Heart Rate and Heart Rate Zone Monitor: Alpha Prototype Performance Results

Tongnian (Minnie) Wang

***Abstract*—This report evaluates the performance of a Heart Rate and Heart Rate Zone Monitor based on quantification of accuracy and affordability. The Product utilised a PIC18F87K22 microcontroller and an infrared biometric sensor. Findings revealed a high accuracy in Heart Rate Zone results but a poor accuracy in Heart Rate measurement. The report concludes with recommendations for improvements through optical design alterations and refinement of calculation methods.**

I. Introduction

GLOBAL health initiatives have successfully promoted an increased public focus on health in recent decades, driving the radio-telemetry equipment industry. The wearable fitness-tracking device sector relies on the valuable insights derived from accurate Heart Rate Monitoring (HRM) [**1**]. Heart Rate (HR) is regulated by the multiple branches of the autonomic nervous system[**2**], making it a prognostic biomarker for health. An elevated resting heart rate of above 80 beats per minutes is associated with a 45% higher risk for all-cause mortality[**3**]. With a direct correlation between elevated resting heart rate and coronary artery disease, strokes and premature mortality; HRM is capable of predicting longevity and cardiovascular diseases [**4**]. Targeted Heart Rate Zone (HRZ) training involves close monitoring of HRZ during aerobic training, maintaining your heart rate within a specific range. This technique allows for optimal training benefits and prevents overtraining, injuries and detraining [**5**].

Radio-telemetry device producers aim to enhance affordability, accuracy and reliability. Notable heart rate monitors include the Polar H7 Chest Strap and Apple Watch 3, which have a Lin's Concordance Correlation Coefficient (LCCC) of 0.98 and 0.96 respectively, indicating near-perfect measurements of HR [**6**].

This report examines the performance of the alpha prototype of a Heart Rate and Heart Rate Zone Monitor constructed using a PIC18 microcontroller interfaced with an optical sensor. Its performance and limitations will be examined through quantifying accuracy and affordability. Following the performance results, a proposition on possible modifications and improvements will be presented for future prototype development phases.

For the remainder of this report, the subject of discussion (the Heart Rate and Heart Rate Zone Monitor) will be referred to as *'the Product'*.

II. High Level Design

Software High Level Workflow

The program begins with the initialisation of the hardware components and the peripherals, including the configuration of the Timer0 module, Liquid Crystal Display (LCD), Universal Asynchronous Receiver/Transmitter (UART) and keypad. The HRZ boundary multipliers are loaded into the Electrically Erasable Programmable Read-Only Memory (EEPROM). Then, the LCD prompts the user to input an age using the keypad. Once a valid two-digit age value has been inputted, the maximum HR and HRZ boundaries are calculated. Then the program enters a polling loop to detect an HR pulse indefinitely, indicated by the shaded area in Fig. 1.

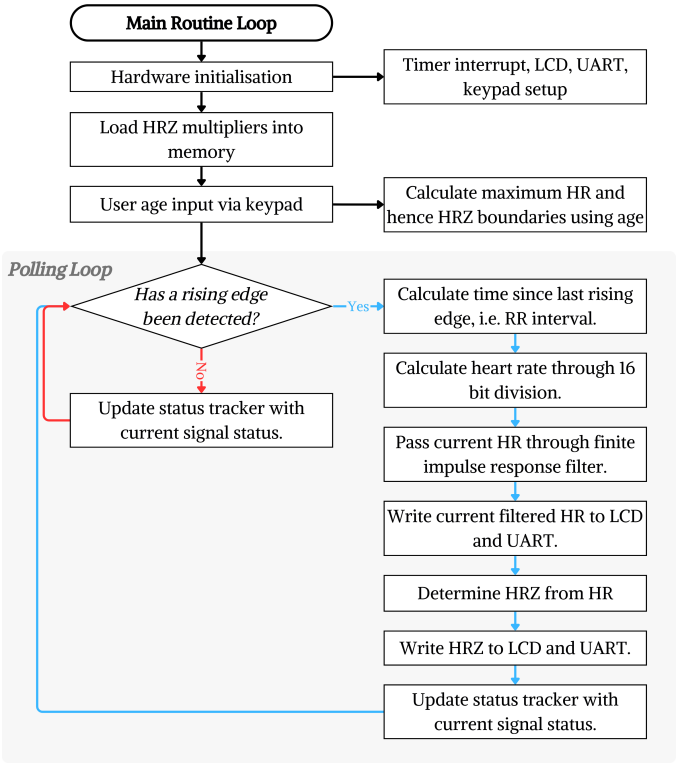

Fig. 1. The high level software workflow of the Product. The software begins with initialisation, then reads age input. Then the program enters a polling process. When a rising edge has been detected, this is interpreted as a pulse occurrence, the program proceeds to carry out the steps indicated by the blue arrows. Otherwise, the system updates the status tracker and continues polling.

