiation to the temperature sensor, since each pair sensor-transistor is bounded with thermal paste. Although Multiple Input Multiple Output (MIMO) experiments can be conducted, in this work only a Single Input and Single Output (SISO) is used. The serial communication between the desktop software and TCLab board is specifically handled in Python with the *pyserial* library. Here the focus is Python language instead of proprietary solutions, since the low cost nature is not only in the hardware but in the software as well.

Due to the unavailability of the TCLab plant shield at the laboratory, the open TCLab schematics in Park et al. (2020) was implemented in a protoboard with some adaptations (Figure 4), thus replicating the operation of the original TCLab. The temperature sensor TMP36 was replaced by the LM35DZ and a thermal pad was used for the bounding LM35DZ-TIP31C. Since only one TIP31C is active for the SISO case, an external power source of 5V/1A was enough to provide electrical current for the heater.

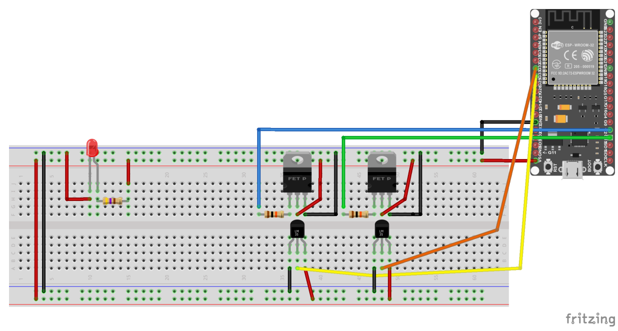

Figure 4. Thermal plant schematics in protoboard

2.2 Firmata-esp32 communication and framework

According to the UML component diagram depicted in Figure 5, the application algorithms (control, identification and so on) written in Python for the host computer, send and receive data through commands of a high level framework (here it was chosen the *pyFirmata*: https://pypi.org/project/pyFirmata/) which communicates with the ESP32 by means of the middleware protocol Firmata written in C++. Since the current Firmata version at https://github.com/firmata/protocol is only compatible with some Arduino boards and ESP8266, a modified version was developed for ESP32 target hardware (Oliveira, 2020), where the file *Boards.h* includes specific board specifications (number of pins and respective types, for instance). Instead of using the common analog and digital report commands, this interface is based on general System exclusive (sysex) messages, which are used to define sets of core features (digital and analog I/O). The general block diagram and basic functions are presented in Figure 6. With the Sysex protocol, data communication format uses MIDI messages.

Four commands were designed within *FirmataExt.cpp* for the *handleSysex* method: 0x01 (digital output), 0x02 (analog input, readTemp()), 0x03 (digital read) and 0x04 (analog output, sendPWM()). These interactions are presented in the sequence chart in Figure 7. Since sysex commands are based on 7-bit bytes (0-127), an encode-decode method is necessary for the commands 0x02 and 0x04 with ADC and PWM resolutions $\geq$ 8 bits, respectively. The source code may be found in Fonseca (2021).

In order to validate such architecture, Figures 9 and 8 present two initial proposed experiments for 1) regression-based First Order Plus Dead Time (FOPDT) model identification (Figure 8) and 2) a PID controller running on the host computer and communicating to the ESP32 board and the actual plant (9). These are the same original TCLab experiments adapted to ESP32. The input/output .csv file from the identification routine feeds the online tool PID Tuner, thus getting the parameters $k = 0.3495, \tau = 113.55, \theta = 31.94$ and the model

$$G_1(s) = \frac{k}{\tau s + 1} e^{-\theta s} = \frac{0.3495}{113.55s + 1} e^{-31.94s}. \quad (1)$$

