



COMSATS University Islamabad (CUI)

Microprocessor and Assembly Language

Project Report

for

Money Exchange Bot

By

Daoud Hussain

CUI/SP21-BCS-102/ISB

Submitted to:

Mr. Taimur Shahzad

Bachelor of Science in Computer Science (2021–2025)

Table of Contents

1. Introduction	2
2. Description	3
3. Core Functionality	4
4. Limitation and consideration.....	6
5. Motivation	7
6. Procedures	8
7. Complete Code.....	10

1. Introduction

Money Exchange Bot is designed to effortlessly convert Pakistani Rupees to various international currencies. Utilizing assembly language, it offers a user-friendly interface where individuals can seamlessly convert their desired amount from PKR to US Dollars, Indian Rupees, Euros, United Arab Emirates Dirhams, Pound Sterlings, and Kuwaiti Dinars. The program prompts users with a clear menu of conversion options, allowing them to select their preferred currency. Upon inputting the amount, it promptly calculates and displays the converted value using specific exchange rates.

This efficient system ensures ease of use, providing accurate conversions based on real-time exchange rates. It's an invaluable tool for anyone needing quick and precise currency conversions in a PKR-centric context.

2. Description

The "Money Exchange Bot" project is an efficient Currency Conversion System developed to streamline the process of converting Pakistani Rupees (PKR) to diverse international currencies. Crafted using assembly language, this system offers a user-friendly interface allowing seamless conversions to US Dollars, Indian Rupees, Euros, United Arab Emirates Dirhams, Pound Sterlings, and Kuwaiti Dinars. The program boasts an interactive menu presenting users with conversion options, coupled with precise and up-to-date exchange rates for each currency. Users input their desired conversion amount, triggering swift calculations and accurate displays of the converted value. Employing loops, the system enables multiple conversions within the same session, ensuring efficiency and convenience. Whether for individuals, businesses, or financial entities, this Currency Conversion System stands as a reliable and accessible tool, simplifying currency exchanges with precision and ease.

3. Core Functionality

The core functionality of the "Money Exchange Bot" project revolves around facilitating seamless currency conversion from Pakistani Rupees (PKR) to various international currencies. Key aspects of its functionality include:

- 1. User Interaction:** The system engages users through a user-friendly interface, presenting a menu of currency conversion options. It prompts users to select the desired currency for conversion.
- 2. Conversion Calculation:** Upon user input of the desired amount in PKR, the program utilizes predefined and accurate exchange rates for each selected currency. It swiftly calculates the converted amount based on the input and the designated exchange rate.
- 3. Precision and Accuracy:** Ensuring accuracy is paramount. The system employs precise exchange rates, ensuring that the converted values are reflective of real-time market rates, guaranteeing reliability in conversions.
- 4. Looping Functionality:** Users have the flexibility to perform multiple conversions within a single session without restarting the program. This looping functionality enhances user convenience and efficiency, allowing for sequential or varied currency conversions in one go.
- 5. Clear Output Display:** After each conversion, the system displays the converted amount clearly, providing users with immediate and accurate results for their financial needs.
- 6. Ease of Use:** The system's design prioritizes simplicity, ensuring that users of varying technical backgrounds can effortlessly navigate the interface, select their desired currency, and obtain precise conversion values without complexity.
- 7. Prompted Interaction:** The system prompts users at each step, guiding them through the process, from currency selection to inputting amounts and viewing the converted values. This interactive approach enhances user experience and ensures a smooth conversion process.

These core functionalities collectively create an efficient, user-centric system that simplifies currency conversion, catering to diverse financial needs while ensuring accuracy, reliability, and ease of use.

4. Project Description

This project is designed in assembly language using Irvine32. The project starts by showing the user a message that welcomes the user by printing on the command line, "Welcome to Currency Conversion". Then the user is shown all the possible conversions it can perform. The prompts are stored in bytes and are called when the offset is moved into edx and then the built-in function WriteString prints the message on the screen. The edx register is often used to store data such as memory addresses, loop counters, and intermediate results of calculations. A loop is started so that the user can perform multiple conversions if the user wants to. Then the user selects the conversion they want to do. After that, the user enters the amount of money that they want to convert. There are multiple procedures in this program, there is a main procedure, then there is a display menu procedure which displays the user options. Then for each conversion there is a different procedure. In each procedure, the user is asked the amount of money they want to convert. Then that number is multiplied with the specific rate for that conversion. And the following converted value is displayed.

