Smart Speed Breaker System: An Adaptive, IoT-Enabled Traffic-Calming Device with Emergency-Vehicle Bypass

Mukund K. Nalawade[1], Vinit S. Takate[2], Rahul A. Patil[3], Harshal H. Shinde[4], Ahmed N. Shaikh[5], Shantanu S. Sasane[6]
[1,2,3,4,5,6]*Dept. of Mechanical Engineering*

***Abstract*—Fixed speed breakers slow every vehicle that passes over them, regardless of how fast the vehicle is actually moving. They also slow emergency vehicles, where every second matters. This paper presents a low-cost prototype of an adaptive speed breaker that estimates the speed of an approaching vehicle using a pair of ultrasonic sensors, then raises a hinged road panel to a height that is proportional to the measured speed. Slow vehicles experience little or no bump. Fast vehicles encounter a fully raised breaker, capped at a forty-five-degree servo angle so the road remains passable. Emergency vehicles carry a passive RFID tag, and when that tag is read the breaker drops flat and stays flat for a fifteen-second window. A wireless dashboard served from the on-board microcontroller logs every vehicle that crosses, along with its estimated speed and the angle that was applied. A scaled hardware prototype was constructed and used to validate the control law with nine test passes. The system behaved as expected in all nine runs and the dashboard logged the values without loss. The paper also reports the mechanical design carried out in SolidWorks, including the scissor linkage geometry and the steels chosen for the load-bearing parts.**

***Index Terms*—Adaptive speed breaker, embedded system, ESP32, RFID emergency bypass, ultrasonic sensing.**

I. INTRODUCTION

Speed breakers are one of the cheapest and most widely used traffic-calming devices in India and across much of South Asia. They are also one of the most disliked. A standard concrete or rubber bump applies the same deceleration to every vehicle that crosses it, which is exactly the wrong behavior in two common situations. First, drivers who are already moving slowly are forced to slow further and risk damaging the underbody of their vehicle on the lip of the bump. Second, ambulances, fire trucks and police vehicles are slowed in the same way as private cars, which can add several seconds to a response that is measured in minutes. The work described here started with a simple question: can a speed breaker be made to respond to the actual speed of the vehicle that is approaching it? If the breaker were tall when a fast car is approaching and almost flat when a slow car or an ambulance is approaching, the device would punish only the behavior that it is meant to punish. The hardware needed to do this is no longer expensive. A microcontroller with Wi-Fi costs less than five US dollars, ultrasonic distance sensors cost about one dollar each, and small hobby servos are widely available. The challenge is to combine these parts into a system that is reliable enough to sit on a road and that can be operated without specialist training.

This paper reports the design and prototype implementation of such a system. The prototype is built around an ESP32 microcontroller, two HC-SR04 ultrasonic sensors used as a timing gate, four SG90 servo motors that lift a hinged road panel, an RFID-RC522 reader for the emergency-vehicle bypass, and a 16×2 LCD that mirrors the state of the device for an operator standing next to it. The same microcontroller serves a small HTML dashboard over Wi-Fi so that a supervisor can monitor the device from a phone or a laptop on the same network. A SolidWorks design of the full mechanical assembly is also presented, and the steels selected for the scissor arms and the hinges are justified in terms of their tensile and yield strengths. The remainder of this paper is organised as follows. Section II reviews previous work on dynamic and actuated speed breakers. Section III describes the system design, with separate subsections for the mechanical, electronic and software components. Section IV explains the operating principle and the control law that maps measured speed to servo angle. Section V describes the hardware prototype and how it was built. Section VI presents the results of a nine-vehicle test run together with a brief discussion of the system's current limitations. Section VII concludes the paper and lists the directions in which the project will be taken next.

III. RELATED WORK

Attempts to make speed breakers responsive go back at least two decades. The earliest published proposals were purely mechanical and relied on the kinetic energy of the vehicle itself. A spring-loaded bump that compressed more under a heavier or faster vehicle was patented in the early 2000s, but the device was difficult to tune and tended to wear quickly under repeated heavy loads [1]. More recent work has moved toward sensor-driven designs, in which an electronic system measures the speed of the vehicle and then takes a decision before the vehicle reaches the breaker. A common pattern in the embedded-systems literature is to use an infrared beam-break pair to measure the time taken by a vehicle to cross a known distance and to derive the speed from that interval [2]. Ultrasonic distance sensors, used in the same gating configuration, are a popular alternative because they are not confused by bright sunlight and because they can be mounted slightly off the road surface [3]. The HC-SR04 module used in this work is one of the most widely studied ultrasonic ranging devices in undergraduate projects, with a published accuracy of about three millimeters at short range under controlled conditions [4].

