KLE Technological University
Creating Value
Leveraging Knowledge

TITLED

DISPENSING MACHINE

BY

Sl.No	Roll.No	Name	USN
1.	402	Manjunath Inamati	01FE21BEC356
2.	405	Gautami Nargund	01FE21BEC177
3.	410	Shrihari Joshi	01FE21BEC184
4.	451	Anusha A D	01FE21BEC231

Under the guidance of

SUPRIYA KATWE

Assistant Professor.

CONTENTS

KLE Technological University
Creating Value
Leveraging Knowledge

Problem Statement

At a shopping mart ,it is required to dispense the products in the bags after billing ,device an approach to implement such a system .

Problem Analysis

- For the dispensing machine ,there would be a display on the LCD regarding the items which are available in the store along with their price ,the user should his/her choice through the UART.
- The input consists of the item along with quantity.
- Specific task is done by the DC motor according to the input:-

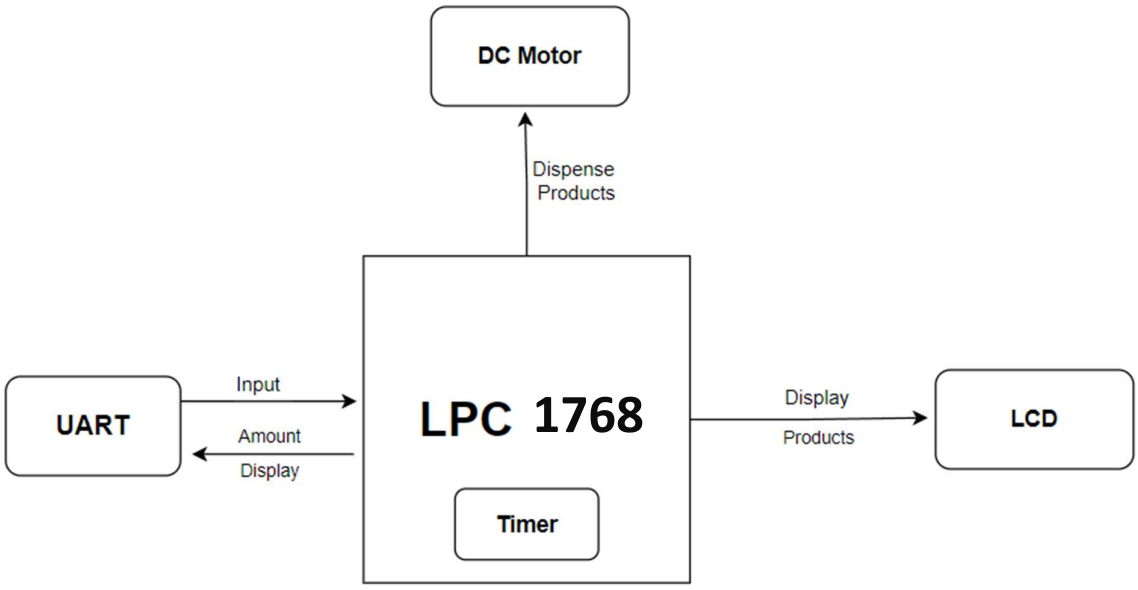
1. For product A, DC motor A is assigned to rotate clockwise for 5 sec.
2. For product B, DC motor B is assigned to rotate clockwise for 5 sec.
3. For product C, DC motor C is assigned to rotate clockwise for 5 sec.

