Logo

Description automatically generated

**UNIVERSITI TUNKU ABDUL RAHMAN**

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY

BACHELOR OF INFORMATION TECHNOLOGY (HONS)

COMPUTER ENGINEERING

**UCCE1064 PROCESSOR AND INTERFACING**

TITLE: Servo Motor Control Module

|  |  |  |
| --- | --- | --- |
| No. | Name | Student ID |
| 1. | Ng Jing Xuan | 23ACB02444 |
| 2. | Ng Yu Heng | 23ACB02307 |
| 3. | Chow Bin Lin | 23ACB01105 |

Contents

[1 General Information and Product Specifications 2](#_Toc198128345)

[1.1 General Information 2](#_Toc198128346)

[1.2 Product Specifications 2](#_Toc198128347)

[2 Hardware and Software Partition 3](#_Toc198128348)

[2.1 Hardware Partition 3](#_Toc198128349)

[2.2 Software Partition 5](#_Toc198128350)

[3 Algorithm 6](#_Toc198128351)

[3.1 Declaration 6](#_Toc198128352)

[3.2 Pseudocode 6](#_Toc198128353)

[4 Screenshots of final product 11](#_Toc198128354)

[5 References 12](#_Toc198128355)

1 General Information and Product Specifications

* 1. General Information

A servo motor is a type of motor which consists of a control circuit that provides feedback on the current position of the motor shaft. This feedback allows the servo motor to rotate an object with great precision at some specific angles (Apoorve,2025). Thus, a Servo Motor Control Module is designed to control a servo motor by using 8051 microcontroller-based development board.

The design in this assignment allows user to choose multiple speed modes for the servo motor through a keypad interface. After selected a speed mode, user can choose confirm to start the servo motor or choose to reset through the same keypad to select another speed mode. When the servo motor is running, LCD will display the current speed mode while two 7-segment display will show the number of complete servo cycle as real-time feedback. The servo motor will move forward and backward with predefined delay times according to the selected speed mode. After the servo motor has completed 10 cycles, user can choose to start again with the same speed mode or reset speed mode. There are also push buttons for emergency stop and reset function to interrupt the running of servo motor.

* 1. Product Specifications

The following are the functionalities provided by the product.

* LCD Menu Interface: Displays user prompts and current speed mode selection.
* Speed Mode Selection: Four predefined modes to control movement speed of servo motor
* Speed Modes Specification:

Speed Mode Sequence Wait Time (Seconds)

0 Backward → Wait → Forward → Wait 3s

1 Backward → Wait → Forward → Wait 2.5s

2 Backward → Wait → Forward → Wait 2s

3 Backward → Wait → Forward → Wait 1s

* Cycle Counter: Counts and displays how many cycles the servo motor has completed (maximum 10 counts).
* Emergency Stop button: Immediately stops the motor and returns to the main menu.
* Reset button: Returns to main menu but allows the servo motor to continue running.
* Confirm / Start button: To confirm the speed mode selection or to start the servo motor again after finish.