During polling, the microprocessor continuously queries the input signal from the sensor. When no high pulse is detected, the system continues to query. When a pulse is detected, the loop exits polling to calculate the heart rate from the R-R interval. The R-R interval is the time duration between successive R-peaks in heart rate. In the Product, this is the time between successive rising edges [**7**]. Divide 60 by the R-R interval in seconds to give an HR in the units beats

per minute (bpm). The HR is then passed through a Finite Impulse Response (FIR) analogue filter to dampen the effect of stochastic jumps in HR [**8**]. The relevant HRZ is determined, then both the HR and HRZ are outputted to the LCD and saved into a text file via UART. Once the results have been outputted, the system returns to polling until another pulse has been detected.

Hardware Design

The hardware structure of the Product was built on an EasyPIC PRO v7, a PIC Development Board with an integrated PIC18F87K22 microcontroller. The high-performance microcontroller operates at frequency $F_{instruction} = 16$ MHz, i.e. each instruction cycle period $= 62.5$ ns. The microcontroller contains peripherals such as Timers and UART, both utilised by the Product. For the Product, the board and all components operate at $3.3\,V$.

The HR detection functionality of the Product was provided by the Grove Ear-clip Heart Rate Sensor (GHRS). GHRS is intended to be worn either on the ear lobe or the finger of the user. GHRS works on the principle of photoplethysmography by using an infrared emitter and a photodetector [**9**]. Blood absorbs light differently in the systolic and diastolic phases of the heartbeat due to its oxygen content. When there is a variation in the absorption of light, the photodetector interprets this as a pulse and outputs a high digital signal to indicate a heartbeat.

The 4x4 keypad is a passive device with keys arranged in a matrix configuration allowing the user to input a value. Under each key is a switch that, when pressed, completes the circuit between a specific row and column and can be detected by a change in voltage on specific pins.

The trans-reflective 2×16 LCD is composed of liquid crystals, which can vary their molecular arrangement in response to an electric field. The controlling of the orientation of liquid crystals allows for the display of characters through specific pixel arrangements. In the Product, the LCD was used to output messages to prompt user age input and output HR and HRZ results.

A USB-UART serial cable was connected to the PC, allowing serial data to be communicated from the microcontroller.

III. Low Level Design

Software

The initialisation of the programme required the correct configuration to set up the LCD, UART and the keypad. In the case of the LCD, this required clearing LATB and setting TRISB such that RB0:5 all outputs. The UART setup required setting the Baud rate to 9600, enabling transmission through TXEN and setting TRISC such that PORTC_TX1_POSN is an output. For the keypad, PORTE was set as an input and the on-chip resistors were pulled up. The initialisation also required the initialisation of global variables used across modules, including the overflow counter, HR and HR_{max}. The software program can be found on Github here.

The age-dependent HRZs used are in Table. I. The useful result here is the fraction representation of the upper boundary of each HRZ. The numerator of each fraction was stored in EEPROM.

HRZ	Percentage of HR_{max} / %	HR_{max} of HRZ
0	$0-55$	$11/20$
1	$55-65$	$13/20$
2	$65-75$	$15/20$
3	$75-85$	$17/20$
4	$85-90$	$18/20$
5	$90-100$	$20/20$

Table I. Heart Rate Zones (HRZs) as a percentage of the maximum heart rate(HR_{max}). Represented as quotients, the upper boundaries of each HRZ have a highest common factor of 5 [**10**]. The 'HR_{max} of HRZ' column indicates the fractional multiplier, when multiplied with HR_{max}, outputs the maximum HR of that specific HRZ.

The timer interrupt was configured after the EEPROM has been loaded, since writing to the EEPROM required all interrupts to be disabled. The timer-interrupt subroutine was located at program memory 0x0008, setting its high-priority status. Timer0 was configured to be a 1 MHz 16-bit timer, overflowing once every 0.065535 s. When the timer overflows, it triggers the Timer0 Interrupt Flag (TMR0IF), initiating the high-priority interrupt subroutine and increments the overflow counter by 1. After the overflow counter has been incremented, Timer0 is reset and the interrupt is exited.

The program then enters a loop to decode the keypad input. The column (RE0:3) and row (RE4:7) are read separately and combined using OR logic. A subroutine decodes the input through a chain of comparison loops. All valid inputs on the keypad correspond to a known RE0:7 value, given the row and column it belongs to - this is unique for each key; if the input matches a valid RE0:7 value, the decoding subroutine returns with the relevant value. If the input is invalid, the keypad will return to read a new value until a valid input has been detected. To find the age from two-digit inputs, the ones-digit was assumed to follow the tens- digit. The two digits required different decoding subroutines: the tens-digit subroutine will return with $n \times 10$ in the Working Register (WREG), and the ones-digit subroutine will return with n - where n is the corresponding key on the keypad that has been pressed. This sets the range of valid input as: $0-99$. Once both digits have been received, the two values are summed to give the age. The maximum heart rate (HR_{max}) is calculated by a subtraction: $HR_{max} = 220 -$ age.