CALCULATIONS

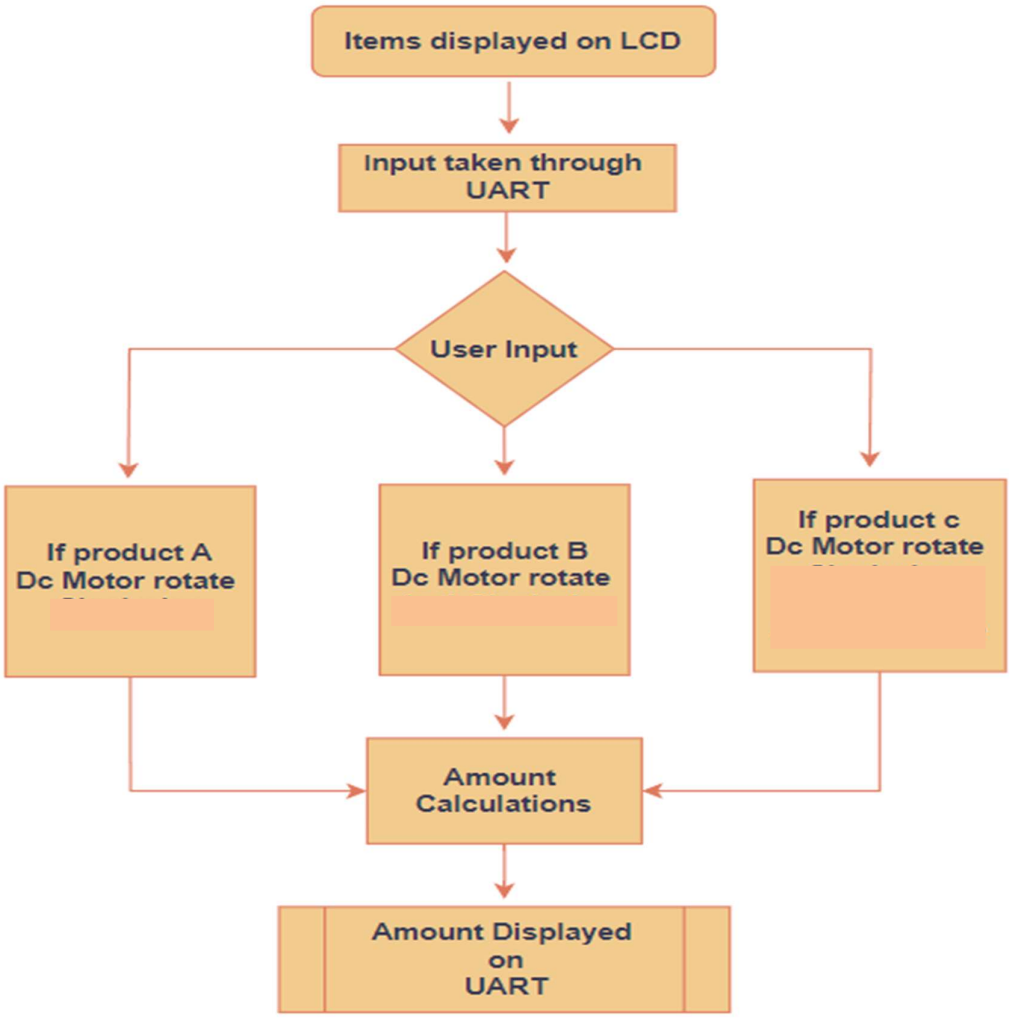
- The total amount is calculated accordingly and displayed on the UART.

Amount = Price * quantity

BLOCK DIAGRAM



Flowchart


Items displayed on LCD
Input taken through UART
User Input
If product A Dc Motor rotate
If product B Dc Motor rotate
If product c Dc Motor rotate
Amount Calculations
Amount Displayed on UART

KLE Technological University
Creating Value
Leveraging Knowledge

Objects

1. Message Queue

   It develops a synchronization between user input and the amount calculated.

ON Chip Peripherals

1. UART0
2. Timer

OFF Chip Peripherals

1. LCD
2. DC Motor

Specifications of peripherals used

1. LCD

- 16x2 display can be used in 8-bit Or 4-bit mode.
- 8 data pins
- 3 control pins (RS, RW, EN)

- RS- Data mode(1) /Command mode(0)
- RW- Read /Write.
- Enable - Latch the data.

