Automated Pill Dispenser for Medication Adherence in Elderly Patient

RAMAZAN ONUR ACAR	2020400084
DENİZ BİLGE AKKOÇ	2020400135
MUSTAFA ATAK	2020400042
ONUR ÇERLİ	2020400372
EREN PAKELGİL	2021400243
ARDA VURAL	2020400252

Definition of the Problem

Medication adherence remains a widespread challenge among elderly and chronically ill patients, posing risks to individual health and healthcare resources. There is a need for solutions to help patients reliably follow their medication schedules, thereby supporting their health and lifting the load from caregivers and healthcare providers.

Severity of the Problem

Medication non-compliance (or poorly compliance) in elderly patients is a severe issue, with rates ranging from 40% to 75%, significantly impacting their health and well-being. Common challenges include overuse, forgetting, and altering prescribed schedules, often due to the complexity of managing multiple drugs.

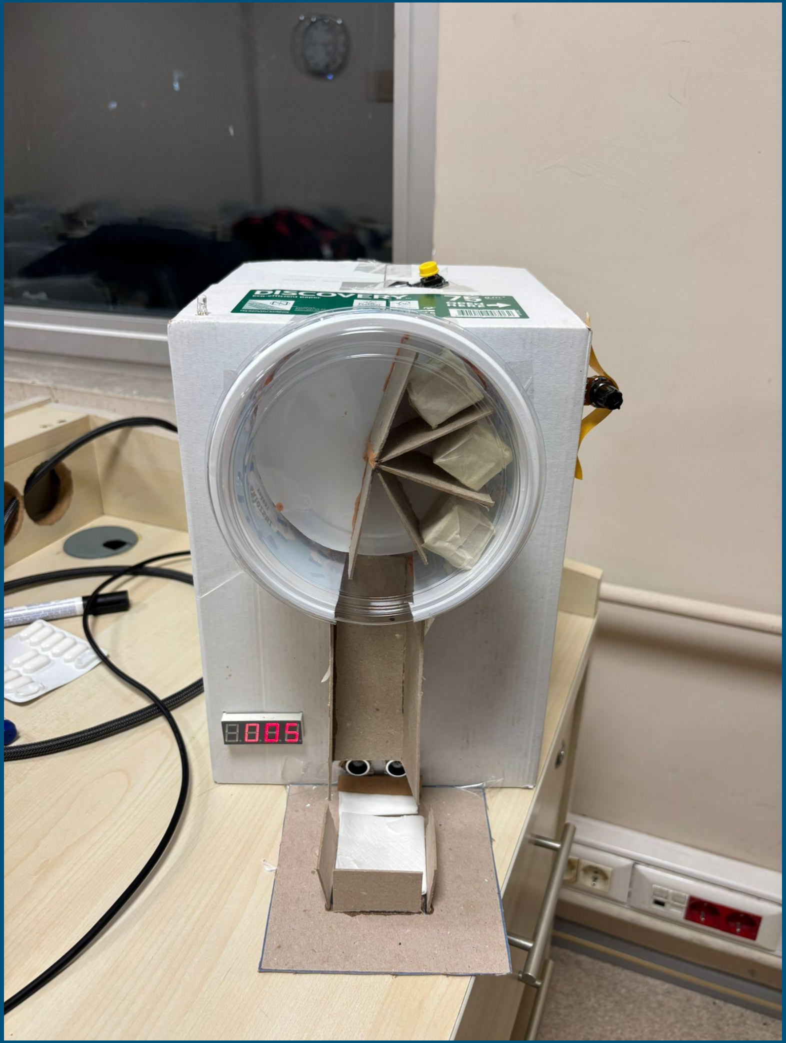
For instance, up to 25% of elderly patients take at least three medications daily, increasing the risk of non-compliance. Additionally, memory impairments from conditions like dementia and depression further complicate adherence, underscoring the need for reliable support solutions (Salzman, 1995).