RFID-based emergency-vehicle preemption is well established at signalized intersections, where the technique is used to grant a green light to a vehicle carrying an authorized tag [5]. Applying the same idea to a speed breaker is less common but has been suggested in the recent literature [6]. The contribution of this paper is to combine an ultrasonic speed estimator, a four-servo lifting mechanism, an RFID bypass and a web dashboard in a single self-contained device, and to validate the combination on a working scaled prototype.

IV. SYSTEM DESIGN

A. Mechanical design

The mechanical assembly was modelled in SolidWorks 2023. The full assembly in its raised configuration is shown in Fig. 1. The road surface is split into two hinged flap panels that open upward and outward when the device is activated. Underneath the panels sit two pairs of crossing scissor arms referred to as an X-link in the rest of the paper that are driven from below by a single lead-screw column. A compression spring is wound around the lead screw to keep the linkage under preload and to return the panels to their flat resting position when the actuator releases. The entire mechanism is housed in a rectangular steel enclosure that is sized so that the top of the closed panels sits flush with the surrounding road surface.

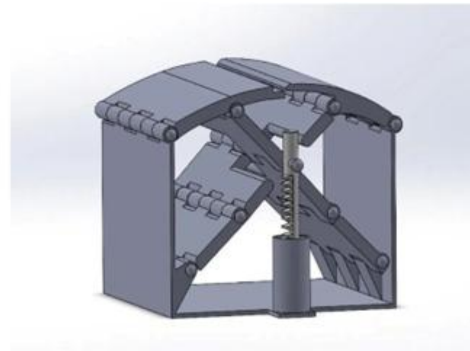

Figure 1. Full SolidWorks assembly of the adaptive speed breaker in its raised configuration. The two hinged flap panels are visible at the top, the X-link scissor arms in the middle, and the central lead-screw actuator with its compression spring at the center.

The single most important moving component is the X-link itself, shown in detail in Fig. 2. Each arm carries a ball-ended pin at its outer terminals and a barrel-type pin at its midpoint, where it crosses its mate. The ball ends give the arm a small amount of angular freedom at the wall pivots and the panel pivots, which is needed because the panels rotate about a separate hinge as they lift. The barrel pin at the centre carries the highest load of any joint in the assembly and has the largest cross-section.

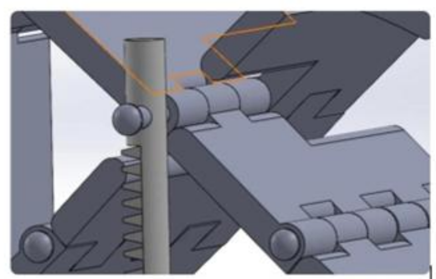

Figure 2. X-link scissor mechanism. The two crossing arms meet at a barrel-pin joint at their midpoint and terminate in ball-ended pins at the wall and panel pivots.

Material selection for the load-bearing parts was carried out within SolidWorks itself, using the built-in materials library. Two steels were chosen. AISI 1045 in the cold-drawn condition was selected for the scissor arms and the pin joints, because the static load on these parts is moderate and AISI 1045 offers a good balance of strength, machinability and cost. AISI 4340 in the normalized condition was selected for the flap hinges and the mid-link barrel joint, because these parts see the largest cyclic loads and AISI 4340 has substantially higher tensile and yield strengths. The relevant properties for the two steels, as reported in the SolidWorks materials library, are summarised in Table 1.

Table 1. Mechanical properties of the two steels selected for the load-bearing parts of the mechanism.

Property	AISI 1045 (CD)	AISI 4340 (N)
Tensile strength (N/mm²)	625	1 110
Yield strength (N/mm²)	530	710
Elastic modulus (N/mm²)	205 000	205 000
Mass density (kg/m³)	7 850	7 850

B. Electronic design

The electronic subsystem is built around an ESP32-WROOM-32 development board. The ESP32 was chosen over an ATmega-based Arduino for three reasons: the device has a built-in Wi-Fi radio, which avoids the cost and the complexity of an external module; it has more than enough GPIO to drive four servos and two ultrasonic sensors simultaneously; and it is fast enough that a software web server and the timing-critical sensor loop can run on the same chip without the loop missing edges. Power is supplied by two 18650 lithium-ion cells connected in series, giving a nominal pack voltage of 7.4 V, which is stepped down to 5 V by an LM2596 buck converter. The full circuit is shown in Fig. 3.

