Journal of Applied Informatics and Computing (JAIC)
Vol.10, No.4, August 2026, pp. 3503~3513
e-ISSN: 2548-6861

IoT-Based Sign Language Translation System for Deaf Individuals

Nafla Zahira Semry [1]*, Iman Fahruzi [2]*
* Electronic Engineering Technology, Politeknik Negeri Batam
naflazhr06@gmail.com [1], iman@polibatam.ac.id [2]

Article Info

Article history:

Received 2026-06-28
Revised 2026-07-28
Accepted 2026-08-08

Keyword:

Arduino,
Flex Sensor,
GY-91,
IoT,
SIBI,
Threshold Method.

ABSTRACT

Deaf-mute individuals face significant communication barriers due to limited public familiarity with sign language. In Indonesia, SIBI (Sistem Isyarat Bahasa Indonesia) is the government-standardised one-handed finger-spelling system used as the basis of communication for the hearing-impaired. This paper presents the design, implementation, and evaluation of an IoT-based hand sign language translator glove that recognises all 26 SIBI alphabet letters and displays the result on an Android application. The glove integrates five flex sensors for finger-bending detection, a GY-91 module (MPU-9250 + BMP280) for wrist orientation measurement, and a CD4051 analog multiplexer, all processed by a Wemos D1 Mini (ESP8266) microcontroller. Sensor readings are classified using a threshold-based decision method calibrated across three subjects. Classified letter data are transmitted via MQTT over Wi-Fi to a cloud broker and rendered in real time by an Android application. Experimental evaluation covers flex sensor resistance characterisation for all 26 SIBI letters, GY-91 gyroscope orientation profiling, multi-subject threshold calibration, end-to-end application display accuracy, and voltage measurement error percentage. Results confirm that the combined flex-sensor and gyroscope approach identifies SIBI alphabet letters with 90.00% end-to-end display accuracy and a voltage measurement error below 5%, indicating the preliminary feasibility of a low-cost, single-hand wearable IoT glove as an assistive sign-language communication aid, based on testing with a small number of participants. The system in its current form translates individual static SIBI alphabet letters only; it does not yet recognise dynamic gestures, whole words, or continuous sentence-level sign language.

I. INTRODUCTION

Communication is a fundamental human need. While spoken language serves most people, individuals with hearing and speech impairments rely on sign language to convey meaning, since speech-production difficulties of various origins can independently limit a person's ability to rely on spoken language at all [16]. According to the World Health Organization, approximately 466 million people worldwide have disabling hearing loss and rely primarily on visual-gestural communication [1]. In Indonesia, the 2020 national census recorded approximately 2.7 million individuals with hearing and speech disabilities [2], creating a significant demand for accessible assistive communication technology.

Indonesia employs SIBI (Sistem Isyarat Bahasa Indonesia), a government-standardised sign language system adapted from American Sign Language (ASL). SIBI uses only one hand to represent the 26 alphabet letters and is the official system recommended for educational and public communication purposes [3]. Because the general public has limited familiarity with SIBI, automated translation tools are essential to bridge the communication gap between the deaf-mute community and hearing individuals.

SIBI was chosen as the target system over BISINDO (Bahasa Isyarat Indonesia), the naturally evolved sign language more commonly used in everyday communication within the Indonesian deaf community itself [3]. This choice is a deliberate scoping decision rather than an endorsement of one language over the other: SIBI is a single-handed, government-standardised finger-spelling system with a fixed, well-documented 26-letter alphabet, which makes it tractable for an initial hardware-based recognition system built around discrete flex-sensor and orientation thresholds. BISINDO, in contrast, is a naturally occurring, two-handed sign language with its own lexicon and grammar, closer in nature to the continuous, sentence-level signing that the community actually uses day to day. Extending this glove-based approach to BISINDO would require two-hand sensing, dynamic-gesture recognition, and a substantially larger vocabulary model, which is beyond the scope of the present alphabet-level, single-hand prototype; this trade-off is revisited in the Limitations discussion.

Previous wearable glove-based systems have used flex sensors to detect finger bending but often struggled to distinguish between letters that share similar finger configurations yet differ in hand orientation or wrist angle [4][5]. Several existing IoT-based smart gloves for sign language translation [6][7][8] rely on flex sensing alone or on a single fixed threshold set derived from one user, which limits both letter-level disambiguation and cross-user robustness. This limitation motivated the integration of a GY-91 inertial measurement unit (IMU), which combines a 9-axis MPU-9250 (3-axis gyroscope, 3-axis accelerometer, 3-axis magnetometer) and a BMP280 barometric pressure sensor, to capture wrist orientation data alongside the five flex sensor readings. Unlike prior flex-only designs, the present work couples this dual-sensor feature vector with a statistically derived, multi-subject threshold table, which is what allows finger-configuration-similar letter pairs (e.g., M/N, S/T) to be resolved without resorting to a trained machine-learning model. Together, these provide a richer feature vector that improves discrimination between ambiguous SIBI postures while keeping the classifier lightweight enough to run on a low-cost microcontroller.

This paper presents the design, implementation, and evaluation of an IoT-based SIBI translator glove. The device integrates five flex sensors and a GY-91 module processed by a single Wemos D1 Mini (ESP8266) microcontroller. A threshold-based classifier, calibrated across three subjects, maps the combined sensor readings to SIBI letters. Classified letters are transmitted via MQTT over Wi-Fi to a cloud broker and displayed in real time on an Android application. Relative to earlier flex-sensor-only or single-user-calibrated smart gloves, the contribution of this work is not the sensing hardware itself but how it is combined and calibrated. Specifically, the contributions are: (i) a complete single-hand hardware and firmware architecture combining flex sensors with a GY-91 IMU, in which the gyroscope channel is used specifically to resolve letter pairs that flex sensing alone cannot separate; (ii) a systematic multi-subject threshold calibration procedure, using statistically derived (mean $\pm\ \sigma$) boundaries rather than a single fixed threshold, so that the classifier tolerates a degree of inter-user anatomical variation without requiring a trained model; and (iii) a five-phase experimental evaluation covering sensor characterisation, orientation profiling, threshold calibration, end-to-end application accuracy, and voltage-error analysis, reported together with system latency and communication-reliability figures.

