DOI: 10.22413/ijastems/2026/v12/i2/4

IOT-ENABLED ARDUINO-BASED SPEED CONTROL SYSTEM FOR SMART ELECTRIC BICYCLE APPLICATIONS

A. Divya, K. Venkata Chandu, D. Sai Vignesh, B. Rahul, K. Balanjaneyulu
Electrical and Electronics Engineering, CBIT, Proddatur, India, 516360

***Abstract*— The use of electric bicycles does not cause any damage to the natural world. Electric bicycles are affordable but do not have effective speed regulators. The project aims to design a mechanism that would regulate speed of the bike. An Arduino Uno microcontroller was used for designing of the project. The speed regulator contains a sensor which gives information about the wheel's rotation to the computer. Then the computer converts the data obtained into a value which could be read as kilometres per hour. Speed was displayed on an LCD display. Two buttons were used by the driver to accelerate or decelerate the bike's speed. These commands gave signals to the bike's engine. The efficiency of the proposed regulator was checked in laboratory conditions. The results were very impressive. There is a high accuracy of the speed readings and quick response to the button pressing. However, there is one more advantage of the regulator. All parts used for creation of the device were cheap; it is necessary to pay less than a thousand rupees.**

***Index Terms*— Arduino Uno, E-Bike Speed Control, IR Sensor, PWM Motor Drive, I²C LCD, Cytron MD10, Embedded System**

I. INTRODUCTION

There is an increase in awareness of the pollution caused by the city. On the other hand, there is a continual rise in the price of fuel. This has led to the adoption of electric bicycles among many individuals because they move on a day-to-day basis.How much it cost to run. The market for e-bikes is really big it was over 40 billion US dollars in 2023. It is expected to keep growing at a rate of around 10 percent each year for the rest of the decade according to people who study the industry. Even though e-bikes are becoming really popular the parts that control the speed on models are often very simple: a twist throttle connected to a basic controller with no digital feedback and no display. The person riding the e-bike does not have any information about how they are going, and the controller cannot adjust to changes in the load.

Small computers, like microcontrollers have changed a lot of things about the electronics people use every day and using them to control e-bikes is a next step. The Arduino platform is particularly good for this it is easy to learn has a lot of open-source libraries. Can process signals in real-time. Other projects have already shown that an Arduino can read sensors generate outputs and communicate with devices, all the things you need for a speed controller.

This project is about making a prototype that combines a few things: a way to measure speed using an infrared beam, a display that shows feedback buttons to increase or decrease the speed and a motor driver that can handle a lot of power all tested with measurements. The system is kept simple so that it is easy to understand the design choices and trade-offs.

II. LITERATURE SURVEY

Fogelberg [1] examined the concept of solar-powered bicycle sharing and highlighted the importance of matching motor speed to available energy in a lightweight platform, a problem that the present speed feedback loop addresses indirectly.

Barve [2][3][4] published a series of papers on solar-assisted hybrid bicycles, detailing the electrical architecture needed to blend pedal effort with motor torque. Although those papers focused on energy harvesting rather than speed regulation, they pointed out that analog throttle circuits introduce nonlinearity that digital PWM control can eliminate.

A closely related work proposed combining an Arduino with a Wi-Fi module so that the rider could adjust speed remotely through a smartphone application [5]. That system used the Blynk platform and operated in two modes: a local push-button mode and a remote web mode. The present design deliberately omits wireless connectivity to reduce latency and eliminate dependence on network availability, which is a realistic concern for riders in rural or semi-urban areas.

A differential springless suspension study for e-bikes [6] addressed stability on uneven ground by using bevel-gear differentials and per-wheel motors. That work is complementary to the present system: combining independent wheel speed control with the differential drive architecture would be a logical next step.

III. PROPOSED SYSTEM ARCHITECTURE

Sensing Layer: An IR transceiver system is used to detect the passage of wheel spokes. The number of pulses within a particular interval is translated into speed measured in km/h by the program.

Control input Layer: There are two mechanical tactile buttons on this layer, each for controlling an increase or

DOI: 10.22413/ijastems/2026/v12/i2/4

decrease in the speed. In that way, the operator can set the desired speed in increments of around 5%.

Processing Layer: The Arduino Uno microcontroller takes inputs from the IR pulse train and mechanical button and computes the present speed value. Then it changes the duty cycle and displays results on an LCD after every 500 ms.

Actuation Layer: The Cytron MD10-POT module receives PWM and direction inputs from the Arduino and generates appropriate current for the brushed DC motor.