2 Hardware and Software Partition

2.1 Hardware Partition

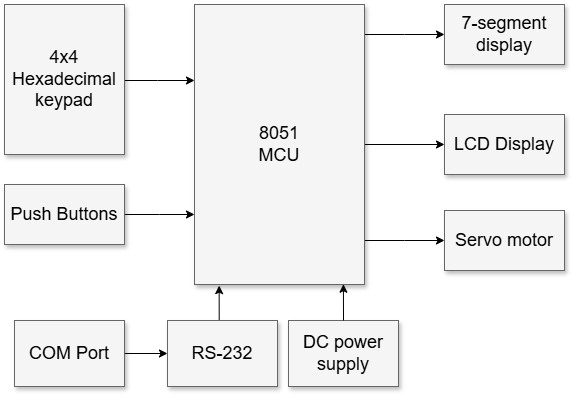


Figure 2.1.1 Block diagram of Servo Motor Control Module

|  |  |  |
| --- | --- | --- |
| **No.** | **Block** | **Description** |
| 1 | 8051 MCU  (89V51RD2 Microcontroller Unit) | * Use to interfaces with 4x4 hexadecimal keypad, LCD, servo motor, 7-segment display, and push buttons. * **Connection**: * P0 → LCD (DB0-DB7) * P1.0-P1.5 → 4x4 Hexadecimal Keypad (R1, R2, C1-C4) * P2 → 7-segment display (CN2) * P3.0, P3.1, P3.4 → LCD (EN, R/W, RS) * P3.5 → Servo Motor PWM * P3.6, P3.7 → 7-segment display (CN1 - 0, 1) |
| 2 | 4x4 Hexadecimal keypad | * Used by the user to select the speed mode (Speed 0, 1, 2, 3), confirm or reset. * **Connection**: * R1, R2 → 8051 MCU (P1.0 – P1.1) * C1-C4, 8051 MCU (P1.2 - P1.5) |
| 3 | Push buttons | * Shows the number of completed back-and-forth cycles (up to 10). * **Connection**: * SW1→ 8051 MCU (P3.2) * SW4 → 8051 MCU (P3.3) |
| 4 | 7-segment display | * Shows the number of completed back-and-forth cycles (0 to 10). * **Connection**: * CN2 - 0, 1 → 8051 MCU (P1.6, P1.7) * CN4 → 8051 MCU (P2) |
| 5 | LCD display | * Displays the menu options, current speed mode, prompt for confirmation/reset, and system status. * **Connection**: * DB0-DB7 → 8051 MCU (P0) |
| 6 | Servo motor | * Rotates back (from 0° to 180°) and forth (from 180° to 0°) based on speed mode selected. * **Connection**: * Vcc → 5V DC power * Ground → GND * PWM → P3.5 |
| 7 | COM Port & RS-232 | * Used for uploading the hex code from PC to 8051 * **Connection**: * USB→RS-232→ 8051 MCU (UART) |
| 8 | DC Power supply | * Powers the 8051 board and servo motor. * **Connection**: * Power supply adapter →Power supply voltage regulator |

Table 2.1 Table of Servo Motor Control Module Hardware partition

2.2 Software Partition

|  |  |  |
| --- | --- | --- |
| **No.** | **Module** | **Description** |
| 1 | Main module | * Initializes the system * Displays menu * Waits for user input * Assign selected speed mode base on keypad press * Confirm or reset selected speed mode * Emergency stop and reset the system |
| 2 | LCD Display module | * Display user options * Display speed mode |
| 3 | Keypad Scanning module | * Continuously scans rows/columns * Debounces input * Maps pressed key to corresponding command |
| 4 | Servo Motor Control module | * Generates PWM signal using Timer 0 interrupts * Sets pulse width based on target angle * Call delay base on selected speed after half cycle rotation * Count cycle count for each full cycle |
| 5 | Interrupt module | * Use timer 0 interrupt to handle PWM high and low cycle * Use external interrupt 0 to detect emergency stop button pressed * Use external interrupt 1 to detect reset button pressed |
| 6 | 7-Segment Display module | * Decode cycle count into 7-segment display code * Display both digits on 7-segment display * Multiplexing display 0 and display 1 with proper delay |
| 7 | Emergency stop and reset module | * Wait for push button inputs * Stop the system when emergency stop button pressed * Reset 7-segment display and LCD when reset button is pressed |
| 8 | Delay module | * Generate delay for LCD to display * Use timer 1 to generate delay for servo motor base on different speed mode chosen * Generate delay for 7-segment display multiplexing |