2. DC Motor

- Typical working voltage – 3.3V
- Free run current - @3.3V 25mA
- Stall current - @3.3V 650mA

- Functionalities

- Clockwise
- Anti-Clockwise
- Stop

3.UART

- Four UART's
- 16 byte Receive and Transmit FIFOs
- Built-in fractional baud rate generator with auto bauding capabilities
- Software flow control through TXEN bit in Transmit Enable Register
- UART0 Registers

- U0RBR(UART0 Receive Buffer Register)
- U0THR (UART0 Transmit Holding Register)
- U0DLL and U0DLM (UART0 Divisor Latch Registers)
- U0IER (UART0 Interrupt Enable Register)

KLE Technological University
Creating Value
Leveraging Knowledge

Code

```
#include <LPC17xx.h>
#include <rtl.h>
#include "Lcd.h"
#include "uart.h"

unsigned char receive();
void delay_ms(unsigned int ms);
void transmit(unsigned char ch);
void delay(unsigned int x);
void complete(unsigned char mg,unsigned char C);

unsigned char mg,C;
unsigned long int temp1=0, temp2=0 ;
unsigned char Msg1[14] = {"A    B    C"};
unsigned char Msg2[15] = {"100   40   10"};


os_mbx_declare(MsgBox, 1);          // Declare an RTX mailbox
_declare_box(mpool, 20, 1);         // Dynamic memory pool
os_mbx_declare(MsgBox2, 1);         // Declare an RTX mailbox
_declare_box(mpool2, 20, 1);        // Dynamic memory pool
```

```
__task void lcd(void);
__task void dispense(void);
__task void order(void);



__task void order(void)
{

        U32 *rptr,*mptr2;
        os_mbx_init(MsgBox2, sizeof(MsgBox2));    // initialize the mailbox
 mptr2 = _alloc_box(mpool2);             // Allocate memory for the message
        while(1)
        {
                os_dly_wait(5);
         os_mbx_wait(MsgBox, (void **)&rptr, 0xffff); // wait for the
message
         mg=receive();
   transmit(mg);
         C=receive();
   transmit(C);
         _free_box(mpool, rptr); // free memory allocated for message
   mptr2[0]=0;
   os_mbx_send(MsgBox2, mptr2, 0xffff); // Send the message to the
mailbox
        }
```

```
}

__task void dispense(void)
{
        unsigned char success[]="Please enter the name of item and no.of items here---->";
        unsigned char fail[]="No such products available";
 unsigned int i;
        U32 *mptr,*rptr2;


 while(1)
 {
          os_mbx_init(MsgBox, sizeof(MsgBox));    // initialize the mailbox
  mptr = _alloc_box(mpool);             // Allocate memory for the message
  for(i=0;success[i]!='\0';i++)
   {
    transmit(success[i]);
   }
 i=0;
 mptr[0]=i;
 os_mbx_send(MsgBox, mptr, 0xffff); // Send the message to the mailbox
 os_dly_wait(5);
 os_mbx_wait(MsgBox2, (void **)&rptr2, 0xffff); // wait for the message
```

```
if(mg=='A')
{
 LPC_GPIO1->FIOSET = 0x00000200;
 LPC_GPIO1->FIOCLR = 0x00000100;
 delay_ms(50);
 LPC_GPIO1->FIOCLR = 0x00000300;
 complete(mg,C);
}
else if(mg=='B')
{
 LPC_GPIO1->FIOSET = 0x00000100;
 LPC_GPIO1->FIOCLR = 0x00000200;
 delay_ms(50);
 LPC_GPIO1->FIOCLR = 0x00000300;
 complete(mg,C);
}
else if(mg=='C')
{
 LPC_GPIO1->FIOSET = 0x00000200;
 LPC_GPIO1->FIOCLR = 0x00000100;
 delay_ms(50);
 LPC_GPIO1->FIOSET = 0x00000100;
 LPC_GPIO1->FIOCLR = 0x00000200;
 delay_ms(50);
 LPC_GPIO1->FIOCLR = 0x00000300;
```

```
 complete(mg,C);
 }
 else
 {
        transmit(0X0D);
  transmit(0x0A);
       for(i=0;fail[i]!='\0';i++)
   {
     transmit(fail[i]);
   }
                transmit(0X0D);
   transmit(0x0A);
 }
_free_box(mpool2, rptr2); // free memory allocated for message
 }

}

void complete(unsigned char mg,unsigned char C)
{
 int i,k,l=0;
 unsigned char a[5];
 unsigned char cmp[]="Amount---->";

 transmit(0X0D);
```