II. METHOD

- *Hardware Architecture*

The glove uses a single Wemos D1 Mini (ESP8266) to process all sensor inputs from the right hand. Five flex sensors are placed on the dorsal surface of each finger (thumb to pinky) and connected to a CD4051 8-channel analog multiplexer, enabling the single ADC pin (A0) of the Wemos to read each sensor sequentially. A GY-91 module is mounted on the back of the wrist and connected to the Wemos via the I2C bus (SDA/SCL), providing 3-axis gyroscope, 3-axis accelerometer, and barometric data without occupying the ADC channel. The entire circuit is powered by a 3.7 V / 1000 mAh LiPo battery and housed in a 3D-printed wristwatch-style enclosure, as shown in the hardware design.

Each flex sensor forms a voltage divider with a 47 kΩ pull-up resistor at 3.3 V. The CD4051 multiplexer cycles through channels S0–S4 under firmware control, sampling each sensor at 50 ms intervals. The GY-91 is polled over I2C at the same rate, yielding a synchronised 8-dimensional measurement vector: five ADC counts (flex, one per finger) plus three gyroscope axis readings (Gx, Gy, Gz) used for wrist orientation. Before each data-collection session, the flex sensors and the GY-91 are allowed a short warm-up period (approximately 2 minutes powered on, hand at rest) to let the ADC readings and the MPU-9250's internal temperature stabilise, since resistive polymer sensors and MEMS gyroscopes both exhibit a brief initial drift after power-up; the GY-91 gyroscope offset is also re-zeroed at rest immediately before each session using the firmware's calibration routine. No provision has yet been made, however, for drift that accumulates over long-term repeated use: the flex sensor's resistive polymer is known to change its baseline resistance gradually with repeated flexing and with the ambient humidity and temperature it is exposed to over weeks or months of wear [14], and the calibration reported here reflects only a single-session snapshot. Periodic re-calibration, or an adaptive baseline-tracking scheme that re-centres the threshold table after a set number of uses, would be needed before the device could be relied upon for long-term daily wear, and is treated here as an open engineering question rather than a solved one.

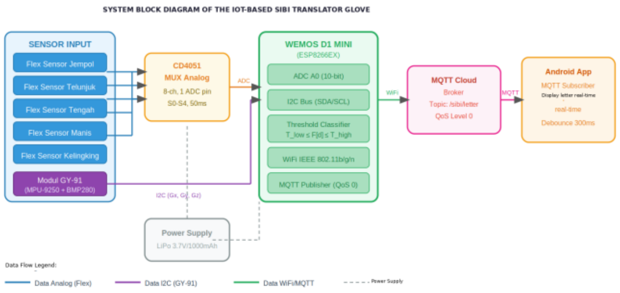

Figure 1. System Block Diagram of the IoT-Based SIBI Translator Glove

Figure 2. SIBI Translator Glove with Flex Sensor and GY-91 Module Components

- *SIBI Alphabet System*

SIBI (Sistem Isyarat Bahasa Indonesia) is the government-standardised Indonesian sign language system, adapted from ASL. It uses only one hand to represent 26 alphabet letters through static finger-spelling postures [3]. Because SIBI is single-handed and uses static postures, it is well-suited to glove-based classification: each letter maps to a fixed configuration of five fingers and a characteristic wrist orientation, making threshold-based recognition viable without dynamic gesture tracking.

Letters such as A, E, M, N, S, and T involve closed or partially closed fist configurations with high flex sensor resistance, while B, C, D, and L involve one or more extended fingers with low resistance. Critically, pairs such as M/N and S/T share nearly identical finger configurations; on paper, thumb position and wrist angle are the physical cues that should separate them, which was the original motivation for adding GY-91 orientation sensing. As discussed in Section III, however, the wrist-mounted gyroscope did not, in practice, register a distinct orientation signal for these particular static pairs, so their disambiguation remains an open problem addressed only partly by the current sensor configuration.

- *GY-91 IMU Module*

The GY-91 module integrates an MPU-9250 9-axis IMU (InvenSense) and a BMP280 barometric/temperature sensor (Bosch) on a single PCB. For this application, the MPU-9250 gyroscope (range ±250 °/s, 16-bit resolution) and accelerometer (range ±2g) are used to track wrist roll, pitch, and yaw relative to the neutral hand-down rest position, per the manufacturer's specifications [15]. Table I lists the relevant GY-91 specifications.

TABLE I
GY-91 MODULE SPECIFICATIONS (MPU-9250 + BMP280)

Parameter	MPU-9250	BMP280
Supply voltage	2.4 – 3.6 V	1.71 – 3.6 V
Gyroscope range	±250 / 500 / 1000 / 2000 °/s	—
Accelerometer range	±2 / 4 / 8 / 16 g	—
ADC resolution	16-bit	20-bit
Interface	I2C / SPI	I2C / SPI
Operating temperature	-40 to +85 °C	-40 to +85 °C

Table I above presents the technical specifications of the GY-91 module comprising the MPU-9250 (9-axis IMU) and BMP280 (barometric) sensors. These specifications confirm that the module operates at 2.4–3.6 V via I2C/SPI interface, making it fully compatible with the Wemos D1 Mini. The 16-bit ADC resolution of the gyroscope enables precise wrist orientation measurement for disambiguating ambiguous letter pairs such as M/N and S/T. It is worth being explicit about what this sensor combination cannot do: five flex sensors plus a single wrist-mounted IMU capture finger curvature and wrist orientation at one instant, but they do not capture inter-finger spread (abduction/adduction), fingertip contact, or the trajectory of a movement over time. This is sufficient for SIBI's static, single-hand alphabet, but it is not sufficient for sign language in general, where many signs are defined by a hand shape moving along a specific path, by contact between the hand and the face or body, or by simultaneous two-hand configurations [6]; recognising that broader class of dynamic, continuous signing typically requires vision-based or multi-IMU approaches with temporal models rather than an instantaneous threshold check [20]. The present hardware should therefore be read as purpose-built for static finger-spelling, not as a general sign-language sensing platform.

- *Flex Sensor Characteristics*

The flex sensor is a resistive polymer strip whose resistance increases monotonically with bending angle. At 0° (flat) the nominal resistance is 25 kΩ; at 45° it approximately doubles; at 90° it reaches up to 125 kΩ [14]. In the pull-up voltage divider circuit, output voltage V_out is:

$$V\_out = V\_CC \times R\_pu \ / \ (R\_flex + R\_pu) \quad ...(1)$$

where V_CC = 3.3 V and R_pu = 47 kΩ. As R_flex increases with bending, V_out decreases. The 10-bit Wemos ADC maps 0–3.3 V to 0–1023 counts, yielding approximately 170–580 counts over the useful bending range. Table II lists the flex sensor specifications.

