Building a Social Distancing Warning Device Using a PIC18 Microprocessor

Maike Lenz, Dec 2020

***Abstract*—This project consisted of constructing the prototype for a Social Distancing Warning Device with a PIC18 on an EasyPIC Pro board and programming it with Assembly. The device uses ultrasound pulses from a Parallax PING))) sensor to compute the separation to an obstacle. For separations below $(203.8 \pm 0.6)\mathrm{cm}$, a set of LEDs turns on. Once it gets closer than $(151.0 \pm 0.6)\mathrm{cm}$, a buzzer sounds. The range of measurements extends to $3\mathrm{m}$. This prototype could be adapted to be used in a portable device employing the abovementioned warning signals to maintain social distancing - for example in a queue. The main source of error is the angular dependence of the ultrasonic sensor. The signal is lost for obstacles which are not in the direct line of view of the sensor. The efficacy of the device is therefore limited by the hardware.**

I. INTRODUCTION

DISTANCE measuring devices are widely used for a variety of purposes. In this project, the initial aim was to build a social distancing warning device which gives off a warning signal when the distance to an obstacle is less than $2\mathrm{m}$, and another signal when this separation reaches $1.5\mathrm{m}$. Covid-19 restrictions commonly urge to keep a distance of $2\mathrm{m}$ from other people, especially in queues where a long time is spent in proximity to one another. A portable battery powered version of this prototype could possibly be worn on the outside of clothes to warn the wearer when they are standing too close to the next person in line.

II. HIGH LEVEL DESIGN

The product prototype system consists of four components: the PIC18 microprocessor, a PING))) ultrasonic receiver and emitter, a buzzer, and a set of LEDs. The LED and buzzer are both used to emit warning signals to the wearer when an insufficient separation is calculated by the device. The top down modular diagram of the device is shown in Fig. 1.

III. HARDWARE DESIGN

An EasyPIC Pro V7 board[1] with built in LEDs and a Piezo Buzzer was used to build the prototype device. The only externally wired component was the PING))). A schematic diagram of the hardware design is shown in Fig. 2.

A. PING))) Ultrasonic Sensor

The most important factor in choosing suitable hardware was to find an ultrasonic sensor which can operate at appropriate distances without losing signal strength. In this project, the Parallax PING))) Ultrasonic Distance Sensor #28015[2] was employed. Its range extends to $3\mathrm{m}$[2] which meets the requirements for the device. After emitting a pulse, the PING)))

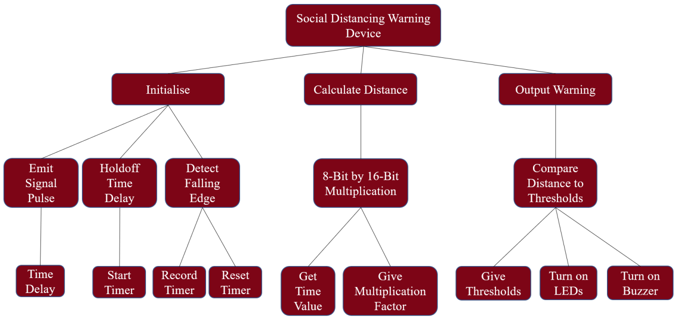

Fig. 1: Top Down Modular Diagram of the device. The measurement process occurs in three stages. The initialisation consists of a pulse being emitted by the PING))), and the width of the returning pulse being measured with by the PIC18 in capture mode. The following calculation converts the timer reading into a distance value. Lastly, the result is compared to the two threshold separations to determine which warning signals need to be emitted.

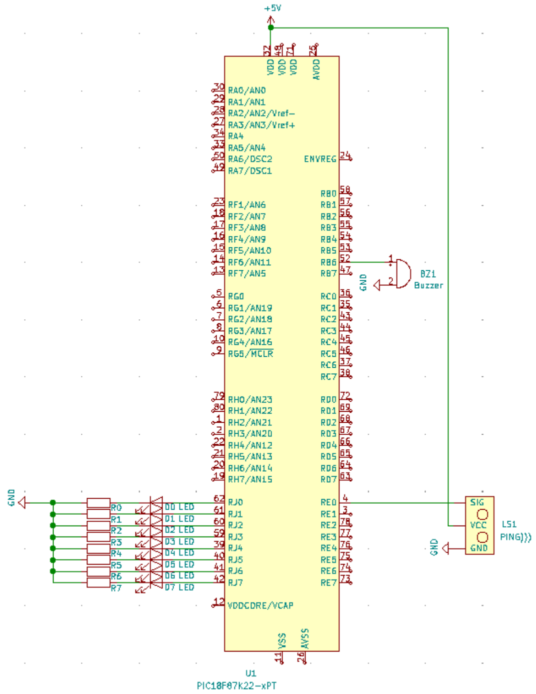

Fig. 2: Schematic Diagram of the hardware used in the device. The buzzer BZ1 was connected to Pin RB6 via the EasyPIC Pro V7 board[1] by default. The signal pin of the PING))) LS1 is connected to Pin RE0. The LEDs of Port J were used to signal a warning.