ISSN: 2525-8311

DOI: 10.20906/CBA2022/3739

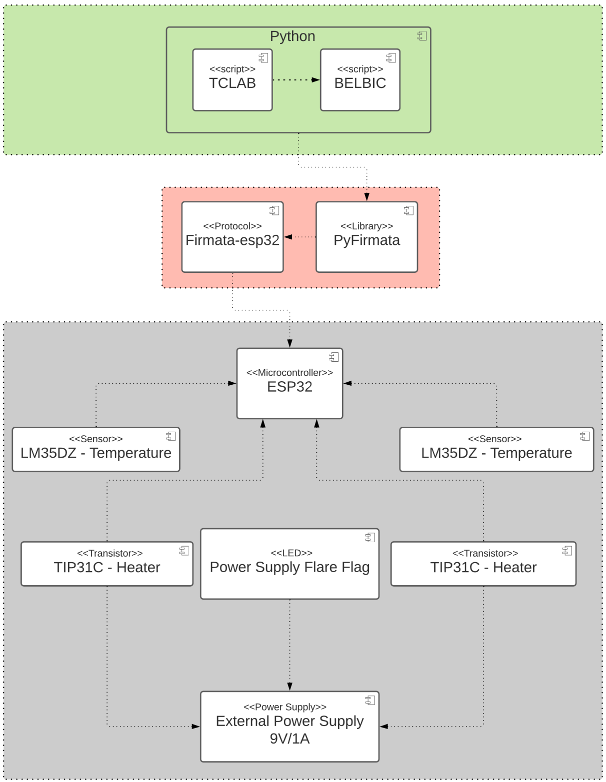

Figure 5. UML component diagram

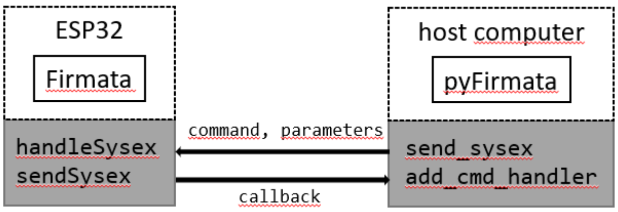

Figure 6. Modified Firmata protocol for ESP32 boards with Sysex commands

PID parameters are $K_p = 10, \tau_i = 50, \tau_d = 1$ with a time constant $T = 1s$. Once these experiments presented suitable results when compared to TCLab performance, the next step was the evaluation of an intelligent control experiment, thus adding another learning skills and control strategies.

2.3 BELBIC

The Brain Emotional Learning Based Intelligent Controller (BELBIC) is a reinforcement learning method (Lucas et al., 2004) which has gained increasing interest for control engineering applications (Beheshti and Hashim, 2010) and therefore its embedded deployments (Lucas, 2011) and simulation tools (Coelho et al., 2017). It computationally mimics the functional parts of the mammalian limbic system, such as the Amygdala (AM), Sensory Cortex (SC), Orbitofrontal Cortex (OC) and the Thalamus (TH), as depicted in Figure 10 (Beheshti and Hashim, 2010). The presence of OC and AM weights acting on the excitatory input signals (also named stimuli) suggests similarities with neural networks, but BELBIC is simpler