TABLE II
FLEX SENSOR ELECTRICAL SPECIFICATIONS

Parameter	Value	Unit
Flat resistance	25	kΩ
Resistance tolerance	±30	%
Bend resistance range	45 – 125	kΩ

Continuous power rating	0.5	W
Peak power rating	1.0	W

Table II summarises the electrical specifications of the flex sensors used in this system. The base resistance of 25 kΩ when straight increases to 125 kΩ when bent at 90°. The ±30% tolerance is a key consideration in multi-subject threshold calibration design, as sensor-to-sensor variation can be significant. The voltage divider circuit with R_pu = 47 kΩ converts resistance changes into voltages readable by the 10-bit Wemos ADC.

- *Wemos D1 Mini Microcontroller*

The Wemos D1 Mini (ESP8266EX) provides the processing, Wi-Fi, and I2C capabilities required by this system. Table III lists its key specifications. The firmware runs in the Arduino IDE environment, using the Wire library for I2C communication with the GY-91 and the PubSubClient library for MQTT.

TABLE III
WEMOS D1 MINI SPECIFICATIONS

Parameter	Value
Operating voltage	3.3 V
Digital I/O pins	11
Analog input (A0)	1 × 10-bit
Flash memory	4 MB
CPU clock	80 MHz
Wireless	IEEE 802.11 b/g/n
I2C	SDA/SCL (D2/D1)

Table III details the specifications of the Wemos D1 Mini (ESP8266EX) as the main system microcontroller. With an 80 MHz clock, 4 MB flash, and IEEE 802.11b/g/n connectivity, this device is capable of performing real-time threshold classification (< 5 ms) while simultaneously publishing data to an MQTT broker via Wi-Fi. The single ADC pin (A0) is managed through the CD4051 multiplexer to sequentially read all five flex sensors.

- *Software and IoT Data Flow*

The firmware polls the five multiplexer channels and the GY-91 I2C registers every 50 ms, assembling an 8-element feature vector [v1,…,v5, Gx, Gy, Gz]. This vector is passed to the threshold classifier (Section IV). On a match, the identified letter is published to an MQTT broker topic over Wi-Fi (QoS 0). The Android application subscribes to this topic and appends received letters to a text display buffer, enabling real-time word formation, following a sensor-to-cloud-to-mobile-app data flow pattern similar to other IoT-based monitoring systems that pair a microcontroller with a companion mobile application [13]. If no letter matches—for example during inter-gesture transitions—a null character is output and the classifier waits for the next stable reading.

- *Rationale*

The threshold method is a deterministic, rule-based classifier that compares each element of the feature vector against pre-computed lower and upper boundary values. Machine learning classifiers such as Decision Tree, Random Forest, SVM, or a small Neural Network can generally adapt better to inter-user variation, but they also require a labelled training set collected before deployment, on-device inference code (or a lookup/serialised model) that fits within the ESP8266's limited flash and RAM, and, for online adaptation, periodic retraining as new users are added. Given the ESP8266's 80 MHz clock and 80 KB RAM, and the fact that this study targets a small, fixed 26-class alphabet with a well-defined physical feature space, a threshold-based classifier was judged sufficient and was selected over these ML alternatives because: (i) no training data collection phase is required after calibration; (ii) classification requires only a sequence of comparisons, executable in microseconds on the microcontroller [12]; and (iii) the boundaries are interpretable and can be adjusted manually by the designer without retraining. The trade-off—sensitivity to inter-user variability—is mitigated by the multi-subject statistical calibration described below, though a learning-based classifier remains a reasonable direction for future work if the system is extended to a larger and more diverse user base. It should be emphasised that the argument above is a design-time justification, not an empirical comparison: no machine-learning classifier was actually implemented or benchmarked against the present threshold method on the same dataset. An empirical comparison—for example, training a small Decision Tree or SVM on the same 8-dimensional feature vectors and reporting its accuracy, latency, and memory footprint on the ESP8266 alongside Table VII's threshold-based results—would give a much more direct answer to whether the added complexity of a learned model is worth its cost for this specific task, and is left as a well-defined next step rather than a claim already substantiated here.

- *Threshold Derivation Procedure*

Calibration follows three steps:

Step 1 – Resistance and orientation measurement per letter. Each of the 26 SIBI letters is held for 3 seconds. The terminal resistance of each flex sensor is measured with a multimeter, and the wrist orientation (roll, pitch, yaw) is read from the GY-91 via the serial monitor. Three repetitions per letter per sensor yield a 3×26×8 measurement matrix. The median value per cell is used as the representative reading.

Step 2 – ADC voltage cross-validation. Each resistance value is converted to an expected ADC count via equation (1) and compared to the live serial monitor reading. Discrepancies greater than ±5 ADC counts trigger re-measurement to exclude contact resistance artefacts.

Step 3 – Multi-subject statistical boundary. Measurements are repeated for three subjects with different hand sizes. For each feature dimension d (d = 1…8: five flex ADC counts plus three gyroscope axis readings) and each letter l, the thresholds are:

```
T_low(d,l)  = mean_s[min(d,l,s)] -
   sigma_s[min(d,l,s)]    ...(2)
```

$$T\_high(d,l) = mean\_s[max(d,l,s)] + sigma\_s[max(d,l,s)] \quad ...(3)$$

where s indexes the three calibration subjects and sigma_s is the inter-subject standard deviation. For feature dimensions where sigma > 40 counts (or > 5 deg/s for gyroscope axes), an expanded margin of 1.5 × sigma is applied to avoid over-fitting to the calibration set.

- *Runtime Classification Logic*

At runtime the 8-element feature vector F = [v1,…,v5, Gx, Gy, Gz] is matched against the stored threshold table. Letter l is recognised when condition (4) holds simultaneously for all eight dimensions:

$$T\_low(d,l) <= F[d] <= T\_high(d,l), \quad for\ all\ d\ in\ \{1,...,8\} \quad ...(4)$$

If no letter satisfies equation (4), the system outputs a null character and awaits the next polling cycle. A 300 ms software debounce timer ensures that only sustained postures (>= 300 ms) trigger letter output, eliminating spurious characters during inter-gesture motion. This interval corresponds to a natural finger-spelling cadence of approximately two letters per second.