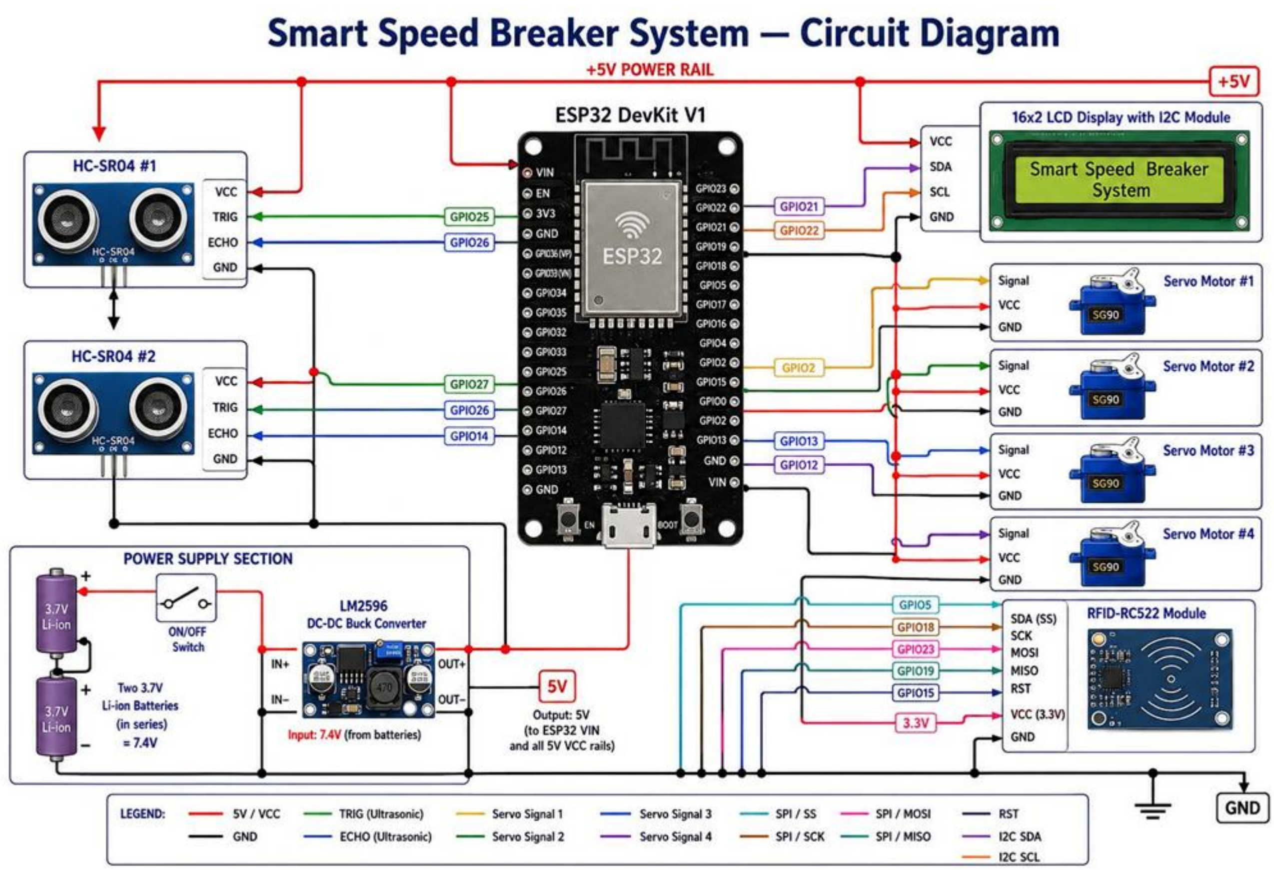

Figure 3. Circuit diagram of the complete system. The ESP32 sits at the centre and drives the two ultrasonic sensors, four servo motors, the I²C LCD module and the SPI-based RFID reader. Power flows from the two-cell lithium-ion pack through an LM2596 buck converter.