5. Limitations and Considerations

Here are some considerations and limitations of the "Money Exchange Bot" project:

i. Considerations:

- **Exchange Rate Fluctuations:** The system utilizes predefined exchange rates, but real-time currency values fluctuate in the global market. Users should be aware that the displayed conversion rates might not always align with current market rates due to fluctuations.
- **Accuracy Dependent on Rates:** The accuracy of conversions heavily relies on the accuracy of the pre-defined exchange rates. Updates to these rates might not be reflected immediately in the system, potentially leading to slight discrepancies in conversions.
- **Single-Directional Conversion:** The system supports conversions from PKR to other currencies but does not offer the functionality to convert back from those currencies to PKR within the same session.
- **Limited Currency Selection:** While offering a variety of currencies for conversion, the system might not include every global currency, limiting its versatility for some specific currency exchanges.

ii. Limitations:

- **Real-Time Data:** The system does not fetch real-time data from live market sources. It relies on pre-set exchange rates, which might not accurately reflect current market conditions.
- **No Decimal Precision Handling:** The system might not handle decimal precision in calculations with absolute accuracy due to the limitations of floating-point arithmetic used in assembly language.
- **User Input Validation:** There might be limitations in handling incorrect user inputs or invalid characters during amount input, potentially leading to errors or unexpected behaviors.
- **Solely PKR-based Conversion:** The system focuses solely on converting from PKR to other currencies and doesn't support other currency-to-currency conversions.
- **Limited Error Handling:** Error handling capabilities within the system might be limited, potentially leading to abrupt terminations or unclear error messages in case of unexpected inputs or issues.

These limitations and considerations highlight areas where the system might have constraints or areas where users should exercise caution or be aware of potential discrepancies in currency conversions.

6. Motivation

The motivation behind the "Money Exchange Bot" project stemmed from a fundamental need for a simple yet effective tool that addresses the challenges individuals and businesses face when dealing with currency conversions involving Pakistani Rupees (PKR) and various global currencies. The primary motivations include:

- **Accessibility and Convenience:** There was a need for a user-friendly platform that simplifies the process of converting PKR into other prominent currencies. This project aims to provide an easily accessible tool that anyone, regardless of technical expertise, can use effortlessly.
- **Real-Time Currency Needs:** Recognizing the everyday necessity of individuals and businesses requiring quick and accurate currency conversions, the project aimed to offer a solution that provides rapid results, catering to immediate currency exchange requirements.
- **Financial Decision Making:** Enabling users to swiftly understand the equivalent value of their PKR in other currencies aids in making informed financial decisions, whether it's for personal travel, business transactions, or investment purposes.
- **Education and Understanding:** By offering a platform that demonstrates currency conversions in a straightforward manner, the project aims to educate

users about exchange rates and how different currencies compare against PKR, fostering financial literacy.

- **Simplicity and Reliability:** The motivation was to create a system that, while simple to use, also maintains reliability in its conversions. By providing predefined exchange rates, the project ensures a quick and reliable estimation of converted values.
- **Addressing User Needs:** Understanding the diverse needs of users involved in international transactions or travel, the project aims to serve as a valuable tool, aiding in converting currencies accurately and efficiently.

Ultimately, the core motivation behind the "Money Exchange Bot" project was to bridge the gap between PKR and other currencies, offering a practical solution that simplifies currency conversions, promotes financial awareness, and meets the immediate needs of users navigating the global financial landscape.

7. Procedures:

From the provided assembly language code for the Money Exchange Bot project, here are the procedures used:

1. DisplayMenu:

- This procedure displays the main menu for currency conversion options.
- It prompts the user to select their desired currency conversion.
- Reads the user's input for the chosen conversion.

2. `conversion1` to `conversion6`:

- These procedures handle the conversion calculations for each selected currency.
- They take the user input (amount in PKR), multiply it by the predefined exchange rate for the respective currency, and display the converted amount.

3. `main`:

- The main procedure orchestrates the entire program flow.
- It displays the welcome message and initiates the loop for performing conversions until the user chooses to terminate.
- Calls `DisplayMenu` to show the currency conversion options.
- Determines the user's choice and calls the respective conversion procedure (`conversion1` to `conversion6`) based on the selection.
- Displays the converted amount and prompts for further conversions until termination.

4. `conv_loop`:

- Controls the loop for continuous conversions until the user decides to terminate the program.
- Contains conditional statements (`cmp` and `je`) to determine the user's choice for conversions or termination.

5. `ending`:

- Displays a thank you message when the user chooses to terminate the program.

6. Other Supporting Procedures:

- `WriteString`, `CRLF`, `WriteFloat`, `ReadInt`, `ReadFloat`, and other procedures used are likely to be part of the library functions or defined elsewhere in the code to handle input/output, string operations, and floating-point calculations.

These procedures collectively facilitate user interaction, handle currency conversion calculations, manage the main program flow, and provide essential functionalities for input/output operations.