With HR_{max} calculated, the upper boundary of each HRZ can be calculated. Since floating point numbers are difficult to deal with in Assembly code, the percentgae multiplication is broken down by presenting each percentage as a fraction. HR_{max} was first multiplied by the numerator stored in EEPROM, and then divided by 20, the common denominator as seen in Table. I, in a 16-bit division subroutine. This value is then saved in a designated address for access later.

Division in assembly was performed by subtracting the denominator from the numerator, whilst keeping a counter that increments by one whenever the subtraction was possible. The division loop exits when this subtraction was no longer possible, i.e. the numerator was smaller than the denominator. The counter was returned as the quotient of the division subroutine. In the case of a 16-bit division, the need to borrow

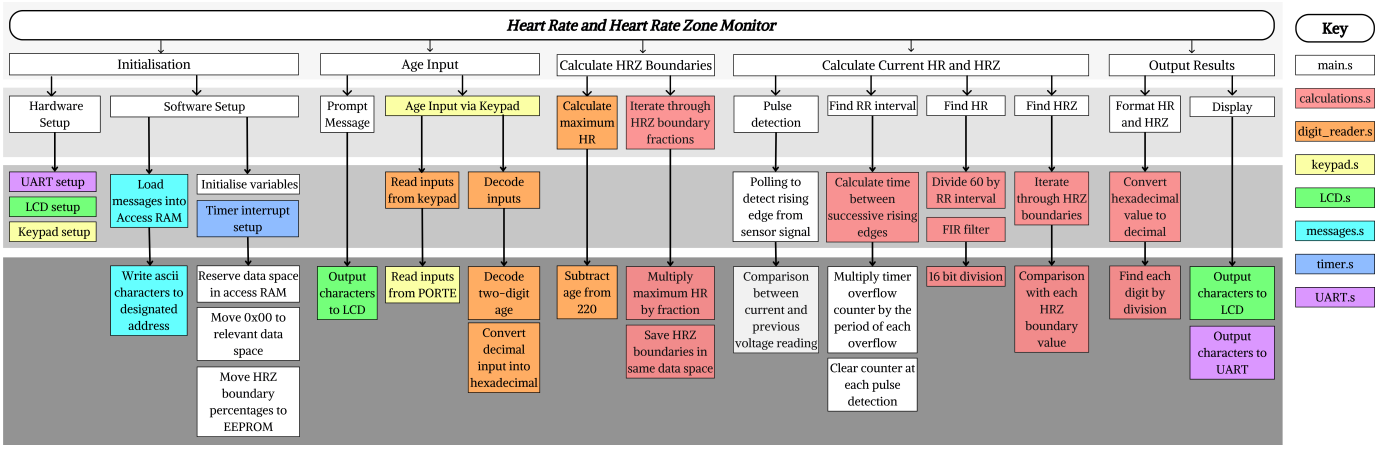

Fig. 2. Top-down modular diagram of the Product's software, where colours represent the location of each subroutine. Organised left to right temporally, with complexity decreasing from top to bottom.

from a higher byte was considered by comparison of the lower bytes of the numerator and denominator. When the numerator was smaller than the denominator, check if borrowing from the higher byte was possible. If borrowing was possible, then borrow, otherwise the division subroutine was terminated.

The program then entered the polling loop that scans for a rising edge from the GHRS signal at RD0. The pulse detection process was achieved by the use of a status tracker (RJ0), the tracker represented the status of the signal at the previous polling instance. At every polling instance, the current signal was compared to RJ0. If RJ0 was low and RD0 was high, this was interpreted as a rising edge. Otherwise, no rising edge has been detected: RJ0 was updated with the current signal and the program continued to poll until a rising edge was detected.

When a rising edge has been detected the program reads from the overflow counter. Since we know the period of overflow, by multiplying the counter by the period of overflow, we can approximate the RR interval. This calculation was approximated and limited in accuracy: the period of Timer0 overflow was approximated as 66 ms. The HR was calculated by a 16-bit division of 60000 ms by the RR interval to give a value of HR in bpm.

$$x'_n = \frac{x_{n-2} + x_{n-1} + 2x_n}{4} \tag{1}$$

The measured HR value was passed through a FIR filter. The design of the FIR filter was a boxcar filter that calculated the moving average to dampen the effect of noise [**11**]. The FIR filter is algebraically presented by Eq. 1, where x_n is the current measurement, x_{n-2} and x_{n-1} are the two previous measurements and x'_n is the filtered current HR measurement. The current measurement x_n has a higher weighting, to preserve the product's sensitivity to changes in HR. The values x_{n-2} and x_{n-1} are updated and the subroutine exits with x'_n in WREG. This HR is then decomposed to its hundreds-, tens- and ones-digit through division by 100, 10 and 1. Each digit was converted to the American Standard Code for Information Interchange (ASCII) representation, saved to a designated address in File Select Register (FSR) and then outputted to LCD and UART a digit at a time.