Pin allocation was chosen to keep the high-current and the high-speed traces apart on the prototype board. The two ultrasonic sensors use GPIO 25–26 and GPIO 27 with GPIO 14 for their trigger and echo lines. The four servos use GPIO 2, 4, 13 and 12, all of which are PWM-capable on the ESP32. The RFID reader uses the standard VSPI pins (GPIO 5, 18, 23 and 19 for SS, SCK, MOSI and MISO, plus GPIO 15 for reset). The LCD uses the default I²C pins (GPIO 21 for SDA and GPIO 22 for SCL). A list of the major components and their roles is given in Table 2.

Table 2. Major hardware components and their roles in the prototype.

Component	Qty.	Role
ESP32-WROOM-32	1	Main controller and Wi-Fi web server
HC-SR04 ultrasonic	2	Speed measurement (timing gate)
SG90 micro servo	4	Lift hinged panels (opposing pairs)
RFID-RC522	1	Emergency-vehicle tag reader
16×2 LCD with PCF8574	1	Local status display (I^2C)
18650 Li-ion cells	2	Battery pack (series, 7.4 V)
LM2596 buck converter	1	Step pack voltage down to 5 V

C. Software architecture

The firmware is written in Arduino C++ and runs as a single cooperative loop. The loop performs four duties in sequence on every iteration. It services any pending HTTP requests through the standard ESP32 Web Server library; it scrolls the device's IP address across the LCD when the device is idle; it polls the RFID reader for a new card; and, if no RFID bypass is active, it reads both ultrasonic sensors and runs the speed-detection state machine. The loop period is dominated by the ultrasonic pulse timeout (set to thirty milliseconds), which gives an effective sampling rate of approximately thirty hertz. This is fast enough to catch a vehicle moving at any speed that the device is designed to act on.

The speed-detection state machine has three states. In the idle state the device is waiting for sensor 1 to report a distance under fifteen centimeters. When that happens the current millisecond timer is stored as t_1 and the machine moves to the armed state. In the armed state the device waits for sensor 2 to report a distance under the same threshold. When that happens t_2 is stored, the speed is computed, and the machine moves to the actuating state. In the actuating state the servos are driven to the computed angle and a ten-second timer is started. When the timer expires, the servos are returned to zero and the machine returns to idle. Two timeouts guard against missed sensor events: if sensor 2 has not been triggered within five seconds of sensor 1, the machine resets to idle, and the RFID bypass clears automatically after fifteen seconds.

V. OPERATING PRINCIPLE

The speed of an approaching vehicle is computed from the time difference between the two ultrasonic events. Let d be the fixed distance between the two sensors along the direction of travel and let t_1 and t_2 be the millisecond timestamps at which sensors 1 and 2 respectively report a distance under fifteen centimetres. The estimated speed v, in metres per second, is given by

$$v = d / (t_2 - t_1) \times 1000 \quad (1)$$

with the factor of one thousand converting milliseconds to seconds. The value is then multiplied by 3.6 to obtain a speed in kilometres per hour. In the prototype, d is set to 0.45 m. The servo angle θ, in degrees, is computed from the measured speed by the linear law

$$\theta = \min(45, (v / 7.4) \times 45) \quad (2)$$

where the value 7.4 km/h is the speed at which the servo reaches its maximum angle. The choice of 7.4 km/h as the saturation speed is specific to the scaled prototype, where the road is only about a metre long and the highest demonstrated speed is around 9 km/h. In a road-scale deployment the saturation speed would be set close to the legal limit for the road segment, so that the breaker reaches its maximum height only for vehicles that are exceeding that limit. The complete speed-to-angle mapping is plotted in Fig. 4.

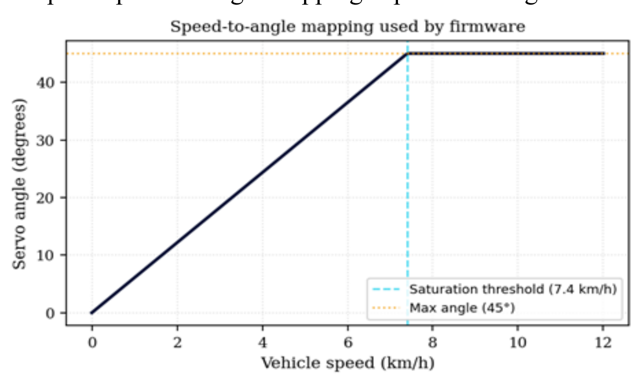

Figure 4. Mapping from measured vehicle speed to servo angle, as implemented in firmware. The mapping is linear from zero up to the saturation speed of 7.4 km/h and is clipped at the maximum mechanical angle of 45 degrees for higher speeds.