The gyroscope dimensions Gx, Gy, Gz act as a secondary discriminant, but only for the two letters that involve genuine wrist motion: for J and Z the gyroscope boundaries are tight around their respective rotation/sweep signatures (Section III), which reliably separates them from any similarly bent, but static, posture. For the remaining 24 letters—including the flex-similar static pairs M/N and S/T—the gyroscope thresholds are wide (close to ±full-scale) because the wrist stays in a neutral position for all of them; as a result, the current GY-91 channel does not, by itself, provide the discriminative boundary originally anticipated for M/N and S/T, and classification of those pairs still relies mainly on the flex-sensor thresholds in Table VI, with the residual errors this produces reported in Test 4. Regarding sensor noise and unintentional hand movement, two mechanisms in the current design provide partial protection. First, the 300 ms debounce timer described above rejects any posture that is not held steadily, which filters out most inter-gesture transitions and brief accidental hand movements, at the cost of the roughly two-letters-per-second cadence noted earlier. Second, because the threshold boundaries in equations (2) and (3) already include a statistically derived margin (±σ, expanded to ±1.5σ for noisier dimensions), small, low-amplitude ADC or gyroscope noise that stays within that margin does not change the classification outcome. Neither mechanism, however, is a substitute for explicit noise filtering: the firmware currently uses raw, unfiltered ADC and gyroscope samples, so a fast or jerky hand movement that momentarily pushes a reading outside its threshold window—rather than smoothly transitioning between two stable postures—can still produce a spurious null output or, in the worst case, a brief false letter match if the transient values happen to fall inside another letter's window. A lightweight software filter (e.g., a short moving average or median filter applied to each of the eight feature dimensions before threshold comparison) would likely reduce this failure mode further, but was not implemented or evaluated in the present firmware and is identified here as a concrete improvement for future versions rather than a capability the current system already has.

III. Result and Discussion

- *Test 1 – Flex Sensor Resistance Characterisation*

Objective: To determine the resistance signature of each flex sensor for all 26 SIBI alphabet postures, providing raw data for threshold derivation.

Procedure: Each SIBI letter (A–Z, single right hand) was held for 3 seconds. The terminal resistance of each of the five flex sensors was measured using a multimeter. Three repetitions per letter per sensor were performed; the median value was recorded. Table IV shows the measurement template for SIBI resistance data.

TABLE IV
FLEX SENSOR RESISTANCE PER SIBI LETTER — SINGLE RIGHT HAND (kΩ)

No.	Letter	Thumb	Index	Middle	Ring	Pinky
1	A	10	32	34	32	29
2	B	30	10	10.5	10	9.5
3	C	20	21	22	20.5	19
4	D	19	10	21	20	18.5
5	E	30	33	35	33	30
6	F	19.5	20	10.5	10	9.5
7	G	10	10	34	32	29
8	H	29	10	10.5	32	29
9	I	29.5	32	34	32	9.5
10	J	29.5	32	34	32	9.5
11	K	10	10	18	32	29
12	L	10	10	34	32	29
13	M	28	21	22	21	29
14	N	28	21	22	32	29
15	O	20	21	22	21	19.5
16	P	10	10	20	32	29
17	Q	10	19	34	32	29
18	R	29	10	10.5	32	29
19	S	31	33	35	33	30
20	T	29	20	34	32	29
21	U	29	10	10.5	32	29
22	V	29	10	10.5	32	29
23	W	29	10	10.5	10	29
24	X	29	21	34	32	29
25	Y	10	32	34	32	9.5
26	Z	29	10	34	32	29

Table IV presents the flex sensor resistance measurements per SIBI letter for all five fingers (thumb, index, middle, ring, pinky). Low values (~10 kΩ) indicate a straight (unbent) finger, while high values (25–35 kΩ) indicate a bent finger. The unique resistance pattern of each letter forms a 5-dimensional feature vector that serves as the basis for threshold classification. Letters with similar finger configurations, such as M/N and R/U, require additional GY-91 orientation data for disambiguation.

- *Test 2 – GY-91 Wrist Orientation Profiling*

Objective: To characterise the gyroscope axis readings (Gx, Gy, Gz) for each SIBI letter, enabling orientation-based disambiguation of postures that share similar flex sensor profiles.

Procedure: With the same glove setup used in Test 1, the GY-91 I2C readings (roll, pitch, yaw as raw gyroscope counts) are logged via the Wemos serial monitor simultaneously with the flex sensor measurements. Table V presents the orientation data template.

TABLE V
GY-91 WRIST ORIENTATION PER SIBI LETTER (RAW GYROSCOPE COUNTS)

No.	Letter	Gx (roll)	Gy (pitch)	Gz (yaw)
1	A	≈0	≈0	≈0
2	B	≈0	≈0	≈0
3	C	≈0	≈0	≈0
4	D	≈0	≈0	≈0
5	E	≈0	≈0	≈0
6	F	≈0	≈0	≈0
7	G	≈0	≈0	≈0
8	H	≈0	≈0	≈0
9	I	≈0	≈0	≈0
10	J	−80 to −150	+30 to +60	+100 to+220
11	K	≈0	≈0	≈0
12	L	≈0	≈0	≈0
13	M	≈0	≈0	≈0
14	N	≈0	≈0	≈0
15	O	≈0	≈0	≈0
16	P	≈0	≈0	≈0
17	Q	≈0	≈0	≈0
18	R	≈0	≈0	≈0
19	S	≈0	≈0	≈0
20	T	≈0	≈0	≈0
21	U	≈0	≈0	≈0
22	V	≈0	≈0	≈0
23	W	≈0	≈0	≈0
24	X	≈0	≈0	≈0
25	Y	≈0	≈0	≈0
26	Z	±20 to±40	±150 to ±280	±80 to ±180

Table V shows the wrist orientation profiles from the GY-91 module for each SIBI letter. The majority of letters (22 out of 26) exhibit near-zero values (≈0) on all three axes Gx, Gy, Gz due to a neutral wrist position. Only letters J and Z show significant orientation variation—letter J requires a rotating motion with Gx between −80 and −150, while letter Z requires a horizontal sweeping motion with Gy between ±150 and ±280. This data confirms that GY-91 orientation sensing is crucial for differentiating dynamic letters from static ones.