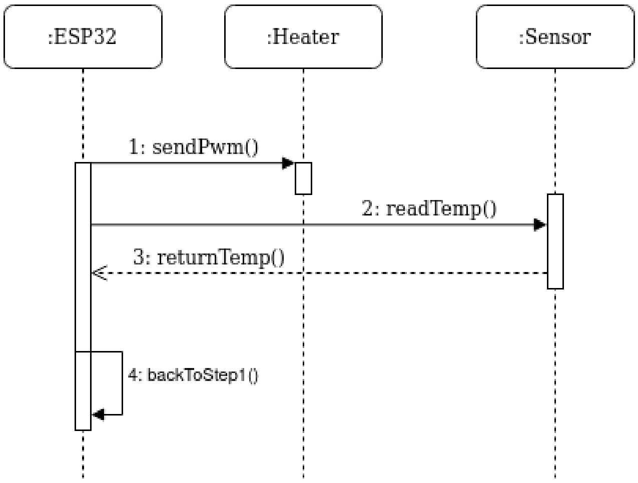

Figure 7. UML sequence diagram

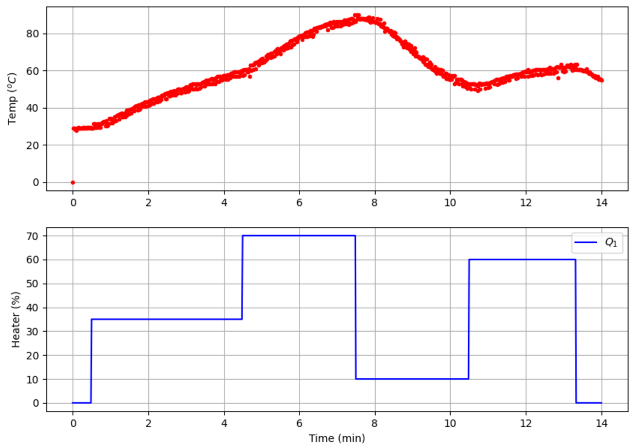

Figure 8. Open loop experiment with ESP32 for FOPDT model identification

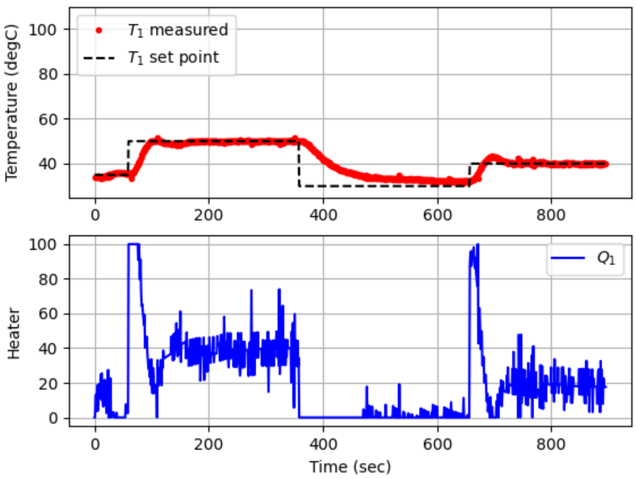

Figure 9. PID experiment on ESP32: hardware and software architecture validation

since there is no activation functions, biases or hidden layers. The Sensory Input may has multiple stimuli, since besides control there are also machine learning tasks such as optimization (Jafari et al., 2020), classification (Mei et al., 2017) and prediction (Parsapoor and Bilstrup, 2012; Parsapoor et al., 2014) where the emotional learning is successfully applied and is under active research (Lotfi and Khazaei, 2018). For control objectives SI and the reward signal (Rew) generated by the Emotional Signal (ES) block in Figure 11 can be scalar values such as the tracking/regulation error e or any function $f(e,u)$ of the tracking/regulation error and the control signal u, as presented in Figure 11. Here the SI is the output of a PID controller within the BELBIC structure as in Darestani et al. (2011) and the Rew is the absolute error $|e|$ to make it simple.

From Figure 10 the following equations were derived by Lucas et al. (2004) for the AM and OC blocks:

$$A_i = S_i \times V_i, \quad i = 1 \dots n, \tag{2}$$

$$A_{TH} = max(S_i), \tag{3}$$

$$O_i = S_i \times W_i, \tag{4}$$

where n is the length of SI array. Each A_i and O_i represents a particular node within its blocks. An additional TH signal A_{TH} acts on the AM when SI has more than one input, what is not the situation in the example when the SI is the output of a PID controller in the SISO case. The following weights V_i and W_i are updated according to the following differences Δ:

$$\Delta V_i = \alpha \times \left(S_i \times max \left(0, Rew - \sum_j A_j \right) \right), \tag{5}$$

$$\Delta O_i = \beta \times (S_i \times (E^\star - Rew)), \tag{6}$$

where $j = 1 \dots n+1$ since it includes A_{TH} and $\alpha, \beta > 0 \in R$ are parameters associated to learning rates, which must be tuned. E and $E^\star$ are the compensatory errors which do include and does not include A_{TH}, respectively and thus are written as

$$E^\star = \sum_i A_i - \sum_i O_i, \tag{7}$$

$$E = \sum_j A_j - \sum_i O_i, \tag{8}$$

and therefore E is indeed the BELBIC control signal u. From (5) one has the excitatory nature for the AM, since when the AM nodes do not exceed the Rew signal, there is no change in V and $\Delta V_i \geq 0$. The inhibitory task is handled by the OC, increasing or decreasing the learning as the process evolves.

Equations (2)−(8) have been implemented mostly in its matrix form on Digital Signal Processors (DSP) as in Golabian et al. (2009), and embedded applications on FPGA (Iranpour and Sharifian, 2017; Jamali et al., 2009) due to its parallel architecture for AM and OC computations. Since ESP32 boards have native RTOS support and are dual core, it is a suitable platform for BELBIC as well.