Table 2.2 Table of Servo Motor Control Module Software partition

3 Algorithm

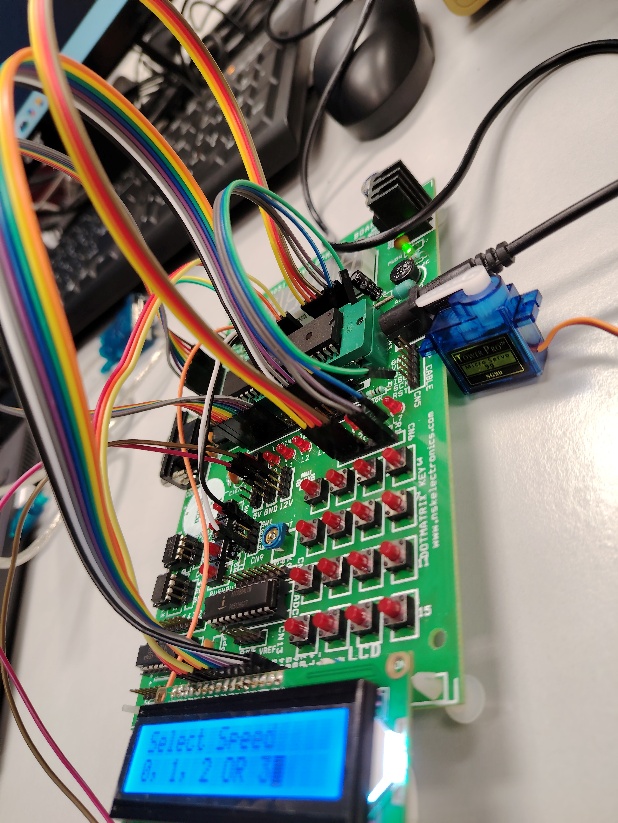
3.1 Declaration

|  |  |  |
| --- | --- | --- |
| Name | Value/ Port Memory address | Function |
| SPEED0 | 60 | Loop count for the servo motor to wait. (60\*0.05s = 3s) |
| SPEED1 | 50 | Loop count for the servo motor to wait. (50\*0.05s = 2.5s) |
| SPEED2 | 40 | Loop count for the servo motor to wait. (40\*0.05s = 2s) |
| SPEED3 | 20 | Loop count for the servo motor to wait. (30\*0.05s = 1s) |
| STOP\_FLAG | 23H.0 | To check if the emergency stop button is pressed |
| RESET\_FLAG | 23H.1 | To check if reset button is pressed |
| MOTOR\_RUNNING | 23H.2 | To check the motor running or not |
| CHOOSE\_MSG | 21H | Choose which message to display |
| DEBOUNCE | 22H | To store the debounce times |
| SPEED\_MODE | 30H | A temporary memory to store the loop count |
| CYCLE\_COUNT | 31H | A temporary memory to store the number of cycles had been ran |
| PWM\_FLAG | 32.1 | Flag to indicate PWM high/ low signal |
| PWM\_PIN | P3.5 | PWM output pin for servo motor |
| SEG\_PORT | P2 | Used for display 7 segment display code |
| DIS0 | P3.6 | To select display 0 |
| DIS1 | P3.7 | To select display 1 |
| RS | P3.4 | Register select, data input or command input |
| RW | P3.1 | decide to write/read to the LCD |
| EN | P3.0 | Enable to activate read or write function to the LCD |