With the filtered HR, the program then entered a subroutine to determine the relevant HRZ. By initiating the HRZ value as 6, the HR was compared to the upper boundary of each HRZ, from highest to lowest. If the HR was smaller than the upper boundary of that HRZ, the HRZ value decreased by one and the comparison continued for the next HRZ. If the HR was higher than an upper boundary, the subroutine returned with the HRZ value in the WREG, representing the HRZ that the input HR is in. This HRZ value was converted to ASCII representation and outputted to LCD and UART. Once all results have been transmitted, RJ0 was updated with the current signal. The program returns to the polling loop.

The software of the Product utilised a Top Down Modular Design approach. Subroutines were organised in separate modules categorised by their relevance, as demonstrated in Fig. 2. This organisation assisted with debugging and enhanced the readability of the software. Rather than hard-coding addresses, the Product used variables and address designation to help with addressing and locating data. The software also made use of global variables, as a way to minimise the data memory necessary and allowed communication between modules.

Hardware

The GHRS signal was connected to RD0, configured as an input, and continuously queried during polling. The sensor had limitations, including only being able to measure heart rates $\geq$ 30 bpm. This is due to the sensor refreshing every 2 seconds, therefore unable to detect longer R-R intervals. The keypad was connected via RE0:7. Depending on whether the rows or columns are of interest, TRISE for the relevant pins will be set as input. The LCD was connected to RB0:5 via the on-board LCD slot. The messages were written directly to LATB for transmission. An 8-bit data is transmitted as two 4-bit nibbles. The UART transmission used peripheral serial communication. With a USB-UART cable connected from a PC to the development board, data can be transmitted via serial communication.

Fig. 3 contains the details of component interfacing.

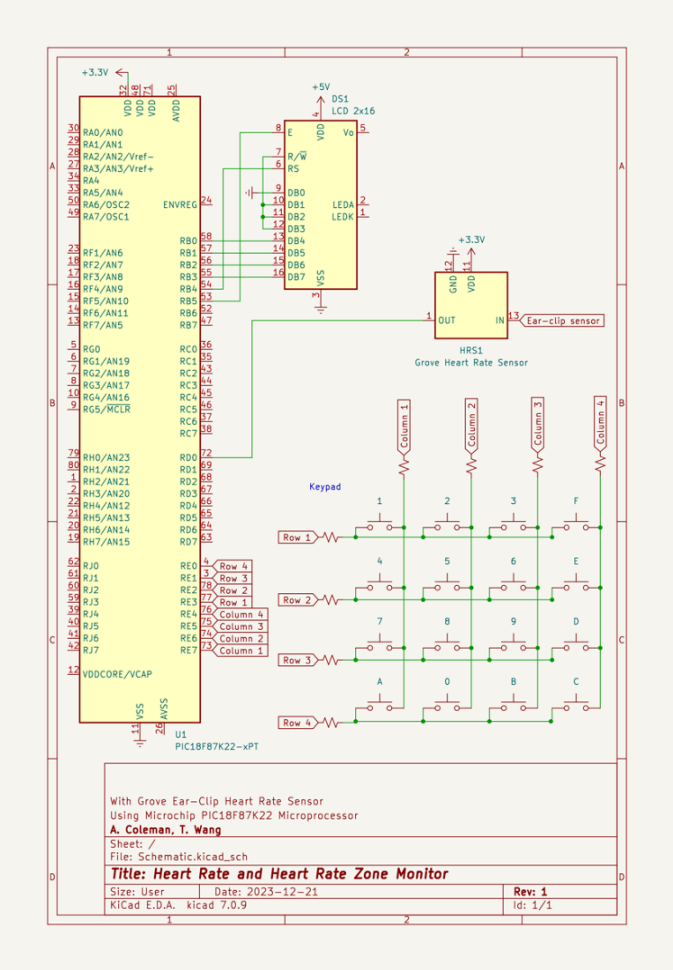

Fig. 3. Schematic diagram for the Product. The circuit was designed to read and decode input from the keypad, calculate HR from the Grove Heart Rate Sensor and output results through the LCD.