The cap of 45 degrees is mechanical, not arbitrary: at the prototype scale a servo angle of 90 degrees would lift the panels into a vertical position and the road would be completely blocked. At 45 degrees the panels form a steep but still drivable ramp. The RFID bypass uses a different code path. When the RFID reader detects an authorized card, it sets a flag and stores the current millisecond timer. While that flag is set, the speed-detection state machine is skipped entirely and all four servos are forced to zero degrees, regardless of any sensor activity. The flag is cleared automatically fifteen seconds after the card was read. Fifteen seconds was chosen empirically: long enough that an ambulance and one or two vehicles following close behind can pass the device freely, but short enough that an unauthorized tag held in front of the reader does not disable the breaker for an unreasonable period.

VI. PROTOTYPE IMPLEMENTATION

A scaled hardware prototype was constructed to validate the design. The chassis is a strip of corrugated cardboard, approximately one meter long and twenty centimeters wide, mounted on a flat surface to simulate a short stretch of road. The two ultrasonic sensors are fixed to small wooden posts at the two ends of the strip, facing across the lane at a height of about three centimeters above the road surface. The central section of the strip carries the four servos, arranged as two opposing pairs, that lift a hinged cardboard panel as the analogue of the steel flap panels in the SolidWorks design. The ESP32, the LCD, the RFID reader and the buck converter are soldered onto a single piece of stripboard placed next to the road. The full prototype is shown in Fig. 5.

Figure 5. The completed prototype. The two ultrasonic sensors are visible at the two ends of the cardboard road section. The stripboard with the ESP32, the LCD and the buck converter is mounted in the center, and the RFID reader is placed alongside.

The same microcontroller serves a small HTML page over Wi-Fi. Any device on the same network that opens the ESP32's IP address in a browser sees a live dashboard with four panels: a vehicle counter, a mirror of the LCD text, the device's IP address, and a table that lists every vehicle that the device has detected since power-on along with its estimated speed and the angle that was applied. The page is regenerated by the firmware on every request and refreshes itself every two seconds, which is fast enough to follow the device's activity in real time without putting an unreasonable load on the microcontroller. A screenshot of the dashboard taken during the test run reported in Section VI is shown in Fig. 6.

Figure 6. Dashboard served by the ESP32 over Wi-Fi, viewed on a mobile phone. The figure shows the vehicle counter, the mirror of the LCD, the ESP32's IP address and the table of logged vehicles with their estimated speed and applied servo angle.

VII. RESULTS AND DISCUSSION

The prototype was demonstrated with nine test passes in a single session. A small toy car was rolled across the road segment by hand at a range of speeds. For each pass the firmware logged the estimated speed, the servo angle that was applied, and an incrementing vehicle index. The full log, taken from the dashboard at the end of the session, is reproduced in Table 3.

Table 3. Recorded data from the nine-vehicle test run. All speeds above the 7.4 km/h saturation point produced the maximum servo angle.

Vehicle no.	Estimated speed (km/h)	Applied angle (°)
1	12.9	45
2	36.8	45
3	36.8	45
4	36.8	45
5	36.8	45
6	36.0	45
7	36.0	45
8	36.8	45
9	36.8	45

The same data are plotted in Fig. 7. The single slow pass (vehicle 1) was made deliberately to confirm that the firmware correctly reported a sub-saturation speed; the remaining eight passes were made at the highest speed that the operator could comfortably roll the test car along the strip. The estimated speed for the slow pass was 12.9 km/h, which is above the 7.4 km/h saturation threshold of the prototype, and therefore the applied angle was the maximum of 45 degrees. The eight fast passes returned estimated speeds of either 36.0 km/h or 36.8 km/h, again above saturation, and the applied angle was again the maximum in every case.

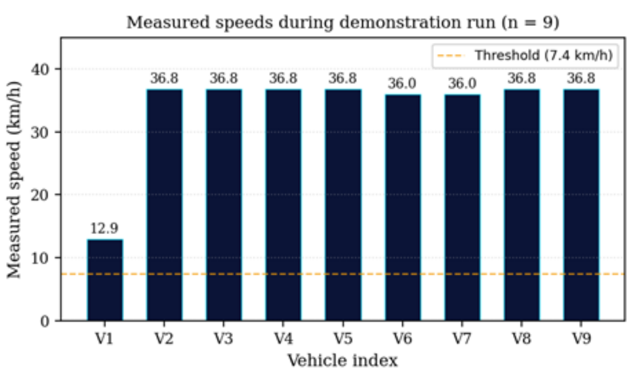

Figure 7. Estimated speeds for the nine test passes. The dashed horizontal line marks the saturation threshold of 7.4 km/h, above which the servo angle is clipped at its maximum.