Proposed & Implemented Design

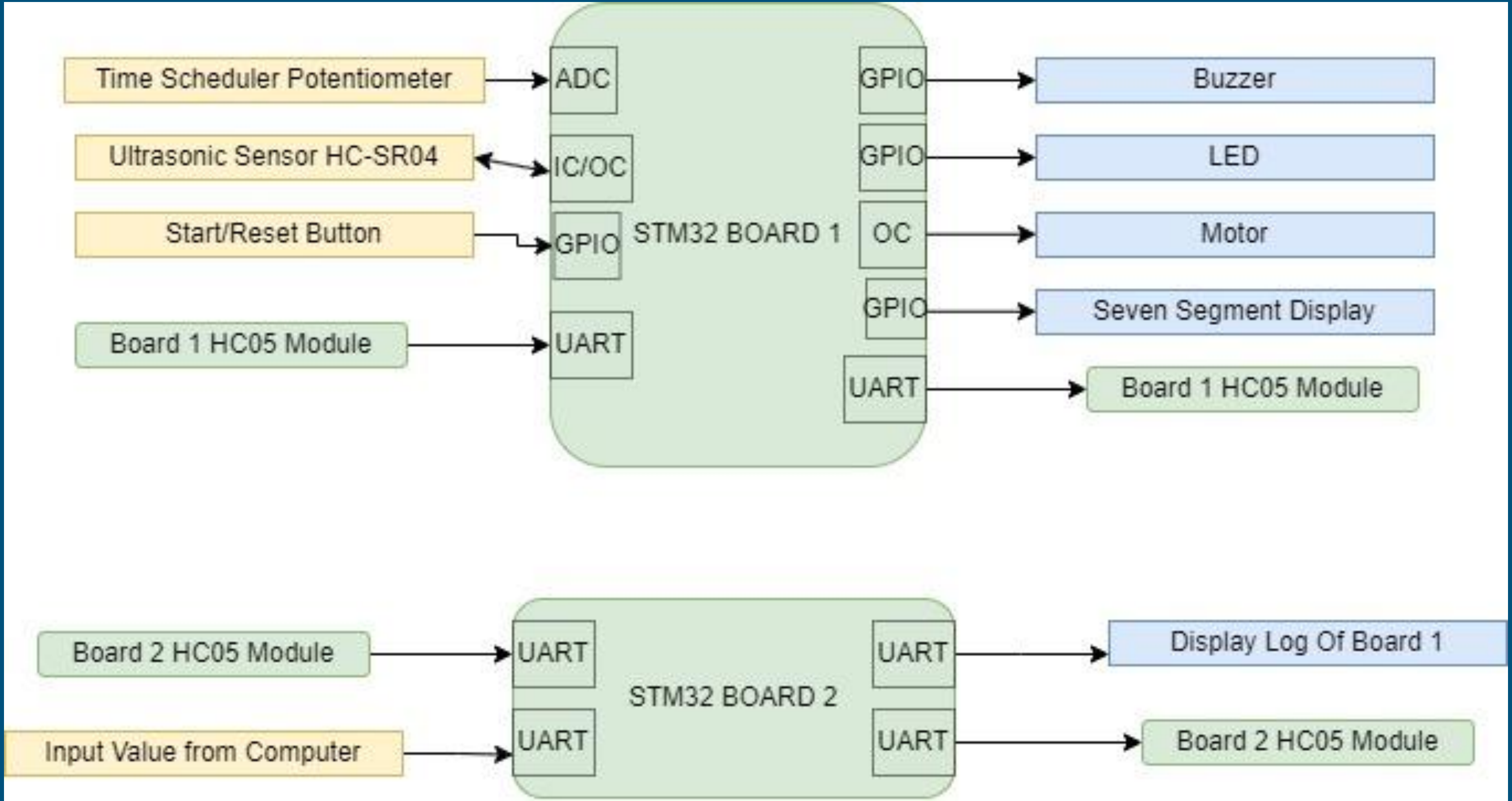
Proposed design: After the values entered by the user into the computer are transferred to the slave board via uart, the pill is dispensed at specified time intervals when the start button is pressed. These dispensed batteries are detected by the Ultrasonic sensor and the patient is notified via buzzer and led. When the patient takes the medicine, the buzzer and led stop. The activating interval of these two elements is regulated by a potentiometer. The system is reset with the reset button.