TABLE I. COMPLETE BILL OF MATERIALS

S. No	Component	Specification	Quantity	Purpose
1	Arduino Uno (ATmega328 P)	5 V, 16 MHz	1	Core controller
2	IR Sensor Module	5 V, digital out	1	Wheel-speed sensing
3	I²C LCD 16x2	5 V, 0x27 address	1	Speed/status display
4	Cytron MD10-POT	7–30 V, 10 A	1	Motor driver
5	DC Brushed Motor (MY6812)	12 V, 120 W, 3350 RPM	1	Traction motor
6	Tactile Push Buttons	5 V, momentary	2	Speed increment/ decrement
7	SMPS Power Supply	12 V / 5 A	1	System power
8	Connecting Wires & PCB	Standard	Lot	Wiring & mounting

IV. HARDWARE DESIGN

A. Power Supply Unit

This circuit makes use of the 230V AC voltage available from the main power line source. This voltage is rectified to generate a DC voltage of 12V. Later, a voltage regulator circuit of 7805 is used for generating a 5-volt source which will be useful for powering the Arduino microcontroller, IR sensor and LCD display. Here, electrolyte capacitors of value 1000 μF and 10 μF are used for eliminating the ripple components from the output of the voltage regulator circuit. Motor driver circuit makes use of the 12V power line.

B. Arduino Uno Microcontroller

The ATmega328P microcontroller present on the Arduino Uno board has a frequency of 16 MHz and needs power supply voltage of 5 V. The number of digital input/output pins of the microcontroller is 14; out of these, six pins can be used to generate pulse width modulation signal by the hardware of the controller. In addition to this, there are six pins that allow the microcontroller to read the values with the help of 10 bits resolution.

C. Infrared Speed Sensor

The components of the IR module include the 850 LED emitter and the phototransistor receiver. These components have been mounted on a PCB board and have a 5mm distance between each other. The disc that has eight slots has been installed to the motor shaft. As one slot moves past the light beam of the sensors, the output signal changes from low to high. This will trigger an interrupt on pin D2 of the Arduino.The operation of the IR module involves cooperation with the Arduino. Arduino will use the pulses that it receives within the 200ms to determine the speed. Therefore, the IR module and Arduino help in determining the speed using the following formula.

The speed is calculated using the formula as follows.

Km/hr = Number of pulses * circumferences of the wheel * 3.6 / Number of slots * Time Window

The wheel that is used during testing should have a circumference of 0.3meters and has a disc with eight slots. Therefore, the speed of the wheel can be measured using the IR module and Arduino. Speed measured by the IR module and Arduino is accurate to 0.7 kilometres per hour per pulse per measurement window.

D. I²C Liquid Crystal Display

The regular HD44780 compatible 16x2 character LCD comes with an I²C back panel using a PCF8574 IC to connect to the Arduino using just two wires instead of the usual eight wires needed (SDA on pin A4, SCL on pin A5). The LiquidCrystal_I2C library takes care of the interfacing between the I²C and the LCD. The upper line displays the speed in kilometres per hour, while the lower line gives the value of the PWM duty cycle in percent.

E. Cytron MD10-POT Motor Driver

The MD10-POT receives an input signal between 0 and 30 volts in PWM format at the PWM pin and a direction control signal at the DIR pin. In its internal operation, it employs a complete H-bridge design that can provide up to 10 A of current constantly and 30 A in brief pulses. The PWM frequency is set at 10 kHz, a value above the threshold of human hearing and contributing to decreased motor heating relative to slower frequencies. Since only forward motion is needed for this project, the DIR pin is kept constantly high, and only the PWM duty cycle is controlled.

F. Brushed DC Motor

The speed of the MY6812 motor is 3350 RPM when fully loaded with a power rating of 120W and a voltage of 12V. This small motor together with its shaft diameter of 8mm is capable of coupling to the wheel through chain/belt transmission. At 50% PWM, the motor spins at 1650 RPM, but with a 3:1 ratio reduction, the speed becomes equal to that of a regular bike, which is 20km/hr.

V. SOFTWARE DESIGN

DOI: 10.22413/ijastems/2026/v12/i2/4

The code for the firmware is programmed in C++ utilizing the Arduino IDE (2.x version). The main loop will perform the following actions in each loop:

1. Extract the number of IR pulses counted in the ISR routine, then clear the IR counter value.
2. Determine the speed (km/h) using the formula mentioned in Section IV-C.
3. Get the state of the two push buttons with a debouncing time of 20 ms.
4. Update the duty cycle value in increments/decrements of 13 (5% of 255) and ensure the new value is within the range of 0 to 255.
5. Set Timer 1 with the new duty cycle using analog Write () at pin D9.
6. Every 500 ms, update the LCD screen with the current speed and duty cycle values.

ISR code is kept as small as possible, simply incrementing a volatile integer counter before exiting. All calculations take place in the loop so that other interrupts can fire. The I2C functionality is implemented through Wire, and caution has been exercised to use lcd. Print () only in the 500 ms update segment in order to avoid introducing any time jitter due to the relatively slow I2C interface running at 100 kHz in standard mode.