3. SIMULATIONS AND BELBIC EXPERIMENTS

To keep the compatibility with TCLab source codes, BELBIC was implemented as a Python script (Fonseca, 2021) and validated with the automatic voltage regulator model (9) in Valizadeh et al. (2008), as presented in Figure 12. The control signal saturation is 3.3 to reflect the actual

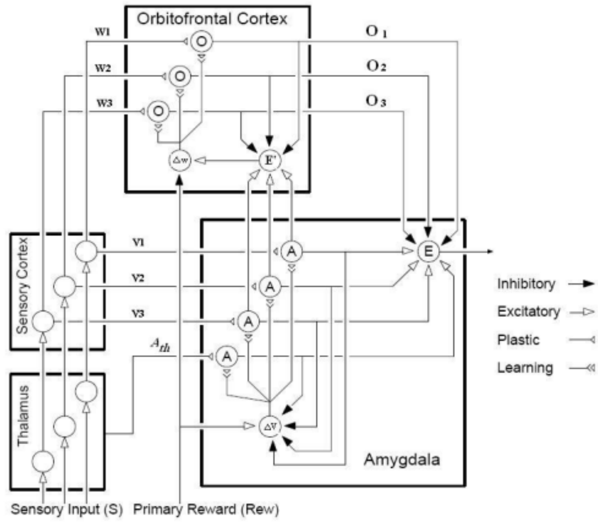

Figure 10. Graphical representation of Brain Emotional Learning

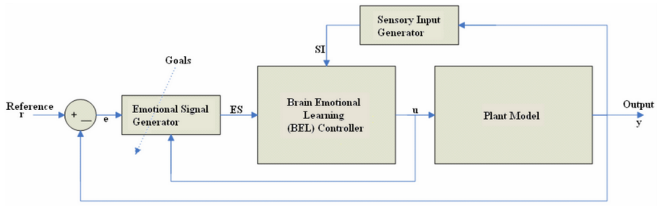

Figure 11. Closed-loop BELBIC block diagram

voltage limit for ESP32 pins. The simulation parameters were $\alpha = 0.45, \beta = 0.01, V_1 = 0.81, W_1 = 1.0, K_p = 3.98, K_i = 0.58, K_d = 0.63$.

$$G_2(s) = \frac{V_T}{V_E} = \frac{K_A K_E K_G}{(1+\tau_A s)(1+\tau_E s)(1+\tau_G s)}, \tag{9}$$

where $K_A = 10, \tau_A = 0.1, K_E = \tau_E = K_G = \tau_G = 1$ and the simulation step is $h = 0.01s$.

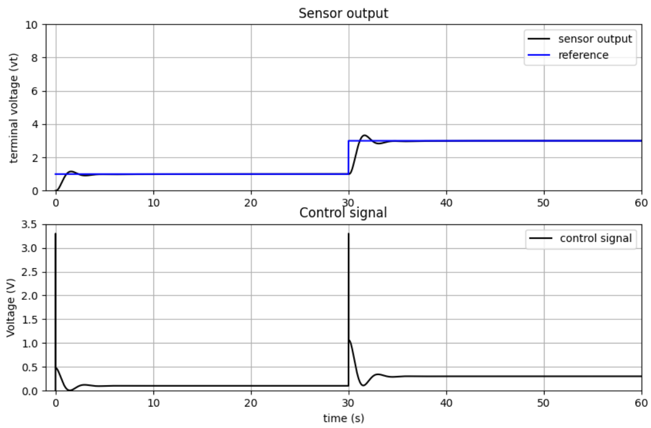

Figure 12. BELBIC simulation - Python script

Once simulations results were validated, the interface to ESP32 and the circuit in Figure 4 was established based on the system architecture in Figure 5. For this experiment the BELBIC parameters were: $\alpha = 0.001, \beta = 0.00001, V_1 = 0, W_1 = 0, K_p = 10, K_i = 0.2, K_d = 10$, with a time constant $T = 1s$. The bottom plot in Figure

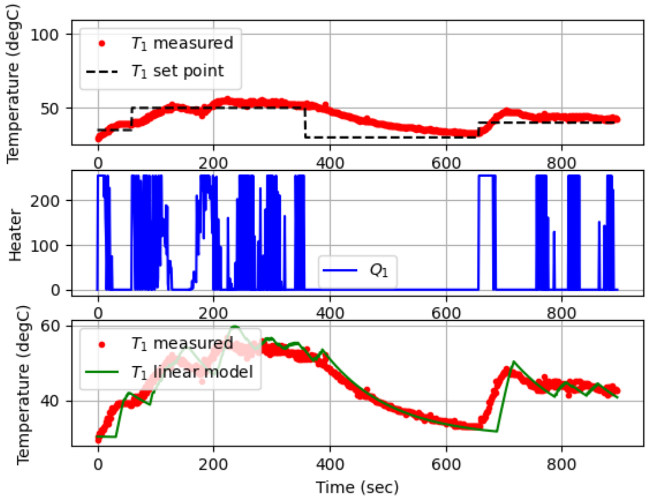

Figure 13. BELBIC running on ESP32 for the TCLab-based thermal system. Top plot: heater temperature tracking; Middle plot: PWM control signal; Bottom plot: detail on actual×model output