Implemented design: Taking the time value from the computer, the time remaining until the drug is dispensed is displayed in the seven segment. The system is activated by pressing the Start button. The 3 seven segments are updated every second. When it reaches 0, the pill is dispensed. The ultrasonic sensor detects the drug and the buzzer and led start beeping while the beeping interval can be changed. Once all medicines have been dispensed and received, the system is restored to its initial state. If you want to reset the system during this period, press the reset button.

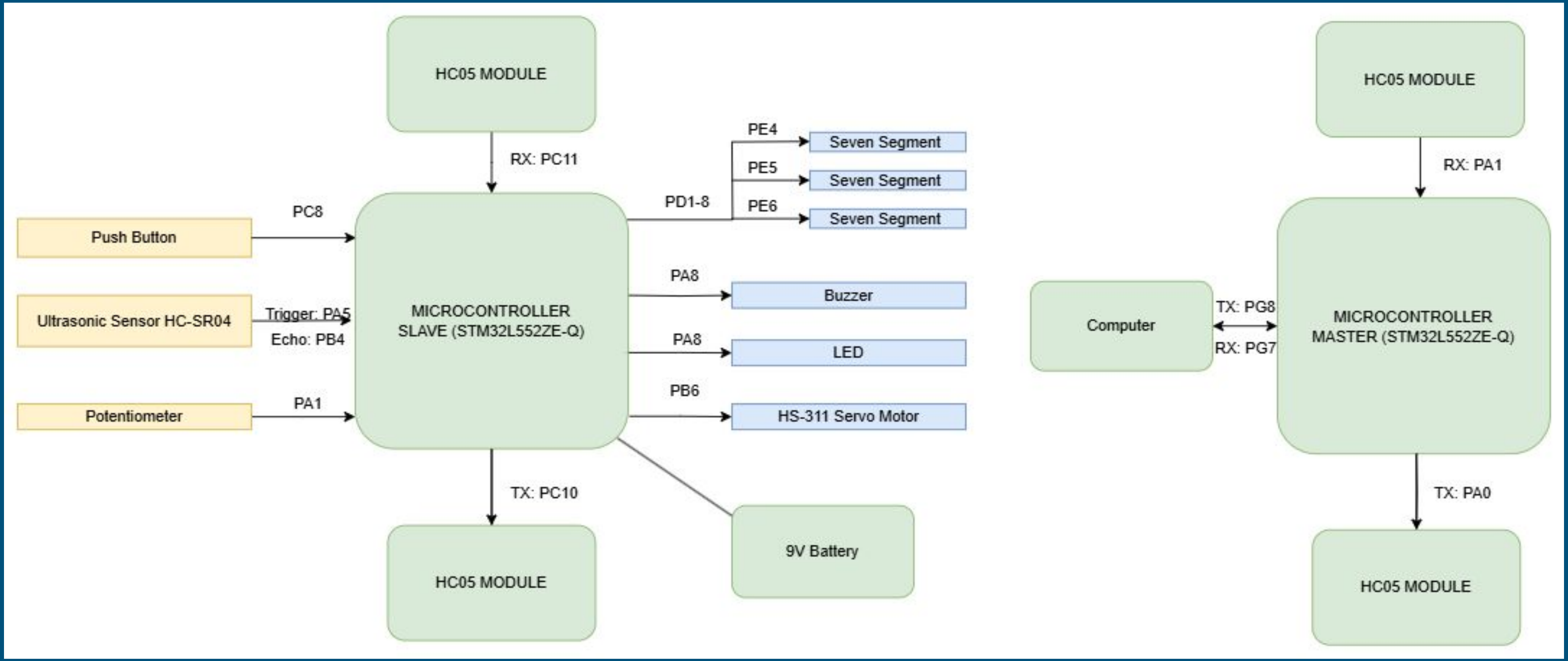
Photo of the Final Product



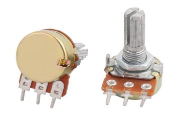
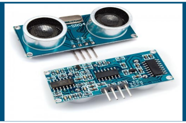
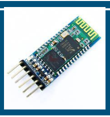
System-level Block Diagram