There is a watchdog timer reset which ensures that if for whatever reason the loop gets stuck for longer than 2 seconds (for instance when the I2C hangs), the processor will be reset automatically. This is crucial in all unattended devices.

VI. RESULT AND DISCUSSION

The prototype was assembled on a general-purpose PCB and tested on a bench using a motor clamped in a vice and fitted with the slotted disc described in Section IV-C. A digital tachometer (Lutron DT-2234C) was used as the reference measurement, and results were recorded at five discrete PWM duty-cycle settings.

TABLE II. SPEED MEASUREMENT ACCURACY

PWM (%)	Reference Speed (km/h)	Measured Speed (km/h)	Error (km/h)	Error (%)
20	6.8	6.6	0.2	2.9
35	11.9	11.6	0.3	2.5
50	17.2	16.8	0.4	2.3
65	22.4	21.7	0.7	3.1
80	27.5	26.6	0.9	3.2

As seen from Table II, the maximal speed error achieved within all measurement points is 3.2% that seems sufficient for the application. This error grows with increasing speed due to the fact that less pulses are recorded by the 200 ms counter window at moderate speeds compared to the entire number of pulses. If the window size is increased to 500 ms, the error will be reduced twice for the high-speed range, but the instrument will seem slower to the rider.

TABLE III. PUSH-BUTTON RESPONSE LATENCY

Test No.	Button Press Time (ms)	PWM Update Time (ms)	Latency (ms)
1	0	58	58
2	0	63	63
3	0	72	72
4	0	67	67
5	0	79	79

The latency caused by the 20 ms debounce code forms the bulk of the response latency; while the rest of the latency is attributed to loop processing time. With a worst-case latency of 79 ms, which is imperceptible to a human rider, it is evident that the current polling-with-debounce strategy is sufficient. False triggering was not experienced in any of the five test runs.
Total energy consumption of the whole system was measured at 4.1 W using a 50% PWM while spinning the motor under no load conditions. Energy consumption by the Arduino board was measured to be 0.35 W.

VII. CONCLUSION

In this paper, a low-cost, Arduino-based speed control and monitor system for e-bikes has been developed. The use of an infrared optical sensor for measuring the speed of the wheel, the inclusion of an I²C liquid crystal display, push buttons for user interaction, and a motor drive from Cytron have resulted in an implementation that measures the wheel speed accurately up to a maximum error of 3.2%, and which reacts within 80 ms when the rider gives commands. The entire hardware bill of materials is not expensive, and the software used for programming the board will be sufficiently concise for a fourth-year student to understand.

Possible improvements for future development include implementing a closed-loop speed controller using a proportional-integral method to keep constant speed despite changes in gradient on the road. Implementing a battery voltage monitor as well as an indicator that shows the state of charge is very practical. Using a Bluetooth or WIFI module would make logging rides possible through an app. Upgrading from the brushed DC motor and the Cytron drive can be done using a brushless motor with its field-oriented control driver.

ACKNOWLEDGMENT

The authors thank the faculty and laboratory staff of the Department of Electrical & Electronics Engineering, KITS College of Engineering, for providing access to equipment

DOI: 10.22413/ijastems/2026/v12/i2/4

and offering guidance throughout this project. Special thanks are due to the project guide Dr. P. Lakshmi Narayana for his invaluable suggestions during the design and testing phases.

REFERENCE

[1] F. Fogelberg, "Design Analysis of a Solar-Powered Bike-Sharing Concept," Viktoria Swedish ICT, Gothenburg, Sweden, Tech. Rep., 2014.
[2] D. S. Barve, "Development of a Solar-Assisted Hybrid Bicycle," Int. J. Current Eng. Technol., pp. 377–380, 2016.
[3] D. S. Barve, "Implementation Aspects of a Solar Hybrid Bicycle System," Int. J. Current Eng. Technol., pp. 378–379, 2016.
[4] D. S. Barve, "Performance Considerations for Solar Hybrid Bicycles," Int. J. Current Eng. Technol., p. 380, 2016.
[5] N. R. Kolhe and S. V. Bhosale, "E-Bike Speed Controller System Using Arduino and Wi-Fi Module," Int. J. Innov. Res. Sci. Eng. Technol., vol. 10, no. 4, pp. 3452–3456, Apr. 2021.
[6] M. P. Patil and R. K. Deshmukh, "Springless Differential Suspension System for Enhanced E-Bike Stability," Int. J. Mech. Eng. Res., vol. 8, no. 2, pp. 101–106, 2022.
[7] Cytron Technologies SdnBhd, "MD10-POT 10 A DC Motor Driver User Manual," Rev. 1.2, 2020. [Online]. Available: https://www.cytron.io
[8] Arduino LLC, "Arduino Uno Rev3 Datasheet," 2023. [Online]. Available: https://www.arduino.cc.