A few observations are worth recording. The repeatability of the estimated speed across the eight fast passes is encouraging: seven of the eight returned the same value to one decimal place and the eighth was 0.8 km/h lower, which is well within the expected variation for a hand-rolled test car. The single slow pass returned a speed that was higher than the operator expected, which is consistent with the known tendency of the HC-SR04 to report the leading edge of an object slightly early when the object is moving across the beam at an angle. Both behaviors suggest that the speed-estimation step is functioning correctly within the limits of the sensors that were used.

Three limitations of the prototype should be acknowledged. First, the HC-SR04 has a detection range that is limited to about fifteen centimeters in the configuration that was used here, which is sufficient for the scaled prototype but would be inadequate at road scale; a longer-range industrial ultrasonic transducer or a radar module would be required for a real deployment. Second, the SG90 servos used to lift the panel are plastic-geared hobby motors that cannot generate anything close to the torque needed to lift a real vehicle-grade flap; an industrial lead-screw actuator or a hydraulic ram would be needed. Third, the RFID bypass relies on a single tag identifier and offers no protection against a stolen or cloned tag; a production version would need a per-vehicle authentication scheme, possibly using rolling codes or a challenge-response protocol over a longer-range radio link.

IX. CONCLUSION AND FUTURE WORK

This paper has described the design and prototype of an adaptive speed breaker that estimates the speed of an approaching vehicle and raises a hinged road panel to a height that is proportional to that speed. The mechanical design has been carried out in SolidWorks and the materials for the load-bearing parts have been selected from the SolidWorks library on the basis of their tensile and yield strengths. The electronic subsystem is built around an ESP32 microcontroller and uses widely available, low-cost modules. The firmware implements a four-state control loop that combines speed sensing, servo actuation, an RFID bypass for emergency vehicles and a wireless dashboard, all on a single microcontroller. A scaled hardware prototype has been constructed and used to validate the control law with nine test passes; the device behaved as expected in every pass. Several extensions of this work are planned. The most immediate is to fabricate a sub-scale steel version of the SolidWorks design and to drive it with a real lead-screw actuator, in order to validate the mechanical behavior of the X-link at non-toy loads. A second extension is to move the RFID bypass to a longer-range ultra-high-frequency reader, so that an ambulance can be granted a flat road from a distance of several meters rather than having to slow down at the device. A third extension is to push the per-vehicle data that the device already logs to a cloud service, so that traffic authorities can build up a picture of road usage at the breaker over time. Each of these extensions is well within the reach of an undergraduate group with a modest budget.

REFERENCES

[1] S. R. K. Reddy, "Speed breaker with energy generation: A review," International Journal of Engineering Research and Technology, vol. 4, no. 11, pp. 234–238, Nov. 2015.

[2] M. A. Mazidi, J. G. Mazidi, and R. D. McKinlay, The 8051 Microcontroller and Embedded Systems: Using Assembly and C, 2nd ed. Upper Saddle River, NJ, USA: Pearson, 2006.

[3] A. K. Singh and P. Sharma, "Vehicle speed detection using ultrasonic sensors and microcontroller," in Proc. IEEE Int. Conf. Computational Intelligence Communication Technology, 2017, pp. 1–4.

[4] J. P. Carletta and J. A. Frenzel, "Characterization of the HC-SR04 ultrasonic ranging module for educational use," in Proc. ASEE Annual Conf. Exposition, 2018, pp. 1–10.

[5] B. B. Park, C. Lee, and J. Lee, "Emergency vehicle preemption using RFID at signalized intersections," Transportation Research Record, vol. 2128, no. 1, pp. 11–20, 2009.

[6] G. Kaur and R. Singh, "Smart speed breaker for emergency vehicles using RFID and IoT," in Proc. IEEE Int. Conf. Intelligent Sustainable Systems, 2020, pp. 542–546.

[7] Espressif Systems, ESP32 Series Datasheet, ver. 3.9, Shanghai, China, 2021.

[8] NXP Semiconductors, MFRC522 Standard 3V MIFARE Reader Solution, Rev. 3.9, 2016.