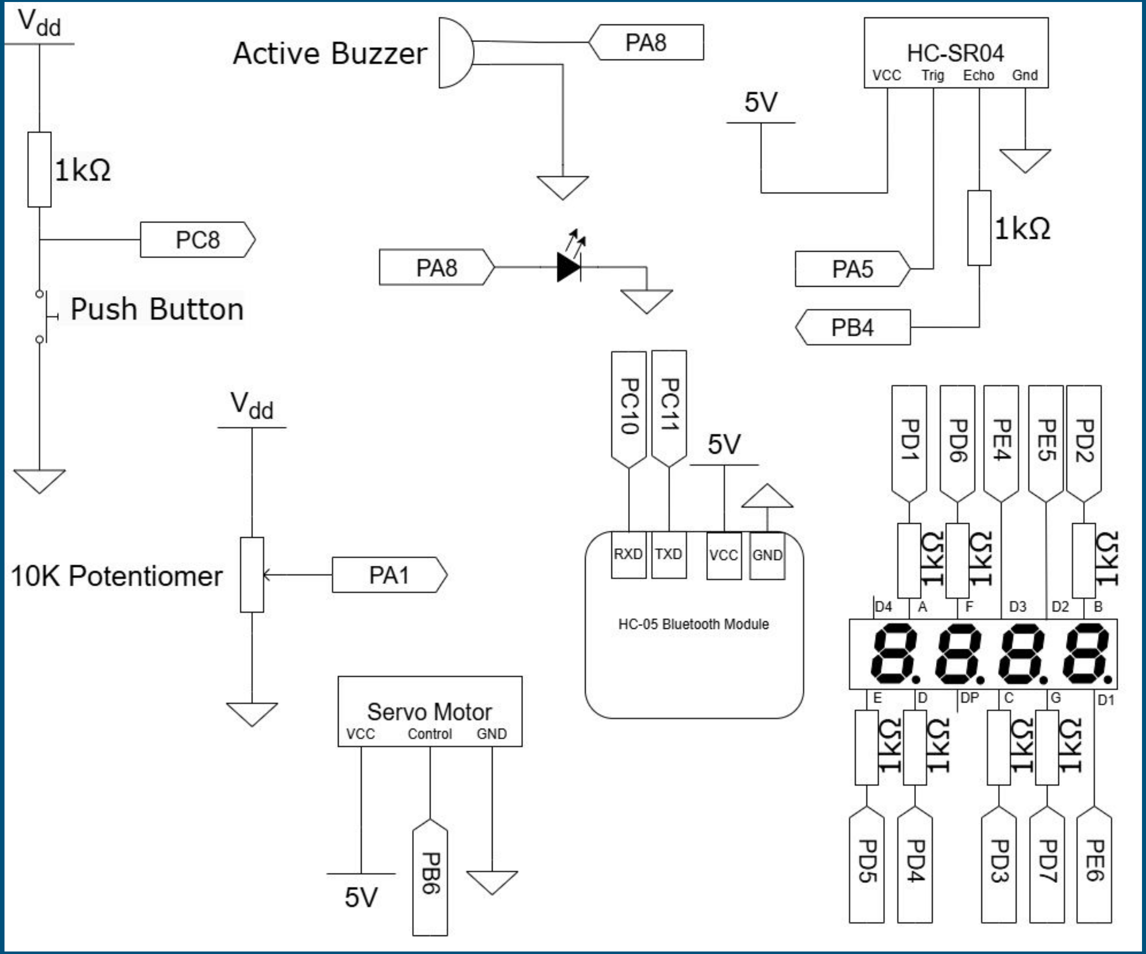
Functional Block Diagram



Components

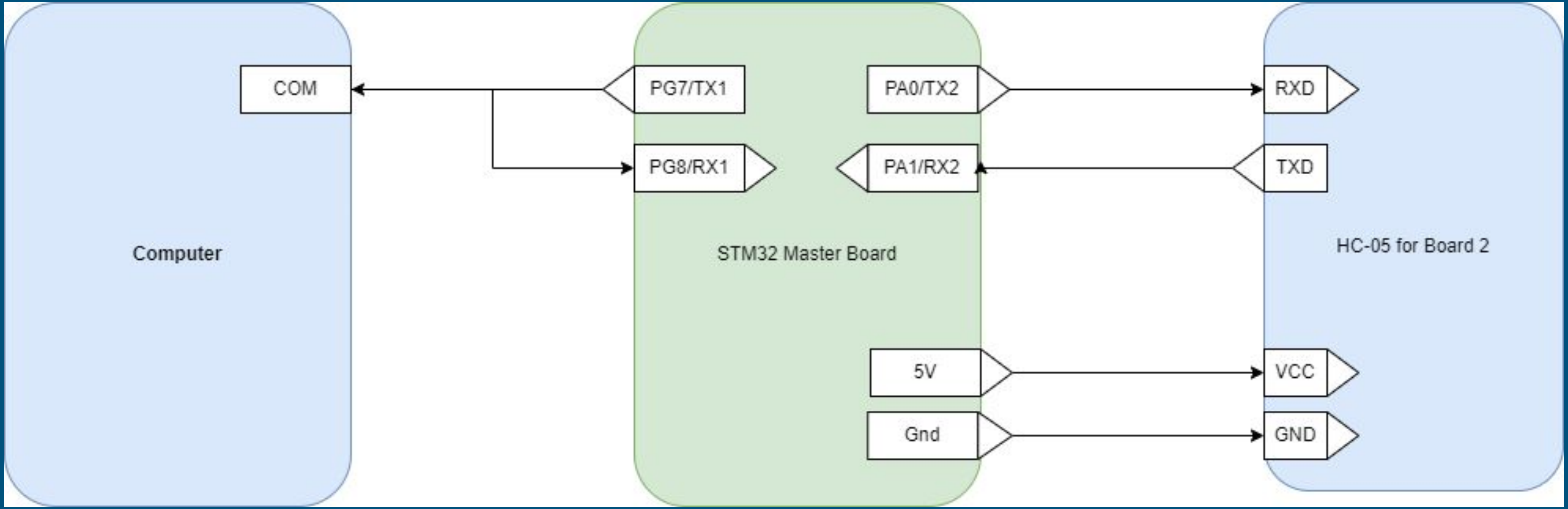
Components Category	Device	Picture	Use in the PI
Input Device	Potentiometer (10K Ohm)		Get timeout interval value between 0.5 and 5 sec for buzzer and led
Input Device	Ultrasonic Sensor HC-SR04		Measure distance and detect pill box if distance is less than threshold
Input Device	Push Button		Start and reset the flow
Output Device	3 Digit Seven Segment Display		Display the amount of minutes left for next pill box drop
Output Device	Led		Visually notify user that pill is dropped
Output Device	Buzzer		Notify user that pill is dropped by generating sound
Output Device	HS-311 Servo Motor		Perform a turn to drop the pill box with pills that user should take in next time slot
Communication Device	HC-05 Bt module		Communicate with each other by Bluetooth and boards they're connected by UART