13 highlights the validation of the FOPDT model (1). In order to contribute as a teaching-learning approach, Tables 1 and 2 present some usual descriptive statistics based on the gathered data of the PID and BELBIC experiments. The idea here is not in the controllers comparison, already detailed for more complex systems in the references cited throughout this work, but how students can derive some simple metric to evaluate oscilations (process output and control signal standard deviation *std*) and average performance (mean and peak errors).

PID performance indicators			
	Temperature (degC)	**Heater (%)**	**error**
count	900.000000	900.000000	900.000000
mean	42.681112	15.084637	-3.019121
std	6.302056	22.948615	5.746636
min	30.791789	0.000000	-21.808407
25%	37.145650	0.000000	-4.335288
50%	41.544477	5.010056	-1.319648
75%	49.853372	22.975460	-0.078201
max	53.763441	100.000000	13.831867

Table 1. PID performance indicators

BELBIC performance indicators			
	Temperature (degC)	**Heater (%)**	**error**
count	900.000000	900.000000	900.000000
mean	43.034648	14.094379	-3.371571
std	5.924422	24.252915	6.313546
min	31.280547	0.000000	-21.808407
25%	37.634409	0.000000	-5.679374
50%	42.521994	0.000000	-2.297165
75%	48.875855	20.660079	-0.830890
max	54.740958	100.000000	12.854350

Table 2. BELBIC performance indicators

4. CONCLUSION

Based on the TCLab software and plant shield, this paper presented a proposal for adapting the source code to ESP32 boards, with advanced hardware specifications and more suitable for the IoT context and control objective, due to its native Wi-Fi, dual core and DAC features. The main existing PID and identification experiments were replicated on ESP32, as well as an intelligent control experiment based on the BELBIC method was proposed and evaluated. These findings do motivate further developments of a TCLab-based PCB version integrated to ESP32 and its continuous use by educational institutions to add active learning based on low cost control experiments.

REFERENCES

Beheshti, Z. and Hashim, S. (2010). A review of emotional learning and its utilization in control engineering. *Int. J. Advance. Soft Comput. Appl.*, 2(2), 191–208.

Coelho, J., Pinho, T., Boaventura-Cunha, J., and Oliveira, J. (2017). A new brain emotional learning simulink® toolbox for control systems design. *IFAC-PapersOnLine*, 50(1), 16009–16014. doi: https://doi.org/10.1016/j.ifacol.2017.08.1912. 20th IFAC World Congress.

Darestani, M., Zareh, M., Roshanian, J., and Smith, C. (2011). Verification of intelligent control of a launch vehicle with HILS. *Journal of Mechanical Science and Technology*, 25. doi:https://doi.org/10.1007/s12206-010-1227-1.

de Moura Oliveira, P., Hedengren, J.D., and Rossiter, J. (2020). Introducing digital controllers to undergraduate students using the tclab arduino kit. *IFAC-PapersOnLine*, 53(2), 17524–17529. doi: https://doi.org/10.1016/j.ifacol.2020.12.2662. 21st IFAC World Congress.

ESPRESSIF (2021). ESP32 technical reference manual. 1–718. URL `https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf`. Version 4.6.

Fonseca, G. (2021). tclab-esp32. `https://github.com/Gustavo053/tclab-esp32`.

Golabian, S., Lucas, C., Jamali, M., and Afrasiabi, M. (2009). Implementation of the emotional controller using DSK TMSC320 digital signal processor. In *CSIT 2009 Proceedings*, 477–480.

Hacioglu, A., Rao, S., and Hedengren, J. (2020). Teaching chemical engineering with matlab, simulink and tclab. 1–53. URL `https://www.mathworks.com/content/dam/mathworks/training/onramp/teaching-chemical-engineering-with-matlab-simulink-and-tclab.pdf`. Online; retrieved 25-March-2022.

Herrera, M., Camacho, O., Leiva, H., and Smith, C. (2020). An approach of dynamic sliding mode control for chemical processes. *Journal of Process Control*, 85, 112–120. doi:https://doi.org/10.1016/j.jprocont.2019.11.008.