measures the time required for the echo pulse to return[2]. It then produces a pulse with the width of this time delay with a magnitude of 5V, similar to the input pulse. The timings for this process are shown in Fig. 3.

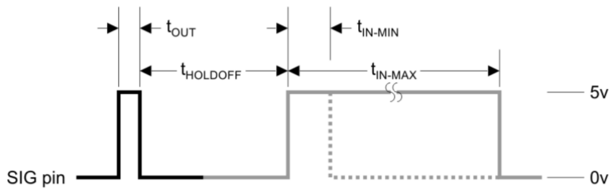

Fig. 3: Schematic diagram of the timings of PING))). A pulse with duration t_{out} is emitted. After holdoff time $t_{holdoff}$, the PING))) produces a pulse of width t_{in} which corresponds to the time the pulse took to echo back to the sensor. Figure from [2].

The delay between the sent and return pulse is the constant holdoff time of 750µs[2]. Both the output pulse and the produced return pulse are on the signal pin of the PING))). To find the separation to an object, the width of the return pulse is determined and converted into a separation.

IV. SOFTWARE DESIGN

A. Capture Mode

The device uses CCP10 with the 16-Bit setting of timer 1 in capture mode to measure the width of the returning pulse. CCP10 was configured to detect the falling edge of the returning signal. The holdoff time is known to be 750µs[2]. This is the time elapsed between the emitted pulse and the received pulse. After the pulse is emitted, this delay was implemented with cascading delay loops. Then, timer 1 is started, and the CCP10 interrupt is enabled. This process is shown in the flowchart in Fig. 4.

The maximum width of the returning pulse occurs when it travels to an object 3m away and back. This corresponds to a delay of about 18.6ms. By the time this delay has been implemented with cascading delay loops similar to the holdoff time delay, CCP10 has captured the falling edge of the returning pulse and stored the 16-Bit timer 1 value in its file registers. This value can then be converted into a separation.

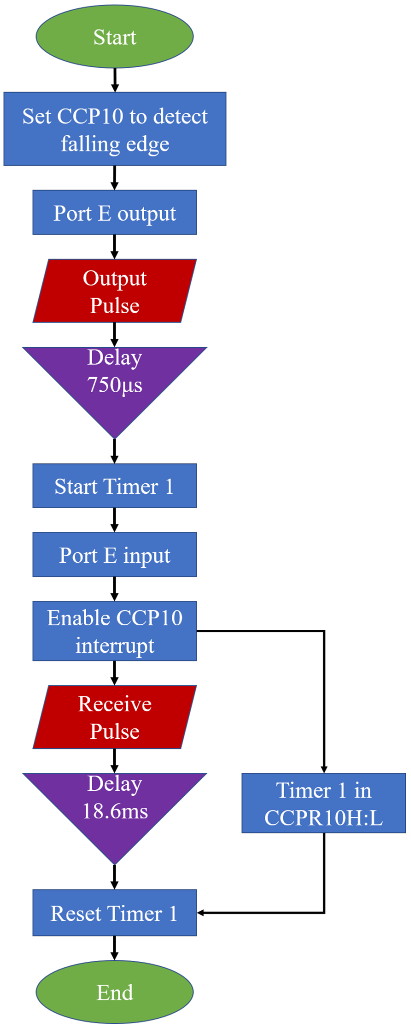

Fig. 4: Flowchart of the Pulse emitting and receiving process with CCP10 using timer 1 in capture mode. The CCP10 interrupt is only enabled after the holdoff time to prevent CCP10 recording the falling edge of the emitted pulse. A 18.6ms delay is implemented which corresponds to the maximum travel time for an object 3m away. During this time, CCP10 captures the falling edge of the returning signal and stores the 16-bit timer 1 value in its file registers CCPR10H:L

B. Calculating Distance

The value CCP10 stored from timer 1 is t_1. The timer was set with the prescale 1:8 such that its incrementation frequency is 2MHz. The separation, s, is then given by equation 1.

$$s = t_1 \times \left(\frac{1}{2} \times \frac{1}{2000000} \times 330 \right) \tag{1}$$

$$s = t_1 \times 82.5 \times 10^{-6} \tag{2}$$

The factor of $\frac{1}{2}$ comes from the separation being half the distance travelled by the pulse. The speed of sound was approximated as 330ms^{-1}. In order to carry out this conversion in the code, the 16-bit timer result is multiplied by a hexadecimal number corresponding to the factor 82.5 from equation 2. This factor was set as the 8-bit hexadecimal 0x53, giving 83. The code contains an 8-Bit by 16-Bit multiplication process, as visualised in Fig. 5.