8. Code:

```
INCLUDE Irvine32.inc

.data
input DD ?           ; input value
output DD ?
output2 DD ? ; output value
selection DWORD ?    ; user's selection

rate1 DD 0.0044       ;1 dollar = 228 pkr
rate2 DD 0.36         ;1 indian rupee = 2.76 pkr
rate3 DD 0.0042       ;1 euro = 240 pkr
rate4 DD 0.0162       ;1 UAE Dirham = 62 pkr
rate5 DD 0.0037       ;1 pound sterling = 271 pkr
rate6 DD 0.0013       ;1 Kuwaiti Dinar = 741 pkr

MainPrompt BYTE "===== Welcome to Pak Money
Exchange Bot =====",0
choice BYTE "Your Choice: ", 0
mainMenu1 BYTE "Select a exchange you want to do: ", 0DH,0AH,

menu1 BYTE "1. Convert from Pakistani Rupee to US Dollar: ", 0DH,0AH,
          "2. Convert from Pakistani Rupee to Indian Rupee: ", 0DH,0AH,
          "3. Convert from Pakistani Rupee to Euro: ", 0DH,0AH,
          "4. Convert from Pakistani Rupee to UAE Dirham: ", 0DH,0AH,
          "5. Convert from Pakistani Rupee to Pound Sterling: ", 0DH,0AH,
          "6. Convert from Pakistani Rupee to Kuwaiti Dinar: ", 0DH,0AH,

prompt BYTE "Enter the amount of money you want to exchange: ",0

ending BYTE "Thank you for using this conversion system!",0

result BYTE "Converted Amount is: ", 0

.code
main PROC
```



```

mov edx, offset MainPrompt
call WriteString
call CRLF
call CRLF
; Set up a loop to allow the user to perform multiple conversions
mov ecx, -1
cmp ecx, 0

```

conv_loop:

```

; Display the conversion menu and get the user's selection
call DisplayMenu
call CRLF

```

```

; Calculate the output value based on the user's selection
cmp eax, 1
je conv1
cmp eax, 2
je conv2
cmp eax, 3
je conv3
cmp eax, 4
je conv4
cmp eax, 5
je conv5
cmp eax, 6
je conv6
cmp eax, 7
je terminate

```

```

conv1:
    mov edx, offset prompt
    call CRLF
    call WriteString
    call ReadFloat
    fstp input
    call conversion1
    jmp done

```

```

conv2:
    mov edx, offset prompt
    call CRLF
    call WriteString
    call ReadFloat
    fstp input
    call conversion2
    jmp done

```

```

conv3:
    mov edx, offset prompt
    call CRLF
    call WriteString
    call ReadFloat
    fstp input
    call conversion3
    jmp done

```

```

conv4:
    mov edx, offset prompt
    call CRLF
    call WriteString
    call ReadFloat
    fstp input
    call conversion4
    jmp done

conv5:
    mov edx, offset prompt
    call CRLF
    call WriteString
    call ReadFloat
    fstp input
    call conversion5
    jmp done

conv6:
    mov edx, offset prompt
    call CRLF
    call WriteString
    call ReadFloat
    fstp input
    call conversion6
    jmp done

terminate:
    mov edx, offset ending
    call WriteString
    call CRLF
    ret

done:
    ; Display the output value
    fld output
    call WriteFloat
    call CRLF
    call CRLF
    jmp conv_loop

```

```

DisplayMenu PROC
    ; Display the conversion menu
    ; Set text color to default (black)
    mov eax, white
    call SetTextColor

    mov edx, offset mainMenu1
    call WriteString
    call CRLF

    ; Get the user's selection
    call CRLF
    mov edx, offset choice
    call WriteString

    call ReadInt

```

```

    ret
DisplayMenu ENDP

conversion1 PROC
    fld input
    fld rate1
    FMul
    call CRLF
    mov edx, offset result
    call WriteString
    fstp output
    ret
conversion1 ENDP

conversion2 PROC
    fld input
    fld rate2
    FMul
    call CRLF
    mov edx, offset result
    call WriteString
    fstp output
    ret
conversion2 ENDP

conversion3 PROC
    fld input
    fld rate3
    FMul
    call CRLF
    mov edx, offset result
    call WriteString
    fstp output
    ret
conversion3 ENDP

conversion4 PROC
    fld input
    fld rate4
    FMul
    call CRLF
    mov edx, offset result
    call WriteString
    fstp output
    ret
conversion4 ENDP

conversion5 PROC
    fld input
    fld rate5
    FMul
    call CRLF
    mov edx, offset result
    call WriteString
    fstp output
    ret
conversion5 ENDP

```

```
conversion6 PROC
    fld input
    fld rate6
    FMul
    call CRLF
    mov edx, offset result
    call WriteString
    fstp output
    ret
conversion6 ENDP

main ENDP
END main
```