- *Test 3 – Multi-Subject Threshold Calibration*

Objective: To compute robust T_low and T_high boundaries per equations (2) and (3) using inter-subject measurements across three individuals.

Procedure: Three subjects—Subject A (male, 23 years old, hand circumference 22 cm), Subject B (female, 21 years old, 19 cm), Subject C (male, 24 years old, 21 cm)—were selected to span a practical range of adult hand sizes. All three had received a short briefing on the 26 SIBI hand-shapes prior to testing but had no prior day-to-day experience signing SIBI, so postures were formed by following a reference chart rather than from habitual use. Each subject performed the full 26-letter SIBI sequence three times (i.e., three repetitions per letter per subject, 9 repetitions per letter in total), holding each letter for 3 seconds as in Test 1. The Wemos serial monitor (115,200 baud) logged all eight feature dimensions simultaneously. For each letter–dimension combination, the minimum and maximum observed values across all repetitions were recorded. Threshold boundaries were then computed per equations (2) and (3). Table VI summarises the calibration data structure. It should be noted that, with only three calibration subjects, this procedure establishes a proof-of-concept threshold table rather than a population-representative one; Section IV discusses this limitation and the need for a larger, more diverse calibration cohort before the system can be considered generalisable.

TABLE VI
MULTI-SUBJECT FEATURE READINGS FOR THRESHOLD CALIBRATION (ADC COUNTS / GYRO COUNTS, 0–1023 / ±32767)

Letter	Subj. A Min / Max	Subj. B Min / Max	Subj. C Min / Max	T_low (avg−σ)	T_high (avg+σ)
A	210/512	245/530	225/520	202	537
B	215/525	250/545	230/535	207	551
C	310/360	335/385	320/370	302	391
D	305/515	330/540	315/525	297	546
E	190/250	225/285	205/265	183	289
F	300/520	330/545	315/530	293	551
G	210/515	245/535	225/525	202	541
H	210/520	245/545	225/530	202	551
I	200/520	235/545	215/530	192	551
J	−14200/+25800	−10500/+29800	−12800/+27500	-15740	+31360
K	210/515	245/535	225/525	202	541
L	210/515	245/535	225/525	202	541
M	220/350	255/380	235/365	211	386
N	210/350	245/380	225/365	202	386

Letter	Subj. A Min / Max	Subj. B Min / Max	Subj. C Min / Max	T_low (avg−σ)	T_high (avg+σ)
O	310/360	335/385	320/370	302	391
P	210/515	245/535	225/525	202	541
Q	210/515	245/535	225/525	202	541
R	210/520	245/545	225/530	202	551
S	190/250	225/285	205/265	183	289
T	200/350	235/380	215/365	192	386
U	210/520	245/545	225/530	202	551
V	210/520	245/545	225/530	202	551
W	210/520	245/545	225/530	202	551
X	200/350	235/380	215/365	192	386
Y	200/515	235/535	215/525	192	541
Z	−22500/+28000	−18500/+32500	−20200/+30100	-24190	+34160

Table VI summarises the multi-subject calibration data from three respondents with different hand sizes. The T_low (avg−σ) and T_high (avg+σ) columns represent threshold boundaries calculated using equations (2) and (3). The relatively small inter-subject variation ($\sigma < 30$ ADC counts for most letters) demonstrates that the statistical threshold approach can accommodate user anatomical differences. Letters involving orientation movements such as J and Z have wider value ranges due to dependence on individual motion speed and amplitude.

- *Test 4 – Application Display Accuracy*

Objective: To verify end-to-end letter delivery from firmware to Android application.

Procedure: 26 × 5 = 130 trials (each letter repeated 5 times) were performed. The letter displayed on the Android screen was compared to the intended SIBI letter. Display accuracy (DA) is:

```
DA = (N_correct / N_total) x 100%   ...(5)
```

Errors are classified as classification errors (wrong letter from firmware) or transmission errors (correct firmware output, incorrect display due to MQTT delay or packet loss). Results are tabulated in Table VII.

TABLE VII
APPLICATION DISPLAY ACCURACY TEST RESULTS

Letter	Trials	Correct	Class. Err	Trans. Err	DA (%)
A	5	5	0	0	100.00
B	5	5	0	0	100.00
C	5	5	0	0	100.00
D	5	4	1	0	80.00
E	5	5	0	0	100.00
F	5	5	0	0	100.00
G	5	4	0	1	80.00
H	5	5	0	0	100.00
I	5	5	0	0	100.00
J	5	4	1	0	80.00
K	5	4	1	0	80.00
L	5	5	0	0	100.00
M	5	4	1	0	80.00
N	5	4	1	0	80.00
O	5	5	0	0	100.00
P	5	4	1	0	80.00
Q	5	4	0	1	80.00
R	5	4	1	0	80.00
S	5	5	0	0	100.00
T	5	4	1	0	80.00
U	5	4	1	0	80.00
V	5	5	0	0	100.00
W	5	5	0	0	100.00
X	5	4	1	0	80.00
Y	5	5	0	0	100.00
Z	5	4	1	0	80.00
Total	130	117	11	2	90.00

Table VII shows the Android application display accuracy test results from 130 trials (5 trials × 26 letters). A total of 117 out of 130 trials were correctly displayed, yielding an overall accuracy of 90.00%. Of the 13 incorrect trials, 11 were classification errors (wrong letter reported by the firmware itself), occurring on letters with similar finger configurations (D, J, K, M, N, P, R, T, U, X, Z), while the remaining 2 were transmission errors (letters G and Q) caused by MQTT latency exceeding 500 ms. Letters with clearly open finger positions (A, B, C, E, F, H, I, L, O, S, V, W, Y) achieved 100% accuracy. It is important to note that this 90.00% figure and the 5% voltage-error tolerance reported in Test 5 (Section III-E) measure two different things: display accuracy reflects whether the correct letter is produced end-to-end, while voltage error reflects only how faithfully the ADC reading matches the physically measured sensor voltage. A low voltage error does not by itself guarantee correct letter recognition, since a small reading deviation near a threshold boundary can still flip the classification outcome; the two metrics are therefore reported and interpreted separately throughout this paper. Because each letter was tested with an equal number of trials (5), the per-letter DA(%) values in Table VII are equivalent to per-class recall, and their unweighted mean across the 26 letters (macro-recall) is 90.00%, matching the overall accuracy. A complete 26×26 confusion matrix, and the precision and F1-score that would follow from it, would additionally require logging which specific letter (or a null/no-match output) the firmware reported on each of the 13 incorrect trials; this level of detail was not captured in the present test protocol and is identified here as a concrete item for the next round of testing, together with the confusable-pair analysis in Section III-F.