Schematic Diagram - Slave

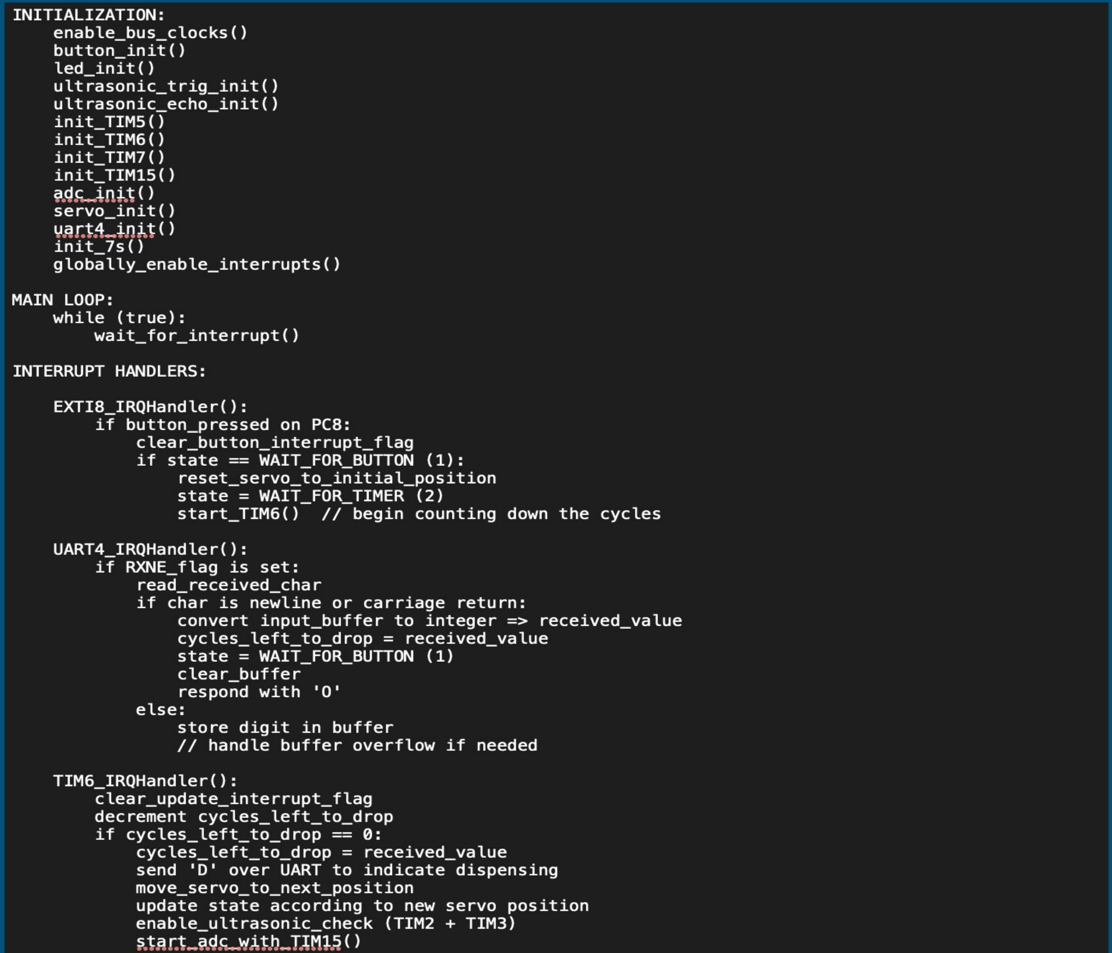


Digit order: D4-D3-D2-D1

Schematic Diagram - Master

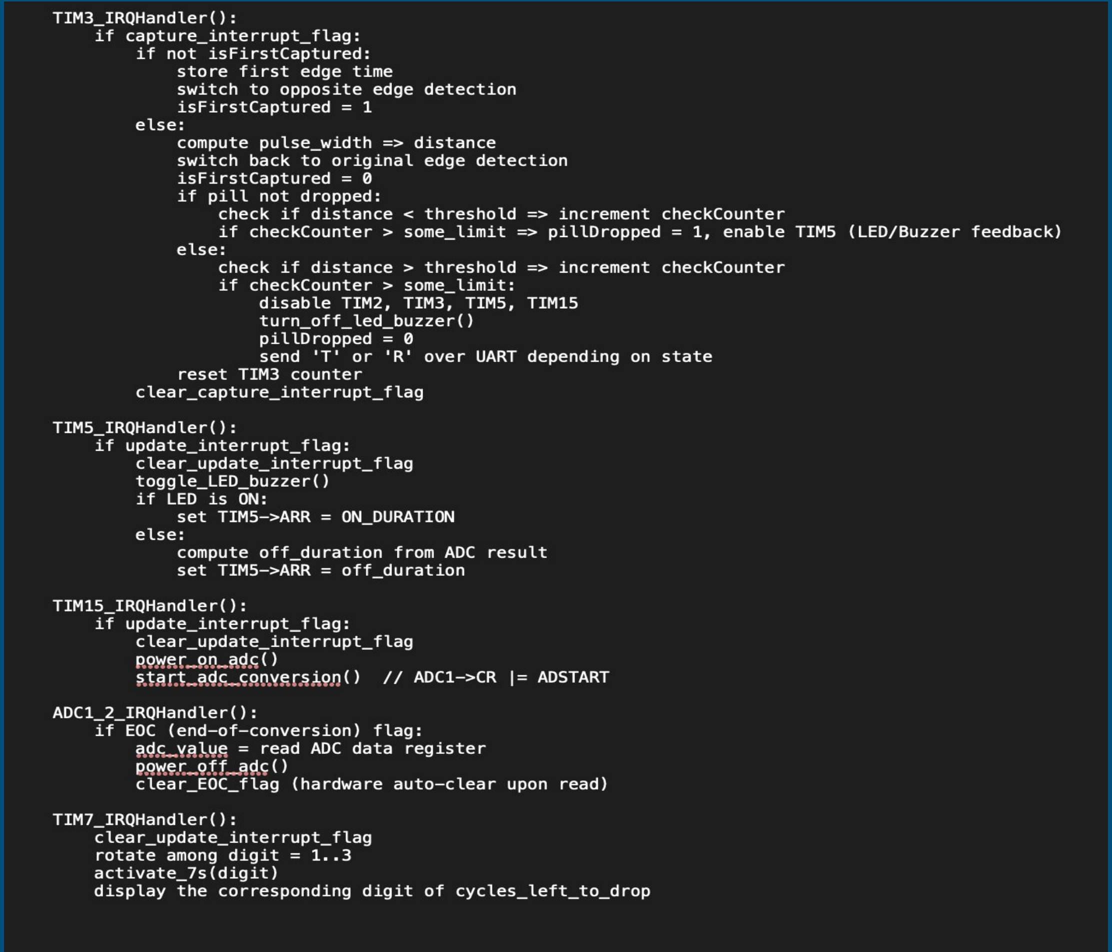


Pseudocode



```
INITIALIZATION:
    enable_bus_clocks()
    button_init()
    led_init()
    ultrasonic_trig_init()
    ultrasonic_echo_init()
    init_TIM5()
    init_TIM6()
    init_TIM7()
    init_TIM15()
    adc_init()
    servo_init()
    uart4_init()
    init_7s()
    globally_enable_interrupts()

MAIN LOOP:
    while (true):
        wait_for_interrupt()

INTERRUPT HANDLERS:

    EXTI8_IRQHandler():
        if button_pressed on PC8:
            clear_button_interrupt_flag
            if state == WAIT_FOR_BUTTON (1):
                reset_servo_to_initial_position
                state = WAIT_FOR_TIMER (2)
                start_TIM6()  // begin counting down the cycles

    UART4_IRQHandler():
        if RXNE_flag is set:
            read_received_char
            if char is newline or carriage return:
                convert input_buffer to integer => received_value
                cycles_left_to_drop = received_value
                state = WAIT_FOR_BUTTON (1)
                clear_buffer
                respond with '0'
            else:
                store digit in buffer
                // handle buffer overflow if needed

    TIM6_IRQHandler():
        clear_update_interrupt_flag
        decrement cycles_left_to_drop
        if cycles_left_to_drop == 0:
            cycles_left_to_drop = received_value
            send 'D' over UART to indicate dispensing
            move_servo_to_next_position
            update state according to new servo position
            enable_ultrasonic_check (TIM2 + TIM3)
            start_adc_with_TIM15()
```