3.2 Pseudocode

|  |
| --- |
| 3.2.1 Main module  START  Locate to RAM location 0000H  Jump to MAIN  // interrupt initialization  Locate to RAM location 0003H // ISR External interrupt 0  Jump to ISR\_INT\_PWM if ISR External interrupt 0 has been triggered  Locate to RAM location 000BH // Timer 0 interrupt  Jump to TIMER0\_INT\_PWM if timer 0 interrupt has been triggered  Locate to RAM location 0013H // ISR 0 External interrupt 1  Jump to ISR\_INT1 if ISR External interrupt 1 has been triggered  Located to RAM location 0050H  MAIN:  enable external interrupt0,1and timer 0 interrupt  Set timer 0 and timer 1 to mode 1  Clear RW to write  Call INITIALIZE\_LCD  Display startup message    START\_AGAIN:  Iteration  Initialize the variables and motor reset to 0 degree    LCD\_RESET:  Call CLEAR\_LCD  Display “Select Speed”  Display “0, 1, 2 OR 3”  WAIT\_MOTOR:  If the motor is running  Jump to CALL\_MOTOR  Call KSCAN1 to wait for user to choose the speed mode  Call CLEAR\_LCD  Display which mode is selected, “Mode X Selected”  Move to the second line of LCD  Display “Confirm or Reset”  Call function KSCAN2 to scan user press confirm or reset button  If the reset button is pressed,  Repeat the iteration  else  Continue to the next step  SERVO\_MOTOR\_AGAIN:  Iteration  Display “Servo Motor”,  Display which mode is running, “Mode X Running..”  Call SERVO  Call CLEAR\_LCD  Display “Running Complete”  Display “Start or Reset?”  Call function GO\_ADDRESS  Call function KSCAN2 to can which button user select start again or reset  If the reset button is pressed  Go to START\_AGAIN  else  Repeat the iteration  CALL\_MOTOR:  Call SERVO  If motor is running  Jump to WAIT\_MOTOR  3.2.2 LCD Display Module  INITIALIZE\_LCD:  Send command 38H (Function Set)  Send command 0FH (Display ON)  Send command 06H (Entry Mode Set)  Return  CLEAR\_LCD:  Send command 01H (Clear display)  Return  WRITE\_STRING:  Set R0 to 0  Iteration  Get character from address in DPTR + R0  If character is zero:  Go to EXIT  Else:  Call WRITE\_CHAR with character  Increment R0  Repeat the iteration  EXIT: Return  GO\_ADDRESS:  Send command C0H (Set DDRAM address to 2nd line)  Return  WRITE\_CMD:  Write A to P0  SET RS to 0  Set EN to 1 then set it to 0  Call LCD\_DELAY  Return  WRITE\_CHAR:  Write A to P0  SET RS to 1  Set EN to 1 then set it to 0  Call LCD\_DELAY  Return  3.2.3 Keypad Scanning Module  KSCAN1:  Wait until keypad first row button is pressed (ROWRD1)  Debounce input  Get which column user is press and set the SPEED\_MODE and CHOOSE\_MSG (COLRD1)  Return  KSCAN2:  Wait until keypad second row button is pressed (ROWRD2)  Debounce input  Get user press confirm or reset (COLRD2)  Call COLRD2  Return  3.2.4 Servo Motor Control Module  SERVO:  Start timer 0  Motor is running  Clear STOP\_FLAG and RESET\_FLAG  If CYCLE\_COUNT is not 0  Go to NEW\_CYCLE  Call SERVO\_DELAY  If STOP\_FLAG is high  Go to EMERGENCY\_STOP  If RESET\_FLAG is high  Go to RESET  NEW\_CYCLE:  Iteration:  Call DIS\_7SEG  ;DEGREE180 (backwards):  Set Timer 0 high duration value: 2.4ms (R5, R6)  Set Timer 0 low duration value: 17.6ms (R3, R4)  Call SERVO\_DELAY  If STOP\_FLAG is high  Go to EMERGENCY\_STOP  If RESET\_FLAG is high  Go to RESET  ;DEGREE0 (forward):  Set Timer 0 high duration value: 0.5ms (R5, R6)  Set Timer 0 low duration value: 19.5ms (R3, R4)  Call SERVO\_DELAY  If STOP\_FLAG is high  Go to EMERGENCY\_STOP  If RESET\_FLAG is high  Go to RESET    Increment CYCLE\_COUNT  If CYCLE\_COUNT <10  Continue the iteration  Motor is not running  RET    3.2.5 Interrupt Module  TIMER0\_INT\_PWM:  If PWM\_FLAG is high  Go to LOW\_CYCLE  HIGH\_CYCLE:  SET PWM\_FLAG to 1  SET PWM\_PIN to 1  Load timer 0 (R5, R6)  Clear TF0  Return from interrupt  LOW\_CYCLE:  SET PWM\_FLAG to 0  SET PWM\_PIN to 0  Load timer 0 (R3, R4)  Clear TF0  Return from interrupt  ISR\_INT0:  Set STOP\_FLAG to 1  Return from interrupt  ISR\_INT1:  Set RESET\_FLAG to 1  Return from interrupt  3.4.6 7-Segment Display Module  DIS\_7SEG:  Convert CYCLE\_COUNT to decimal  Display CYCLE\_COUNT on 7 segment display  Return  3.4.7 Emergency Stop and Reset Module  EMERGENCY\_STOP:  Stop Timer 0  Clear STOP\_FLAG  Clear PWM\_PIN  Clear TF0  Go to START\_AGAIN  RESET:  Clear RESET\_FLAG  Call LCD\_RESET  Go to SERVO  3.4.8 Delay Module  LCD\_DELAY:  Load timer 1(DBF0H)  Activate timer 1  Wait until TF1 = 1 then move to the next step  Stop timer 1 and clear TF1  Return  DELAY\_1s:  Return  SERVO\_DELAY:  Repeat the SPEED\_MODE times begin  Call DIS\_7SEG  Load timer 1(4BFDH)  Start timer 1  Loop begin  Call DIS\_7SEG  End if TF1 = 1  Stop timer 1 and clear TF1  RET |