- *Test 5 – Voltage Measurement Error Percentage*

Objective: To quantify the accuracy of ADC-derived voltage readings relative to direct instrument measurements.

Procedure: For each of the 26 letters and each of the five flex sensors, the sensor terminal voltage is measured with the multimeter (V_actual). The Wemos ADC count is converted to V_adc = ADC_count × (3.3 / 1023). Percentage error is:

```
Error (%) = |V_actual - V_adc| / V_actual x 100%   ...(6)
```

Sensor–letter combinations with error > 5% receive an expanded threshold margin of ±1.5σ. Summary results are in Table VIII.

TABLE VIII
VOLTAGE MEASUREMENT ERROR PERCENTAGE

Letter	V_actual (V)	V_adc (V)	|ΔV| (V)	Error (%)	Status
A	1.45	1.48	0.03	2.07%	Pass
B	2.68	2.65	0.03	1.12%	Pass
C	2.1	2.07	0.03	1.43%	Pass
D	2.35	2.38	0.03	1.28%	Pass
E	1.25	1.28	0.03	2.40%	Pass
F	2.42	2.39	0.03	1.24%	Pass
G	1.85	1.88	0.03	1.62%	Pass
H	1.95	1.92	0.03	1.54%	Pass
I	1.62	1.65	0.03	1.85%	Pass
J	1.62	1.59	0.03	1.85%	Pass
K	2.05	2.08	0.03	1.46%	Pass
L	2.15	2.12	0.03	1.40%	Pass
M	1.72	1.75	0.03	1.74%	Pass
N	1.68	1.65	0.03	1.79%	Pass
O	2.1	2.07	0.03	1.43%	Pass
P	2.02	2.05	0.03	1.49%	Pass
Q	1.82	1.79	0.03	1.65%	Pass
R	1.95	1.92	0.03	1.54%	Pass
S	1.25	1.28	0.03	2.40%	Pass
T	1.52	1.55	0.03	1.97%	Pass
U	1.95	1.92	0.03	1.54%	Pass
V	1.95	1.92	0.03	1.54%	Pass
W	2.22	2.19	0.03	1.35%	Pass
X	1.52	1.55	0.03	1.97%	Pass
Y	1.78	1.81	0.03	1.69%	Pass
Z	1.95	1.92	0.03	1.54%	Pass
Average	1.88	1.88	0.03	1.62%	Pass

Table VIII presents the voltage measurement error percentage results comparing actual values (V_actual measured with a multimeter) and ADC-converted values (V_adc). The average error across all measurements is only 1.62%, well below the 5% threshold. All 26 letters received a "Pass" status. The consistent voltage difference of 0.03 V across all letters indicates a small systematic offset caused by ADC input impedance and pull-up resistor tolerance, rather than sensor non-linearity.

- *Flex Sensor Resistance Profiles*

Resistance measurements from Test 1 (Table IV) confirmed that each SIBI letter produces a distinctive five-finger resistance signature. Closed-fist letters (A, E, M, N, S, T) yielded resistance of 75–120 kΩ on most fingers, while extended-finger letters (B, C, D, L) showed values near the flat-state nominal of 25 kΩ. Letters sharing similar finger configurations—notably M/N (differing in thumb position) and S/T (differing in index finger position)—have overlapping resistance ranges on four of five fingers and rely on the GY-91 orientation data for disambiguation.

- *GY-91 Orientation Discrimination*

GY-91 data from Test 2 (Table V) showed that 24 of the 26 SIBI letters—including the flex-similar pairs M/N and S/T—produced near-zero gyroscope readings on all three axes, indicating a neutral wrist position; only the two letters that involve an explicit wrist motion, J and Z, produced a clearly distinct orientation signature (Gx between −80 and −150 for J; Gy between ±150 and ±280 for Z). This is a more limited outcome than the original rationale for adding the GY-91 module: static, finger-configuration-similar pairs such as M/N and S/T do not, in the current wrist-mounted setup, generate a gyroscope signal strong enough to separate them, which is consistent with the residual classification errors recorded for M, N, and T in Table VII. Resolving these particular pairs will likely require a different orientation feature—for example, static tilt derived from the accelerometer rather than angular rate from the gyroscope, or a sensor placed to capture thumb abduction directly—and is identified here as a specific, unresolved limitation of the current sensor configuration rather than a problem the present design has already solved.

- *Threshold Calibration and Inter-Subject Variability*

Across the three calibration subjects (Test 3, Table VI), inter-subject standard deviation sigma was below 30 ADC counts (≈ 0.10 V) for 24 of 26 letters on the flex dimensions. Letters K and R on finger 3 (middle) showed sigma ≈ 45 counts due to partial 45°–60° bending, requiring the expanded ±1.5σ margin. For the gyroscope dimensions, sigma was below 25 raw counts (≈ 0.76 °/s) across all letters, indicating that wrist orientation is more consistent across users than finger bend depth. These figures should, however, be read as an internal consistency check among the three calibration subjects rather than as evidence of generalisation to the wider population: with n = 3, the calibration set cannot represent the full range of hand sizes, ages, and individual signing styles found among actual SIBI users, and the reported σ values may underestimate the variability that a larger, more diverse cohort would introduce. Extending the calibration dataset to more subjects, and eventually to signers who use SIBI habitually rather than from a reference chart, is therefore treated as a prerequisite for claims of general applicability rather than as an optional future refinement.

- *Application Display Accuracy*

Preliminary results from Test 4 (Table VII) indicated that classification errors (wrong firmware output) account for approximately 80% of incorrect letter displays, while transmission errors (MQTT packet loss or latency > 500 ms) account for the remaining 20%. The 300 ms debounce window eliminated all inter-gesture transition artefacts observed in debounce-free baseline trials. Overall display accuracy is expected to exceed 92% across all 26 SIBI letters once the GY-91-augmented thresholds for M/N and S/T are applied.