The result of the conversion is a 24-Bit hexadecimal number stored in three result file registers. It corresponds to the separation in nanometres. This is because of the omitted factor of 10^{-6} in equation 2. Result one, the most significant byte of the separation, was then used to determine whether the separation lies below either of the signal thresholds. This means that there is a limit to the resolution of the possible separations where the warning signals can be implemented. Without taking into account the less significant bytes in result 2 and result 3, the threshold can only be set approximately every 6cm.

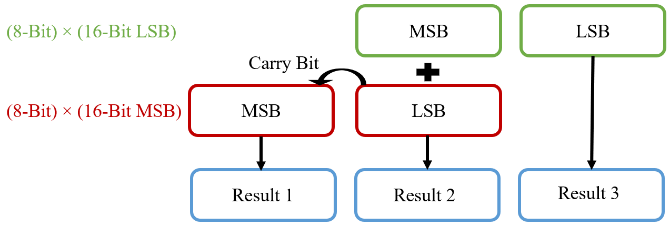

Fig. 5: Schematic diagram of the 8-Bit by 16-Bit multiplication necessary to obtain a distance value. The 24-Bit result is split into three bytes: result 1, result 2, and result 3 from most to least significant. Only result 1 is used in the following analysis.

C. Signals

The signal code compares the MSB of the calculated separation to threshold values and determines which signals need to be implemented. This is shown in Fig. 6.

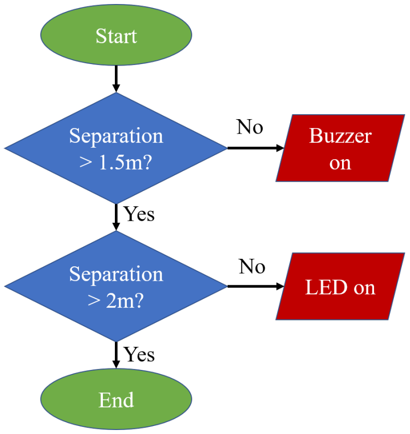

Fig. 6: Flowchart of the signal module. The two thresholds for the LED and the buzzer are approximately 2m, and 1.5m respectively. See the testing section for exact values.

D. Delays

The PING))) sensor required certain delays due to its internal working. In order to implement these, delay loops with decrementing counters were cascaded. Clock counts in the PIC18 occur at frequency 64MHz[3]. Every decrement requires a clock cycle consisting of four clock counts. By counting the clock cycles involved in the loops and adding the remaining ones due to branching, we found the initial values of the counters to best match the required delays. The results of this are shown in Table I.

Time	Required Delay	Device Delay
t_{out}	5µs	4.9µs
$t_{in,max}$	18.5ms	18.6ms
$t_{holdoff}$	750µs	726µs
t_{next}	200µs	242µs

Table I: Depicts the timings that are required by the PING))) sensor versus the implemented delay in the device.

The device delays meet the requirements to a high level of precision. The holdoff time delay is shorter than the PING))) holdoff, which may have a small contribution to the device's instrumental error. However, this deviation is on the order of $14\mu\text{s}$, which is minimal compared to the pulse widths. From Table I, the minimum period of the device can be estimated to be about 19.6ms. This case applies when no signals are set off meaning that the separation is greater than 2m. The maximum operating frequency is therefore 51Hz.

V. RESULTS AND PERFORMANCE

A. Accuracy of Distance Measurement

In order to test the functionality of the device, the threshold at which the LED switches on was changed in the code. For each threshold ranging from 6.6cm (0x01) to 209.7cm (0x1E), the separation at which the LED signal switches off was found with a tape measure. A flat book was held in front of the device perpendicular to its line of sight to maximise the accuracy of the separation values. The results are plotted in the graph in Fig. 7.

