A close up of a word

Description automatically generated

EMBEDDED SYSTEMS   
*MINI PROJECT REPORT*

***TITLE*: STOPWATCH USING A KEYBOARD INPUT, A TIMER AND A LCD DISPLAY.**

*Submitted To*

**Department of Computer Science & Engineering**

*Submitted By*

**TEAM MEMBERS:**

61 Namrata Raj 220905550

51 Aarohi Jadav 220905472

34 Mrunal Kulkarni 220905243

49 Prakhar Arya 220905458

15 Venkatesh Gaur 220905089

Name & Signature of Evaluator

**Introduction to LPC1768 Stopwatch Project**

The LPC1768 Stopwatch Project uses the LPC1768 microcontroller from NXP Semiconductors to create a digital stopwatch using embedded C programming. This project aims to build a stopwatch that can accurately measure time using the microcontroller's built-in timers, without needing extra components.

The project will involve:

Setting up and using the LPC1768's timers to control the stopwatch functions like start, stop, and reset.

Writing code in embedded C to handle timer events and user inputs.

Creating a simple interface, an LCD to interact with the stopwatch.

The goal is to build a working stopwatch that showcases the capabilities of the LPC1768 microcontroller for real-time applications, while also providing practical experience in microcontroller programming and firmware development. This project is ideal for learning about embedded systems and timer-based applications.

**APPARATUS**

Hardware Setup:

-NXP LPC1768 DEVELOPMENT BOARD

ALS-SDA-ARMCTXM3-01

Power supply (+5V)

Cross cable for programming and serial communication

10 core FRC cables

USB to B type

Connections

CND to CNAD (P0.23-P0.28)

CNB to CNB3 (P2.10-P2.13)

SOFTWARE:

-KEIL 4

- FLASH MAGIC

**Main Function:**

The main function serves as the entry point of the program.

SystemInit() - Essential initializations required for the microcontroller to operate correctly. It enables or disables clocking for various peripherals on the chip, this is called clock gating. The LPC1769 allows placing the interrupt vector table at different memory locations also known as Vector Table relocation.

SystemCoreClockUpdate() - This function calculates and updates the variable SystemCoreClock. This variable holds the current core clock frequency in MHz, which is important for various system functions like delays and timers.

It calls the lcd\_init() and stopwatch function to start the stopwatch functionality.

**Initialization of the system and LCD display:**

It sets up the display parameters and prepares it for use within the program.

**LPC\_PINCON->PINSEL1** refers to a register that controls the pin select function for pins on Port 1.

The bitwise & operation with 0xFC003FFF clears the bits corresponding to pins P0.23 to P0.28. This ensures these pins are set as GPIO (General Purpose Input/Output) pins and not assigned to other functionalities.

**LPC\_GPIO0->FIODIR** refers to the data direction register for Port 0.

The bitwise | operation with 0x0F << 23 | 1 << 27 | 1 << 28 sets the corresponding bits for pins P0.23 to P0.28 as output.

The next line calls a function clear\_ports (not shown in the code snippet), which likely sets the output pins to a specific state (likely all low).

The delay(3200) function introduces a delay of 3200 microseconds. This might be required by the LCD to power up and stabilize.

The LCD initialization sequence begins with setting display mode and font size commands, followed by delays to ensure proper execution. Subsequent commands configure cursor movement and turn on the display with invisible cursor through the two parameter function input lcd\_comdata(0x32, 0)

Overall, the lcd\_init function configures the necessary pins, sends initialization commands to the LCD, and prepares it to display messages.

**Stopwatch Functionality**

The stopwatch function is responsible for controlling the stopwatch functionality. It continuously scans for keypresses to determine whether to start, stop, or reset the stopwatch, handling the timing aspect of the stopwatch, ensuring accurate measurement of time.

Within the stopwatch function, there is a loop that continuously scans for keypresses. These keypresses determine the actions to be taken, such as starting, stopping, or resetting the stopwatch.

Once the stopwatch is started, the elapsed time is displayed on the LCD display. This display is continuously updated to reflect the current elapsed time. The stopwatch function is responsible for timing the duration of the stopwatch. It updates the display with the elapsed time, ensuring that users can see the current stopwatch reading in real-time

**LCD control functions**

The lcd\_comdata function is a crucial function in the codebase, as it is responsible for sending initialization data to the LCD display. This function takes two parameters, `temp1` and `type`, where `temp1` contains the data to be sent, and `type` specifies if it is a command or data when type value is 0 it is a command and when 1 it is data.

To send the data properly, the function first separates the higher 4 bits and lower 4 bits of the `temp1` parameter and then sets the data lines accordingly. Following this, it calls the `write` function to actually send the data to the display, including a slight delay to ensure proper communication with the device.

On the other hand, the `write` function is utilized specifically for writing data to the command/data register of the LCD display. It also accepts `temp` and `type` as parameters, where `temp` contains the data to be written and `type` specifies if it is a command or data.

To successfully write the data, the function begins by clearing all the ports and then setting the data lines to the specific value of `temp`. Additionally, based on the `type` parameter, it either sets or clears the RS register select line to indicate if it is a command or data. Following this, the EN (enable) line is set to signal to the LCD display that data is ready to be read, and a delay occurs before clearing the EN line as it is made negative edge enable so from high to low.

**Key scanning**

The scan function plays a crucial role in scanning keypad input. The fourth that is the last row of 4x4 keyboard of the LPC1768 is made enabled and then through this function it is decided that the user has pressed key corresponding to which column . Through reading the FIOPIN register, it can determine if any key on the keypad is pressed. If a key is indeed pressed, the function extracts the necessary bits to identify which key was pressed, storing this information in the “key” variable for further processing. Furthermore, it shifts the value of “tempn” that stores the corresponding row by 10 bits to update the key value with the correct keypad input.

**Delay function:**

In this function the input parameter is milliseconds (unsigned int) and the timer used is TIMER0 is used.

When the function is called, the TCR is first reset, then enabled. After that it waits till the timer counter reaches the desired delay. Then TCR is disabled.

**Conclusion:**

In conclusion, the "Stopwatch Using Keyboard Input, Timer, and LCD Display" project achieves its objectives of creating a user-friendly time-tracking system with precise measurement and clear visual feedback.

By integrating keyboard input, timer functionality, and an LCD display, the system offers intuitive controls for starting, stopping, and resetting the stopwatch, ensuring ease of use. Leveraging embedded C programming techniques, the system achieves reliable performance and versatility, suitable for various applications such as sports timing or industrial processes.

Overall, the project serves as a valuable learning experience in embedded systems development, showcasing the team's skills in hardware-software integration and engineering.

**References:**

Muhammad Ali Mazidi, Sarmad Naimi, Sepehr Naimi, Janice Mazidi - ARM Assembly Language\_ Programming and Architecture (2013)

The Intel Microprocessors by Barry Brey,4th edition.

Embedded\_Systems\_Introduction\_to\_Arm\_Cor.