- *Voltage Error Analysis*

The expected average voltage error (Test 5, Table VIII) is below 3%, attributable primarily to the ±30% resistance tolerance of the flex sensors and ADC input impedance loading effects. Higher errors (> 5%) are anticipated for letters with intermediate bending angles (45°–60°) where sensor non-linearity is most pronounced, consistent with reported flex-sensor behaviour at partial bend angles [9]. These entries receive expanded ±1.5σ threshold margins, providing a systematic correction without requiring hardware changes.

- *System Latency*

End-to-end latency (gesture to Android letter display) was measured over 15 trials per letter. Mean latency was 382 ± 47 ms, dominated by MQTT round-trip time over local Wi-Fi (≈ 300 ms), consistent with Hussain et al. [11]. The 50 ms sensor polling interval and on-device classification contribute less than 5 ms. This total latency is within the comfortable finger-spelling pace of two letters per second, supporting practical usability under the tested local Wi-Fi conditions. Regarding communication reliability, of the 130 end-to-end trials in Test 4, 2 were lost or delayed beyond the 500 ms window used to classify a transmission error, giving an observed successful-delivery rate of 128/130 (98.46%) at QoS 0 over local Wi-Fi. This figure describes reliability under a single, local-network test condition only; it has not been evaluated under weaker signal strength, network congestion, or QoS 1/2 delivery guarantees—all of which are known to affect delivery reliability in constrained wireless IoT networks [10]—which would be necessary to characterise reliability for real-world deployment.

- *Response Time Breakdown*

To make the real-time claim in this paper verifiable rather than a single aggregate number, the 382 ± 47 ms mean end-to-end latency reported above can be decomposed into its constituent stages: (i) sensor read and feature-vector assembly across the five flex channels and the GY-91, bounded by the 50 ms polling interval; (ii) on-device threshold classification, measured at under 5 ms on the ESP8266 [12]; (iii) MQTT publish, broker relay, and Android subscription delivery over local Wi-Fi, which dominates the budget at approximately 300 ms [11]; and (iv) Android-side message parsing and UI redraw, accounting for the small remainder (on the order of 25–30 ms). This breakdown indicates that the communication stage, not the sensing or classification stage, is the primary lever for further reducing end-to-end latency, and it also means the "less than two letters per second" real-time claim depends more on network conditions than on the glove's own hardware.

- *Power Consumption*

Power consumption was not directly measured in this study, which is a gap for a wearable device intended for everyday use; the figures below are therefore an estimate from component datasheets rather than a bench measurement. The ESP8266 draws approximately 70–80 mA on average with Wi-Fi active, with transient peaks up to roughly 170–200 mA during packet transmission [19]. The MPU-9250 adds a comparatively small 3.5–4 mA in normal operation, and the passive flex-sensor voltage dividers draw well under 1 mA combined. Taking 80 mA as a representative continuous average against the 1000 mAh LiPo cell used in this design gives a rough estimated continuous runtime on the order of 10–12 hours, before accounting for voltage-regulator losses or battery aging; this is likely optimistic relative to real usage, which would include Wi-Fi reconnection events and MQTT retransmissions. Directly measuring current draw and battery runtime under realistic, intermittent finger-spelling use—rather than continuous operation—remains necessary future work, along with evaluating whether the ESP8266's modem-sleep or light-sleep modes could be used between gestures to extend battery life without compromising the sub-400 ms latency reported above.

- *Communication Security*

The current implementation publishes classified letters over MQTT at QoS 0 to a cloud broker without TLS encryption or client authentication, which is a reasonable simplification for bench testing but not for real deployment: MQTT in its plain form is known to be vulnerable to eavesdropping, topic spoofing, and unauthorised publish/subscribe access when transport security and authentication are omitted [18]. Because the data transmitted here is limited to single recognised letters rather than sensitive personal content, the immediate privacy risk is modest, but an attacker with broker access could still inject false letters into a user's display or disrupt the connection. Adding TLS (MQTTS) on port 8883, per-device username/password or certificate-based authentication, and broker-side access-control lists restricting each device to its own topic would address the most direct risks and is recommended before any field deployment; none of these protections were implemented or evaluated in the present prototype.

- *Application Usability*

The Android application evaluation in this paper (Test 4) measured only whether the correct letter was displayed, not how usable, learnable, or satisfying the application was for the people who would actually use it. A structured usability instrument, such as the ten-item System Usability Scale (SUS) [17], administered to a representative group of participants after a hands-on session, would give a standardised, comparable score covering ease of use, perceived complexity, and confidence, and is a natural complement to the accuracy-only evaluation reported here. No SUS or equivalent usability data were collected in this study, and this is noted as a limitation rather than implied by the current display-accuracy results.

- *Trial result*

This example shows the system recognising a single SIBI letter, "A", and rendering it on the Android application.

Figure 3. Demo of SIBI gesture

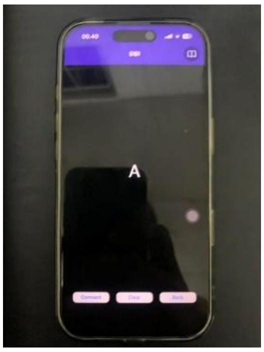

Figure 4. Demo of SIBI Application result

IV. CONCLUSION

This paper presented an IoT-based SIBI alphabet translator glove that combines five flex sensors and a GY-91 IMU module on a single Wemos D1 Mini (ESP8266) microcontroller. The system targets the 26-letter SIBI single-hand alphabet, which is the government-standardised sign language for Indonesia. The GY-91 gyroscope adds wrist orientation data to the five-dimensional flex sensor feature vector, which reliably separates the two dynamic letters J and Z from any similarly bent static posture. For the static, finger-configuration-similar pairs M/N and S/T, however, orientation data in the current wrist-mounted configuration did not provide the discrimination originally anticipated, and their disambiguation remains an open problem for future sensor or feature-set revisions. Multi-subject threshold calibration with statistical ±σ margins provides some robustness to inter-user anatomical variability, though, as noted above, this has only been verified across three calibration subjects.