Iranpour, E. and Sharifian, S. (2017). An fpga implemented brain emotional learning intelligent admission controller for saas cloud servers. *Transactions of the Institute of Measurement and Control*, 39(10), 1522–1536. doi:10.1177/0142331216644042.

Jafari, M., Xu, H., and Carrillo, L. (2020). A biologically-inspired reinforcement learning based intelligent distributed flocking control for multi-agent systems in presence of uncertain system and dynamic environment. *IFAC Journal of Systems and Control*, 13, 100096. doi: https://doi.org/10.1016/j.ifacsc.2020.100096.

Jamali, M., Arami, A., Dehyadegari, M., Lucas, C., and Navabi, Z. (2009). Emotion on fpga: Model driven approach. *Expert Systems with Applications*, 36(4), 7369–7378. doi:https://doi.org/10.1016/j.eswa.2008.09.067.

Junior, A.S., Gonçalves, L., de Paula, G., Tamanaka, G., Hernandes, A., and Aroca, R. (2020). BIPES: Block based integrated platform for embedded systems. *IEEE Access*, 8, 197955–197968. doi: 10.1109/ACCESS.2020.3035083.

Lotfi, E. and Khazaei, F. (2018). Competitive brain emotional learning. *Neural Processing Letters*, 47, 745–764. doi:https://doi.org/10.1007/s11063-017-9680-9.

Lucas, C. (2011). *BELBIC and Its Industrial Applications: Towards Embedded Neuroemotional Control Codesign*, volume 3, 203–214. doi:10.1007/978-3-642-17384-4$_1$7.

Lucas, C., Shahmirzadi, D., and Sheikholeslami, N. (2004). Introducing BELBIC: Brain emotional learning based intelligent controller. *Intelligent Automation and Soft Computing*, 10(1), 11–21. doi: 10.1080/10798587.2004.10642862.

Mei, Y., Tan, G., and Liu, Z. (2017). An improved brain-inspired emotional learning algorithm for fast classification. *Algorithms*, 10(2). doi:10.3390/a10020070.

Misal, S., Prajwal, S., Niveditha, H., Vinayaka, H., and Veena, S. (2020). Indoor positioning system (ips) using ESP32, MQTT and bluetooth. In *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*, 79–82. doi: 10.1109/ICCMC48092.2020.ICCMC-00015.

Oliveira, J. (2020). firmata-esp32. `https://github.com/josenalde/firmata-esp32`.

Oliveira, P., Hedengren, J., and Pires, E.S. (2020). Swarm-based design of proportional integral and derivative controllers using a compromise cost function: An arduino temperature laboratory case study. *Algorithms*, 13(12). doi:10.3390/a13120315.

Oliveira, P. and Hedengren, J. (2019). An apmonitor temperature lab pid control experiment for undergraduate students. In *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 790–797. doi:10.1109/ETFA.2019.8869247.

Park, J., Martin, R., Kelly, J., and Hedengren, J. (2020). Benchmark temperature microcontroller for process dynamics and control. *Computers Chemical Engineering*, 135, 106736. doi: https://doi.org/10.1016/j.compchemeng.2020.106736.

Parsapoor, M. and Bilstrup, U. (2012). Brain emotional learning based fuzzy inference system (belfis) for solar activity forecasting. In *2012 IEEE 24th International Conference on Tools with Artificial Intelligence*, volume 1, 532–539. doi:10.1109/ICTAI.2012.78.

Parsapoor, M., Bilstrup, U., and Svensson, B. (2014). A brain emotional learning-based prediction model for the prediction of geomagnetic storms. In *2014 Federated Conference on Computer Science and Information Systems*, 35–42. doi:10.15439/2014F231.

Rossiter, J., Pope, S., Jones, B., and Hedengren, J. (2019). Evaluation and demonstration of take home laboratory kit. *IFAC PapersOnLine*, 52(9), 56–61.

Tupac-Yupanqui, M., Vidal-Silva, C., Pavesi-Farriol, L., Ortiz, A., Cardenas-Cobo, J., and Pereira, F. (2022). Exploiting arduino features to develop programming competencies. *IEEE Access*, 10, 20602–20615. doi: 10.1109/ACCESS.2022.3150101.

Valizadeh, S., Jamali, M.R., and Lucas, C. (2008). A particle-swarm-based approach for optimum design of belbic controller in avr system. In *2008 International Conference on Control, Automation and Systems*, 2679–2684. doi: 10.1109/ICCAS.2008.4694214.