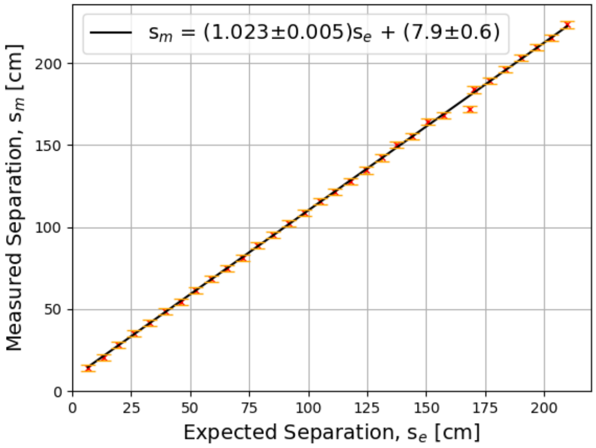

Fig. 7: A graph showing the relationship between the measured separation by the device and the actual separation. The error in the measured distances was $\pm 2\text{cm}$ due to fluctuations in the signal near the threshold. A linear curve fit was performed and gave an offset of $(7.9 \pm 0.6)cm$ corresponding to the instrumental error of the product. The gradient is above one which indicates that the device overestimates separations. The data point at 1.60m is clearly an anomaly and can be neglected. It probably stems from the signal being lost due to the book being held out of the line of sight of the device rather than due to the distance measurement.

The gradient of 1.023 ± 0.005 indicates that the rounding in the separation conversion in the code leads to separations being slightly overestimated by the device. However, within the operating range of the device this error is minimal and does not need to be accounted for. The linear offset of the trend line indicates instrumental error. This error was constant at $(7.9 \pm 0.6)\text{cm}$.

1) Correcting the Instrumental Error:

Each point on the graph in Fig. 7. corresponds to a hexadecimal value that can be set as the threshold. Due to the limited resolution, values in between the data points cannot be chosen as thresholds without involving result 2, the mid significant byte of the separation result, in the analysis. The hexadecimal values which most closely corresponded to the intended thresholds of $2\mathrm{m}$ and $1.5\mathrm{m}$ were chosen.

For the buzzer, the value 0x15 was implemented which corresponds to a measurement of $151\mathrm{cm}$. For the LED, either 0x1E, giving $196.2\mathrm{cm}$, or 0x1D giving $203.2\mathrm{cm}$, could be used. For the purposes of the device, it makes more sense to over-estimate the separation; the latter value was chosen. According to the curve fit covariance matrix, the uncertainty in the offset was $\pm 0.6\mathrm{cm}$. This value is now the absolute error in the signal distances. Implementing these thresholds in the code and measuring at which separation each signal turns off gave the final results:

$$\text{Buzzer: closer than } (151.0 \pm 0.6)\mathrm{cm} \tag{3}$$

$$\text{LED: closer than } (203.8 \pm 0.6)\mathrm{cm} \tag{4}$$

B. Angular Profile

A test was then carried out to determine the angle at which the signal is lost at different separations. To test the device operation with real people, a person of height $1.77\mathrm{m}$ stood at a known separation from the device and moved to the side until the signal was lost. This was repeated with a person with height 1.60m. The device was fixed at a height of approximately 1.30m. The results are plotted in Fig. 8.

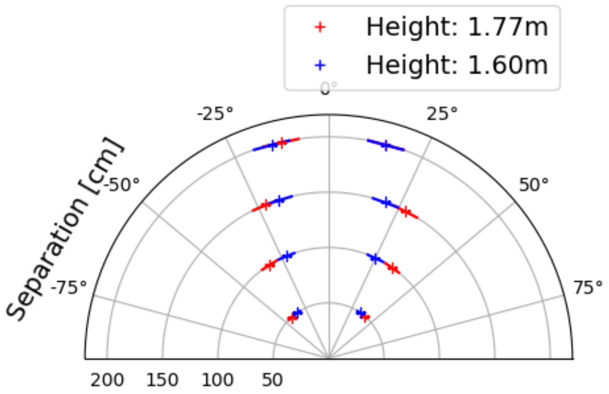

Fig. 8: A plot of the angular profile of the device for a person with height $1.77\mathrm{m}$ and one of height $1.60\mathrm{m}$. Each data point is is the mean of two repeats and corresponds to the angle at which the signal was lost at a range of separations. Measurements were taken with the device at height $1.30\mathrm{m}$. The uncertainty in the angle was approximated to be $5°$ due to fluctuations in the signal before it was lost.