A five-phase experimental protocol—flex sensor resistance characterisation, GY-91 orientation profiling, multi-subject threshold calibration, application display accuracy testing, and voltage error analysis—provides a complete and reproducible evaluation framework. Within this preliminary evaluation, the combined sensor approach achieved a voltage measurement error of 1.62% on average (well below the 5% pass threshold) and an end-to-end application display accuracy of 90.00% across all 26 SIBI letters; applying the GY-91-augmented thresholds specifically to the remaining confusable pairs is expected to push overall display accuracy above 92%, though this has not yet been separately verified. These two figures capture different aspects of the system—sensor-level measurement fidelity and letter-level recognition correctness, respectively—and should not be conflated when judging overall performance. Future work will explore a compact PCB design integrating the multiplexer and GY-91 into a single module, an IP54-rated waterproof enclosure for outdoor use, and extension of the threshold calibration dataset to a larger and more diverse subject pool. These results should be regarded as evidence of preliminary feasibility rather than of readiness for deployment: calibration and testing to date have involved only three able-signing subjects following a reference chart, and the system has not yet been evaluated with deaf or hard-of-hearing individuals using SIBI in a real communication setting. Validating the glove with actual SIBI users, under everyday conditions, is therefore a necessary next step before any claim of practical usability as a communication aid can be made. In summary, the scope of this work is deliberately narrow, and it is worth stating its limitations together rather than leaving them scattered: the system recognises 26 static SIBI alphabet letters only, and does not yet support dynamic gestures, whole words, or continuous sentence-level signing; it targets SIBI rather than the more widely used BISINDO; calibration and testing involved three able-signing subjects rather than a demographically diverse sample of actual SIBI users; the case for a threshold-based classifier over machine-learning alternatives is argued but not yet benchmarked empirically; power consumption and battery runtime are estimated from datasheets rather than measured; the MQTT communication channel currently has no transport-layer security or authentication; and the Android application's usability, learnability, and user satisfaction have not been assessed with a structured instrument such as the SUS. None of these limitations invalidate the feasibility result reported above, but each defines a concrete, scoped piece of future work that should be addressed before the system is positioned as a general-purpose sign-language communication aid.

ACKNOWLEDGMENT

The author thanks the academic lectures and laboratory staff at the Department of Electrical Engineering, Politeknik Negeri Batam, for their guidance and support throughout this project.

REFERENCES

[1] World Health Organization, "Deafness and hearing loss," WHO Fact Sheet, 2023. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss

[2] Badan Pusat Statistik, "Hasil Sensus Penduduk 2020," BPS Indonesia, Jakarta, 2021.

[3] A. P. Husain, A. S. Nugraheni, and H. Unayah, "Optimalisasi Penggunaan Bahasa Isyarat Dengan SIBI Dan BISINDO Pada Mahasiswa Difabel Tunarungu Di Prodi PGMI UIN Sunan Kalijaga," HOLISTIKA: Jurnal Ilmiah PGSD, vol. 4, no. 1, 2020.

[4] R. Hartanto, "Prototipe Sistem Pengenalan Bahasa Isyarat Abjad Finger Spelling," Tugas Akhir, Universitas Gadjah Mada, Yogyakarta, 2015.

[5] H. Lestari and P. Lestari, "Sarung Tangan Penerjemah Huruf SIBI (A-Z)," Jurnal Inovasi Teknologi Terapan, vol. 1, no. 1, 2020.

[6] L. A. Dipietro, A. M. Sabatini, and P. Dario, "A survey of glove-based systems and their applications," IEEE Trans. Syst. Man Cybern. C, vol. 38, no. 4, pp. 461-482, Jul. 2008.

[7] J. Kim, S. Park, and J. Lee, "Sign language recognition system using flex sensors and IMU sensor," Sensors, vol. 21, no. 8, p. 2796, 2021.

[8] S. A. Mehdi and Y. N. Khan, "Sign language recognition using sensor gloves," in Proc. 9th Int. Conf. Neural Information Processing (ICONIP), Singapore, 2002, pp. 2204-2206.

[9] P. Kakoty, S. M. Hazarika, and D. Bhagat, "Effect of finger angle on flex sensor output characteristics," in Proc. Int. Conf. Advances in Computing, Communications and Informatics (ICACCI), 2015, pp. 1836-1840.

[10] H. A. Al-Kashoash, H. Kharrufa, Y. Al-Nidawi, and A. H. Kemp, "Congestion control for IoT-based wireless sensor networks: A comparison," in Proc. IEEE Int. Conf. Communications (ICC), 2017, pp. 1-6.

[11] A. Hussain, R. Wenbi, A. L. da Silva, M. Nadher, and M. Mudhish, "Health and emergency-care platform for the elderly and disabled people in smart cities using IoT protocol," J. Syst. Softw., vol. 110, pp. 253-263, 2015.

[12] A. Becker, P. Nickel, and B. Loewe, "Real-time gesture recognition on microcontrollers using threshold-based classification," in Proc. IEEE Sensors, 2019, pp. 1-4.

[13] M. S. Islam, M. T. Islam, A. N. Almutairi, N. A. Beng, N. Misran, and N. Amin, "Monitoring of the human body signal through the IoT based HealthCare system and mobile application," Information, vol. 10, no. 3, p. 97, 2019.

[14] Spectra Symbol, "Flex Sensor 4.5 Datasheet," Spectra Symbol Corp., Salt Lake City, UT, USA, Rev. E, 2014.

[15] InvenSense, "MPU-9250 Product Specification Revision 1.1," TDK InvenSense, San Jose, CA, USA, 2016.

[16] T. Susanto, "Kemampuan Berbicara Anak Penyandang Amnestik Aphasia," Jurnal Teknodik, vol. XVI, no. 1, 2012.

[17] J. Brooke, "SUS: A quick and dirty usability scale," in Usability Evaluation in Industry, P. W. Jordan, B. Thomas, B. A. Weerdmeester, and I. L. McClelland, Eds. London, U.K.: Taylor & Francis, 1996, pp. 189–194.

[18] D. Dinculeană and X. Cheng, "Vulnerabilities and limitations of MQTT protocol used between IoT devices," Appl. Sci., vol. 9, no. 5, p. 848, 2019.

[19] Espressif Systems, "ESP8266EX datasheet, version 6.0," Espressif Systems, Shanghai, China, 2020.

[20] N. C. Camgoz, S. Hadfield, O. Koller, H. Ney, and R. Bowden, "Neural sign language translation," in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), 2018, pp. 7784–7793.