Pseudocode Continued



```
TIM3_IRQHandler():
    if capture_interrupt_flag:
        if not isFirstCaptured:
            store first edge time
            switch to opposite edge detection
            isFirstCaptured = 1
        else:
            compute pulse_width => distance
            switch back to original edge detection
            isFirstCaptured = 0
            if pill not dropped:
                check if distance < threshold => increment checkCounter
                if checkCounter > some_limit => pillDropped = 1, enable TIM5 (LED/Buzzer feedback)
            else:
                check if distance > threshold => increment checkCounter
                if checkCounter > some_limit:
                    disable TIM2, TIM3, TIM5, TIM15
                    turn_off_led_buzzer()
                    pillDropped = 0
                    send 'T' or 'R' over UART depending on state
            reset TIM3 counter
        clear_capture_interrupt_flag

TIM5_IRQHandler():
    if update_interrupt_flag:
        clear_update_interrupt_flag
        toggle_LED_buzzer()
        if LED is ON:
            set TIM5->ARR = ON_DURATION
        else:
            compute off_duration from ADC result
            set TIM5->ARR = off_duration

TIM15_IRQHandler():
    if update_interrupt_flag:
        clear_update_interrupt_flag
        power_on_adc()
        start_adc_conversion()  // ADC1->CR |= ADSTART

ADC1_2_IRQHandler():
    if EOC (end-of-conversion) flag:
        adc_value = read ADC data register
        power_off_adc()
        clear_EOC_flag (hardware auto-clear upon read)

TIM7_IRQHandler():
    clear_update_interrupt_flag
    rotate among digit = 1..3
    activate_7s(digit)
    display the corresponding digit of cycles_left_to_drop
```

Bill of Materials (BOM)

Name	Unit Price (TL)	Amount	Total Price (TL)	Source
HS-311 Servo Motor	597.87	1	597.87	Previously owned
4 Digit 7 Segment Display	33.6	1	33.6	Previously owned
Push Button	2.09	1	2.09	Previously owned
Ultrasonic Sensor HC-SR04	35.94	1	35.94	Borrowed
RGB Led	5.07	1	5.07	Borrowed
10K Potentiometer	5.07	1	5.07	Borrowed
HC05 Serial Port Bluetooth Module	182.72	2	365.44	Borrowed
9V Battery Connector	11.66	1	11.66	Previously owned
Buzzer	14.60	1	14.60	Previously owned
Total Cost			1051.34	

Integration Task Allocation

PIs	Components	Name of the Student	Which Task?
GPIO - Timer	Buzzer - Led	Eren	Controlling buzzer and led in specific intervals decided from ADC. Power saving improvements. Implementing Reset functionality.
GPIO - Timer	Seven Segment	Arda	Display time left to dispense the pill. Handling the data sent from the UART using interrupt. Implementing Start button.
IC/OC	Motor	Ramazan Onur	Turning the motor 45 degrees and returning to the base point based on the machine state.
IC/OC	Ultrasonic Sensor	Onur Çerli	Detecting the pill box by ultrasonic sensor using IC/OC.
Timer - ADC	Potentiometer	Deniz Bilge Akkoç	Reading the potentiometer value in order to change off duration between 0.5 and 5 seconds.
UART(Tx/Rx)	Bluetooth	Mustafa Atak	Tx: Sending the pill dispense time by getting the input from user. Rx: Displaying the log of the dispenser to the terminal for U/I.

Self Evaluation of Integration Requirements

Requirements	Achieved?	Explain
Prototype looks like an ES	Yes	No visible electronics other than the ones users interact with.
Full-interrupt driven	Yes	It is fully interrupt driven.
Complex GPIO Component	Yes	Parallel seven segments driven with timer interrupts.
“Meaningful” UI	Yes	Concise logs shown to the terminal.
Floating Point Operations	No	No floating point operation is done.
Power efficient code	Yes	ADC Power Saving is especially implemented. WFI instruction is run in main function.
Bonus Action	Yes	- The ES runs on a 9V battery. - EXTI peripheral is used with button input.

Integration

We used 2 nucleo boards. Multiple breadboards are connected to the slave board. Female connectors were used to adjust the positioning of the components in the prototype.

All integration tasks cover the functionalities that an automated pill dispenser require.

All team members took the main responsibility implementing their assigned PI. Additionally, each member contributed to the preparation of at least one other PI.

Some pins were used by multiple PI codes therefore we changed some of them. Debugging as a whole was challenging therefore we debugged the code state by state.

References

https://community.st.com/

https://app.diagrams.net/