IV. Results and Performance

The accuracy of the Product was tested through three different quantifications: Lin's Concordance Correlation Coefficient (LCCC), the mean error and the percentage of measurements in the confidence interval.

$$\rho_c = \frac{2\rho\sigma_x\sigma_y}{(\mu_x - \mu_y)^2 + \sigma_x^2 + \sigma_y^2} \tag{2}$$

LCCC measures the concordance between a new measurement and a standard measurement, this quantifies the agreement between two measures of the same variable. Eq. 2 calculates LCCC, where ρ is the correlation coefficient between x and y, μ is the mean of a measure and σ is the standard deviation of a measure. In the performance testing, the standard measurement is the result outputted by the Polar H10 Chest Strap which we assumed to output the true HR. Performance testing was done for when the user was both stationary and moving, to mimic the usage of the Product. The results are shown in Table. II: the over all accuracy of the Product is poor. The highest accuracy measurement had an LCCC of 0.2439, which is interpreted as 'weak' [**13**].

User	Motion	$\text{LCCC}_{\text{Raw data}}$	$\text{LCCC}_{\text{Ajusted}}$
1	Moving	0.0128	0.0317
1	Stationary	0.1446	0.1578
2	Moving	-0.0066	-0.0496
2	Stationary	0.2439	0.2746

Table II. Lin's Concordance Correlation Coefficient (LCCC) for performance testing data. LCCC_{Raw} was calculated with the raw data with no cleaning or adjustments. $\text{LCCC}_{\text{Adjusted}}$ for FIR was calculated by removing data points heavily affected by the Finite Impulse Response filter. For LCCC ≤ 0.2, the correlation is 'very weak'; for LCCC ≤ 0.4, the correlation is 'weak' as per Altman's interpretations [**13**].

The FIR filter introduced errors to the initial results after each program initialisation, as the variables used in the moving average calculation were initialised as 0. The average percentage error of measurements is observed to decrease as n increases, as seen in Fig. 4. The error introduced by the FIR filter dominated for initial measurements and was negligible for measurements where $n > 7$. By excluding the first 7 data points, all LCCC of the measurements improved as seen in Table. II, pointing towards the inaccuracy contributions from the FIR filter.

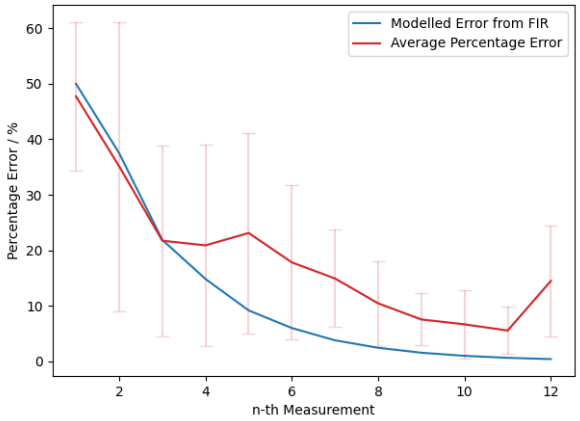

Fig. 4. The average error of measurements as a function of n, where n is the number of measurements taken since initialising the program. The blue line indicates the theoretical error contribution from the FIR filter if a constant HR of 60 was measured. The error from the FIR filter dominates for initial measurements and has a maximum of $\sim 50\%$. This error decays for measurements of higher n.

The second indicator of accuracy was the deviation of the measurement from the true value. The mean error across all measurements was 8.205 ± 0.247. Fig. 5 demonstrates a consistent positive error indicative of a systematic overestimation of HR in the calculations.

The division method considered only the quotient, leading to an underestimation of the results of the division. This in turn led to a consistent overestimation of HR. Whilst the percentage error associated with this decreases over time, the absolute error due to the rounding increases with an increase in HR. This pattern can be observed in the data for User 2: as HR increased, the average error also increased.

The last measure of accuracy was the percentage of data points where the true value lies within its confidence interval. The following sources of errors contributed towards the

inaccuracy of the final result. The RR interval was calculated using only the overflow counter and neglected the time on the timer, this introduces a maximum possible underestimation of 0.065535 ms. For an outputted HR of 60, the true HR can be as low as 56.32. This overestimation amplified for higher HR, possibly contributing to the results demonstrated in Fig. 5.

Secondly, the overflow period approximation: the overflow period was approximated as 66 ms in the calculations, where the true value is 65.535 ms, this introduced a consistent 0.709% overestimation to the calculated RR interval. The absolute error was proportional to the true RR interval: the error was higher for longer RR intervals, therefore lower HR.