4 Screenshots of final product

 A green circuit board with many colorful wires

AI-generated content may be incorrect.

Figure 4.1 Connection of ports and wires Figure 4.2 Connection of ports and wires

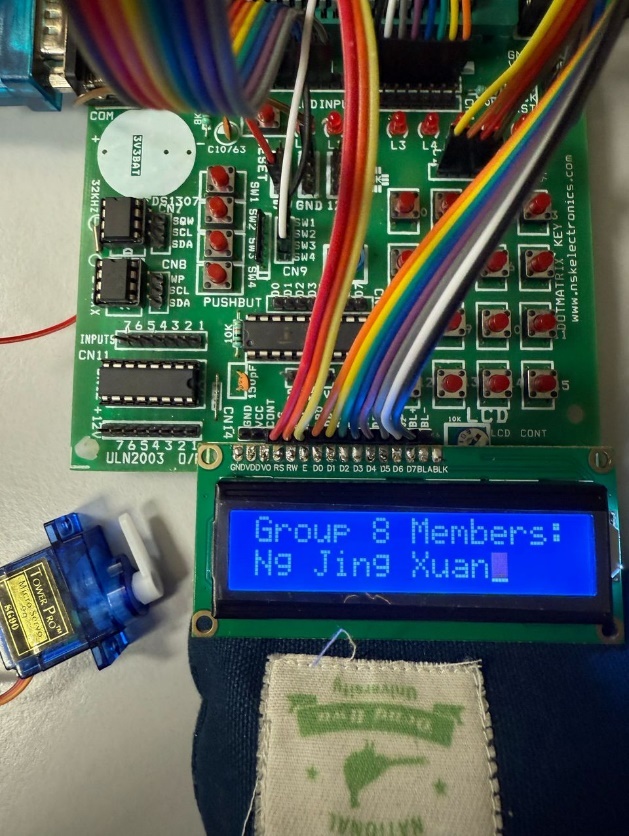
 

Figure 4.3 System startup message Figure 4.4 System startup message (2)

Figure 4.5 Selecting speed message Figure 4.6 Confirmation message

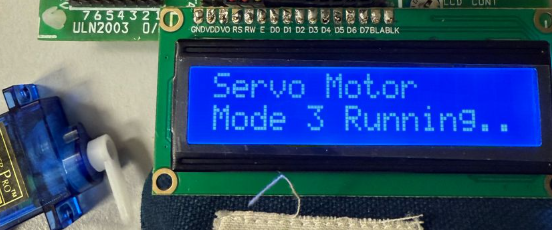


Figure 4.7 Servo motor running message

5 References

Apoorve (2025). *What is a Servo Motor? - Understanding the basics of Servo Motor Working*. [online] Circuit Digest. Available at: https://circuitdigest.com/article/servo-motor-working-and-basics.

‌