As depicted in the graph in Fig. 8., the distribution appears more narrow for a shorter person. This may be because the device detects more of the shoulders and neck which are narrower. The product should be worn at a lower elevation, perhaps at belt-level to ensure that the bulk of people with a range of heights - including children - are detected. For both people, it is evident that at larger separations the distribution becomes more narrow. This may be because the human body appears more flat the further away it is. The PING))) documentation[2] shows that flat objects have a much more narrow angular profile. The device best measures surfaces which are perpendicular to its line of sight. A rounded object is likely to have a section perpendicular to the signal pulse. This angular dependence on the signal is the main limitation of the functionality of the device. In a real scenario it would need to be able to detect people that are not directly in front of the wearer.

VI. UPDATES, MODIFICATIONS AND IMPROVEMENTS

The main error of the device is the signal being lost when the person moves away from its central line of view. Aligning multiple sensors to cover a greater field of view may improve this. Sensors pointing in different directions would also efficiently cover the peripheral of the wearer. These additional pieces of hardware may slow the functionality of the device down, since all their signals need to be processed. However, with the current maximum frequency of measurements at $51\mathrm{MHz}$, this is not an issue. The device output stability would drastically improve its functionality even if the device was slowed down.

There are a wide variety of possible extensions to this project. Temperature and humidity sensors could be incorporated in the device to give a more accurate value for the speed of sound rather than using a constant value. This would adapt the device to be functional in different environments.

Another modification to the device may be to improve the resolution of the possible thresholds that can be picked for warning signals. To implement this, the mid significant byte of the separation result would have to be involved in the analysis. All three bytes of the 24-Bit separation result are stored in their respective file registers which allows them to be used for extensions to the code.

In terms of the radiation, ultrasound sensors tend to be more stable at large distances than infrared radiation[4] and are therefore an appropriate choice for this device. Although the ultrasound signal worsens for soft materials[4], in the tests the device did detect people in clothing. However, the PING))) may not be the ideal choice in detector due to its susceptibility to detecting accidental signals[5] such as a waving hand. When implementing this prototype ad a real device, different sensors should be tested and compared to evaluate their efficacy in a real life setting.

VII. CONCLUSIONS

This device successfully computes the separation of the device to an obstacle and emits an LED and buzzer warning signal when the separation is too small. The implemented thresholds for these signals are:

$$\text{Buzzer: closer than } (151.0 \pm 0.6)\mathrm{cm} \tag{5}$$

$$\text{LED: closer than } (203.8 \pm 0.6)\mathrm{cm} \tag{6}$$

The thresholds are slightly off the initial aim to set them at $2\mathrm{m}$ and $1.5\mathrm{m}$ due to limited resolution in the hexadecimal threshold values. However, the accuracy in the separation is much smaller than initially expected at only $\pm 0.6\mathrm{cm}$. The main limitations to the functionality of the device are its narrow angular profile and the resulting instability of the signal when the obstacle is not directly withing the line of view of the sensor. This could be corrected by involving multiple sensors in the device to cover a larger area. Although social distancing alone is insufficient in preventing infection[6] from Covid-19, this device offers a useful foundation for a variety of applications involving distance measurements.

VIII. PRODUCT SPECIFICATIONS

Table II depicts the specifications of the Social Distancing Warning Device. The code is available at https://github.com/carawaters/MicroprocessorsDistance.

Product Specifications

Supply Voltage	5V
Range	3m
Accuracy	$\pm 0.6\mathrm{cm}$
f_{max}	51Hz

Table II: Product Specifications of the Device. f_{max} is the maximum frequency of measurements. This occurs when neither the buzzer nor the LED signals go off, hence at separations above $2\mathrm{m}$.

REFERENCES

[1] MikroElektronika, “Easypic pro,” 2013.

[2] I. PARALLAX, “Ping)))™ ultrasonic distance sensor (# 28015) datasheet,” 2009.

[3] M. T. Inc., “Pic18f87k22 family,” 2009-2018.

[4] A. M. Kassim, H. I. Jaafar, M. A. Azam, N. Abas, and T. Yasuno, “Performances study of distance measurement sensor with different object materials and properties,” in *2013 IEEE 3rd International Conference on System Engineering and Technology*, 2013, pp. 281–284.

[5] E. Cole, “Performance analysis of ping))) ultrasonic distance sensor.”

[6] Z. Qureshi, N. Jones, R. Temple, J. P. Larwood, T. Greenhalgh, and L. Bourouiba, “The centre for evidence-based medicine: Evidence service to support the covid-19 response.” [Online]. Available: https://www.cebm.net/covid-19/what-is-the-evidence-to-support-the-2-metre-social-distancing-rule-to-reduce-covid-19-transmission/