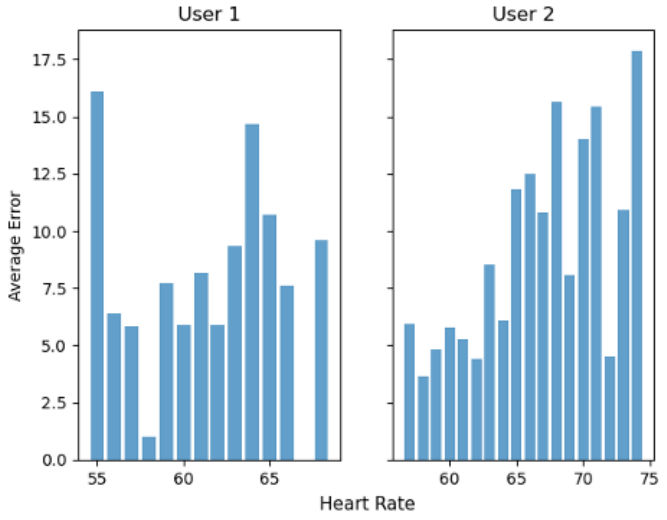

Fig. 5. Average error ($HR_{measured}$ - HR_{true}) as a function of HR. The average is consistently positive, indicating a systematic overestimation. For User 2, as HR increased, the average error also increased.

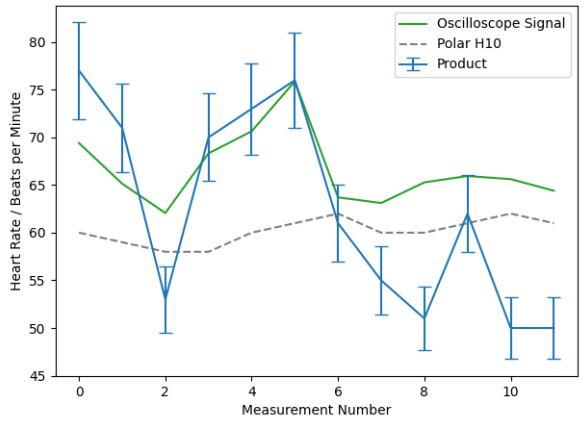

Fig. 6. Comparison of $HR_{Polar\ H10}$ to the $HR_{Oscilloscope\ Signal}$ calculated directly from the signal from GHRS. $HR_{Oscilloscope\ Signal}$ was calculated by finding the RR interval directly from the oscilloscope data and finding a HR from that value. $HR_{Oscilloscope\ Signal}$ was consistently higher than $HR_{Polar\ H10}$, indicating a source of hardware error. Observation of difference between $HR_{Product}$ and $HR_{Oscilloscope\ Signal}$ indicates further errors introduced by the software of the Product during the calculation of HR.

Thirdly, by considering the data from GHRS through an oscilloscope, an average error of 6.46 ± 3.96 was observed between the true and measured HR. The R-R interval was measured directly from the GHRS output. Without involvement from the software of the Product, the discrepancy demonstrated in Fig. 6 indicates an underlying hardware uncertainty.

By combining all the uncertainties above, a confidence interval for the heart rate measurements was defined. For 30.91% of all stationary measurements, the true value was within the confidence interval and 45.24% for moving measurements. The HRZ measurements were more accurate: the measured HRZ was accurate 92.14% of the time for User 1 and 97.73% of the time for User 2.

The product performs better in its accuracy in responding to change. The accuracy in responsivity is defined as when the sign of the change in measured HR is the same as that of the true HR. The accuracy was 80.77% for stationary measurements and 68.54% for moving measurements. This feature is valuable in athletic training, where the athlete should aim to maintain a stable HR and be alerted of changes.

An aim of the Product was to minimise the latency between detecting a pulse and outputting visual updates of the HR and HRZ. The delay between polling detecting a rising edge and both the HR and HRZ results being transmitted to the LCD and UART is $2.9598 + (r \times 0.0025)$ ms where r is the HR. If success is defined relative to the human reaction time: the median human reaction time to visual stimuli is $\sim$ 200 ms [**14**], therefore a $\sim$ 3 ms delay would not be registered. The Product was able to output results nearly instantaneously.

The last measure of performance was affordability: the Product required 200 bytes of data memory and $12,367$ bytes of program memory, these occupied only 5% and 2% of the full capacity of the PIC18F87K22 respectively. This presents the opportunity to downsize, resulting in a cheaper and more transportable device. The Product, with a smaller-scale development board, is priced at $£49.45$. This is significantly less than most wearable technology on the market: with an Apple Watch Series 9 priced at $£399$ and a Garmin Vivosmart 5 at $£129.99$. However, it must be noted that a direct comparison is difficult, as most other products contain other functionalities - this makes it difficult to determine the value of the HR monitoring functionality independently.

V. Updates, Modifications and Improvements

From the results in Fig. 6, it is evident that the GHRS introduced errors in heart rate measurements. Without errors introduced by the software, the GHRS introduced a consistent overestimation of the heart rate. This is likely due to the optical hardware of the GHRS. A possible improvement is replacing the infrared light source with a green LED light. A green LED is more commonly used in industry, due to its shorter wavelength and higher signal-to-noise ratio [**3**]. Moreover, red LEDs and infrared light sources are susceptible to motion artefacts [**15**], this was observed from the results in Table. II, where the moving measurements were consistently less accurate. This inaccuracy during motion can also be accounted for by the poor mechanical design of the GRHS. The ear clip, when worn either on the ear or finger, does not provide

Table III. Product Specifications

Product Features	Age input via keypad
	Calculation of age-dependent heart rate zones
	Calculation of heart rate from R-R interval
	Determination of heart rate zone given heart rate
	Finite impulse response filter
	Display of heart rate and heart rate zone via LCD
	Measurement results exported in a text file via UART
Process Peripherals Used	Timer0 module
	Interrupt controller
	UART Serial Communication Module
External Devices Used	Grove Ear-clip Heart Rate Sensor
	4x4 Keypad
	2x16 Liquid Crystal Display
	Serial Communication Cable
Size of Compiled Code	$2,367$ bytes
Size of Memory Resources Used	200 bytes
Range of Age Input	$01 - 99$
Range of Valid Heart Rate	$001 - 256$
Update Delay [1]	$2.9598 + (n \cdot 0.0025)$ ms, $n =$ measured heart rate
Heart Rate Measurement: Accuracy [2]	Stationary = 30.91%
	Moving = 45.24%
Heart Rate Measurement: Average Error	Stationary = 15 ± 13
	Moving = 25 ± 28
Heart Rate Zone Measurement: Accuracy [3]	94.94%
Responsivity [4]	Stationary = 80.77%
	Moving = 68.54%

1) Delay between occurrence of a pulse and results outputted to the LCD.
2) Accuracy defined by when the true heart rate lies within the confidence interval of the measured heart rate from the Product.
3) Accuracy defined by when the heart rate zone measured by the Product corresponds to the true heart rate zone.
4) Accuracy defined for when the sign of change in heart rate is the same.

good contact with the user's skin. This led to background light interfering with the photodetector causing a variation in light detected that can be misinterpreted as a pulse. An improved mechanical design of the sensor that adheres better to the human anatomy is capable of eliminating this issue.

The latency between a pulse occurrence and the pulse being detected by polling can introduce a maximum overestimation 0.06535 s to the RR interval. This can lead to an underestimation of the HR by 21.06 bpm for a true HR of 150. One possible way to eliminate this error is by the use of the Capture/Compare/PWM (CCP) Module. The Capture Module is able to capture the value of a timer when a specific external event occurs, for example, a rising edge. This allows for a more precise measurement of the RR interval and the uncertainty would be reduced to the period of the timer, which can be as small as 62.5 ns. However, with a RR interval of ~ 1 s for a HR of 60, the accuracy provided by using CCP is not necessary and the accuracy of the polling process is sufficient. The polling process can be improved by considering both the overflow counter and the time on the counter at the time of the pulse. This method will reduce the uncertainty of the RR interval to the period of the timer.

A possible issue faced by polling is missing a pulse. When no pulse has been detected, the Product is polling at a frequency of once every 625 ns due to instructions that update the status tracker when no pulse is detected. The GHRS signal remains high for 0.24595 ± 0.02375 s, this is significantly longer than the polling time period 625 ns, hence we can make the assumption that no pulse will be missed. Another source of error occurs when the pulse is high but the polling process is in the 'no pulse' loop, and the timer overflows. This introduces an overestimation of the RR interval by 0.065535 s. The Timer0 module can be configured with a higher frequency to increase the accuracy of the RR interval calculation, as the overflow period will be shorter.

The FIR filter was able to optimise the trade-off between temporal resolution and precision of measurement by updating the results at every pulse and improving accuracy by calculating the moving average. The use of FIR filters in the Product can be improved: to eliminate the error introduced in the initial measurements, the program can be set to only start outputting results once 3 or more measurements have been made.

VI. Conclusion

In conclusion, the alpha prototype of the Product successfully delivers timely visual updates and accurate Heart Rate Zone information. The Product was responsive to change and could output Heart Rate Zone information accurately. However, the alpha prototype suffered from poor accuracy in Heart Rate calculations and was limited by the optical and mechanical design of the biometric sensor. The Product's key strength was its affordability compared to industry competitors. Future improvements in sensor hardware and software calculations are expected to enhance the Product's accuracy.

REFERENCES

[**1**] F. El-Amrawy and M. I. Nounou, "Are Currently Available Wearable Devices for Activity Tracking and Heart Rate Monitoring Accurate, Precise, and Medically Beneficial?," *Healthcare Informatics Research*, vol. 21, no. 4, p. 315, 2015, doi: https://doi.org/10.4258/hir.2015.21.4.315.

[**2**] M. Bohm, J.-C. Reil, P. Deedwania, J. B. Kim, and J. S. Borer, "Resting Heart Rate: Risk Indicator and Emerging Risk Factor in Cardiovascular Disease," *The American Journal of Medicine*, vol. 128, no. 3, pp. 219–228, Mar. 2015, doi: https://doi.org/10.1016/j.amjmed.2014.09.016.

[**3**]B. W. Nelson, C. A. Low, N. Jacobson, P. Areán, J. Torous, and N. B. Allen, "Guidelines for wrist-worn consumer wearable assessment of heart rate in biobehavioral research," *npj Digital Medicine*, vol. 3, no. 1, Jun. 2020, doi: https://doi.org/10.1038/s41746-020-0297-4.

[**4**] M. Woodward et al., "The association between resting heart rate, cardiovascular disease and mortality: evidence from 112,680 men and women in 12 cohorts," *European Journal of Preventive Cardiology*, vol. 21, no. 6, pp. 719–726, Jun. 2012, doi: https://doi.org/10.1177/2047487312452501.

[**5**] A. Jeukendrup and A. V. Diemen, "Heart rate monitoring during training and competition in cyclists," *Journal of Sports Sciences*, vol. 16, no. sup1, pp. 91–99, Jan. 1998, doi: https://doi.org/10.1080/026404198366722.

[**6**] S. R. Pasadyn et al., "Accuracy of commercially available heart rate monitors in athletes: a prospective study," *Cardiovascular Diagnosis and Therapy*, vol. 9, no. 4, pp. 379–385, Aug. 2019, doi: https://doi.org/10.21037/cdt.2019.06.05.

[**7**] M. H. Kryger, T. Roth and W. C. Dement (2011). *Principles and practice of sleep medicine*. 5th ed. Philadelphia, Pa: Elsevier, pp.226–236.

[**8**] A. V. Oppenheim, A. S. Willsky and I. T. Young(1983). *Signals and systems*. Englewood Cliffs, N.J.: Prentice Hall.

[**9**] K. A. Panicos and A. John (2022). *Photoplethysmography : technology, signal analysis and applications*. London: Elsevier Academic Press.

[**10**] A. T. Scanlan, J. L. Fox, J. L. Poole, D. Conte, Z. Milanović, M. Lastella and V. J. Dalbo (2018). A comparison of traditional and modified Summated-Heart-Rate-Zones models to measure internal training load in basketball players. *Measurement in Physical Education and Exercise Science*, 22(4), pp.303–309. doi:https://doi.org/10.1080/1091367x.2018.1445089.

[**11**] L. R. Rabiner and B. Gold (1975). *Theory and Application of Digital Signal Processing*. Englewood Cliffs, New Jersey: Prentice-Hall, Inc.

[**12**] T. Steiched and N. Cox (2002). 'A note on the concordance correlation coefficient'. *The Stata Journal*, 2, 183-189

[**13**] D. G. Altman (1991). *Practical statistics for medical research*. London: Chapman Hall.

[**14**] S. Thorpe, D. Fize and C. Marlot (1996). Speed of processing in the human visual system. *Nature*, 381(6582), pp.520–522. doi:https://doi.org/10.1038/381520a0.

[**15**] L. Mahloko and F. Adebesin (2020). 'Responsible Design, Implementation and Use of Information and Communication Technology'. *Lecture Notes in Computer Science* (eds Hattingh, M., et al.)Springer International Publishing, Cham. Vol. 12067, pg. x96–107.

APPENDIX A
HEART RATE ZONE INTERPRETATION

Heart Rate Zone	Percentage of HR_{max} / %	Interpretation
0	$0-55$	Below Recovery
1	$55-65$	Recovery / Easy
2	$65-75$	Aerobic / Base
3	$75-85$	Tempo
4	$85-90$	Lactate Threshold
5	$90-100$	Anaerobic

APPENDIX B
LAB CYCLE 1 FEEDBACK: MICROMAGNETICS

Marker Summary Comments: Good systematic tests with aspect ratio Good overall analysis including references to relevant literature You demonstrated good understanding of the basics of micromagnetics Engage with your slides and avoid reading from the slides Put the LLG equation and highlight the different terms Always good to use a figure to explain concepts. For example, anisotropy discussions could be done much better using figures

APPENDIX C
LAB CYCLE 2 FEEDBACK: MICROPROCESSOR

Marker Summary Comments: Nice context provided to jutisfy the need of a product like what you are proposing. Formatting and style looks nice and consistent. The proportion of your report in bullet points is too large. They're benificial from time to time making points clear, but in your case it's a bit too much ($\approx 1/3$ of your report). You'd benefit from a top-down modular diagram, which would include extra information on how you would organise module/routine/subroutine at various levels. After all you promised a 'modular approate' to your software design. Try to estimate your projected performance with measurable quantities, instead of just giving a list of aspects you'd consider.