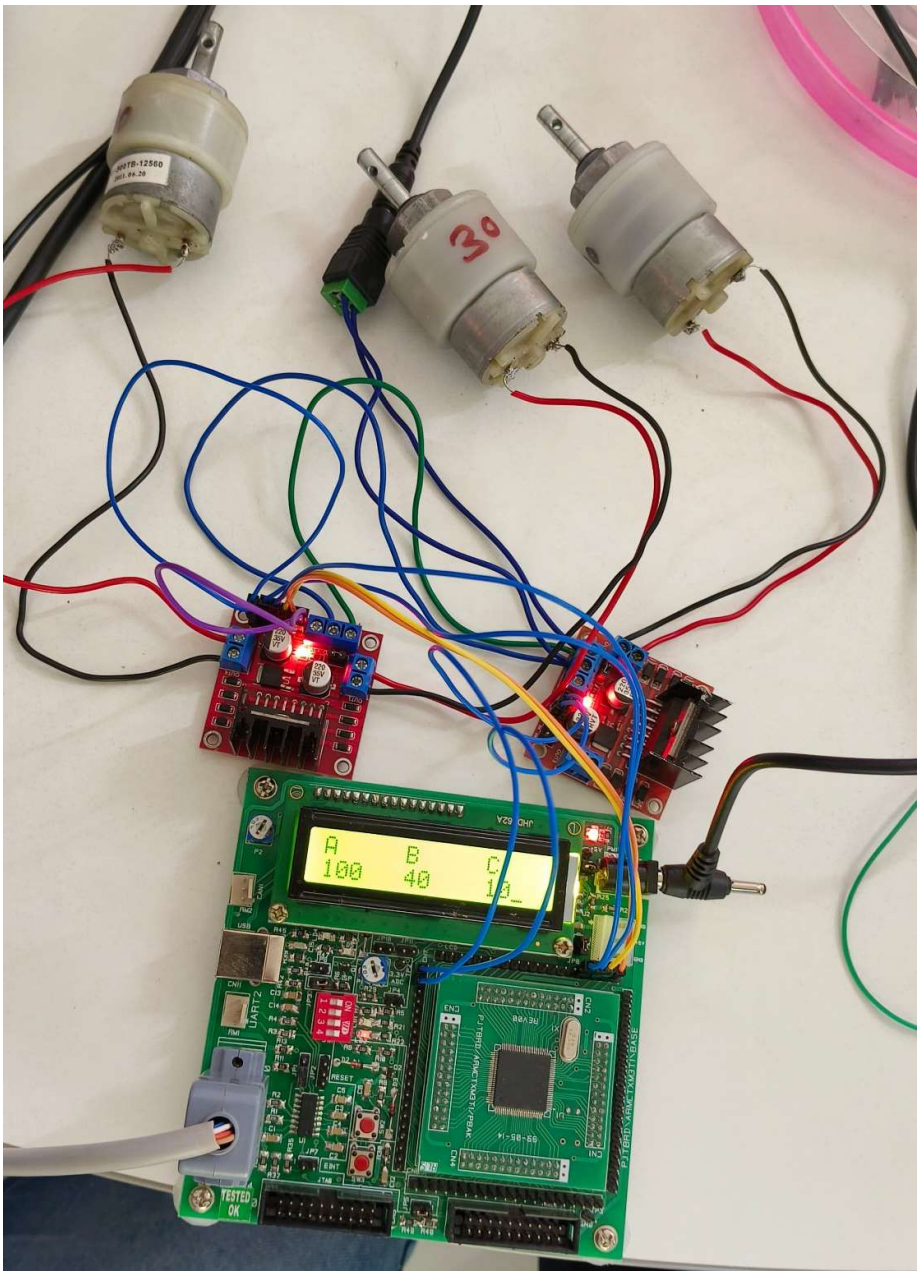
```
 transmit(0x0A);
for(i=0;cmp[i]!='\0';i++)
 {
  transmit(cmp[i]);
 }

k=(C-'0'); // total amount multiplication
if(mg=='A')
k=k*100;
if(mg=='B')
k=k*40;
if(mg=='C')
k=k*10;
i=0;

 while(k)
 {
  a[i]=((k%10)+'0');
  k=k/10;
  i++;
        l++;
 }
 for(i=l-1;i>=0;i--)
 {
   transmit(a[i]);
```

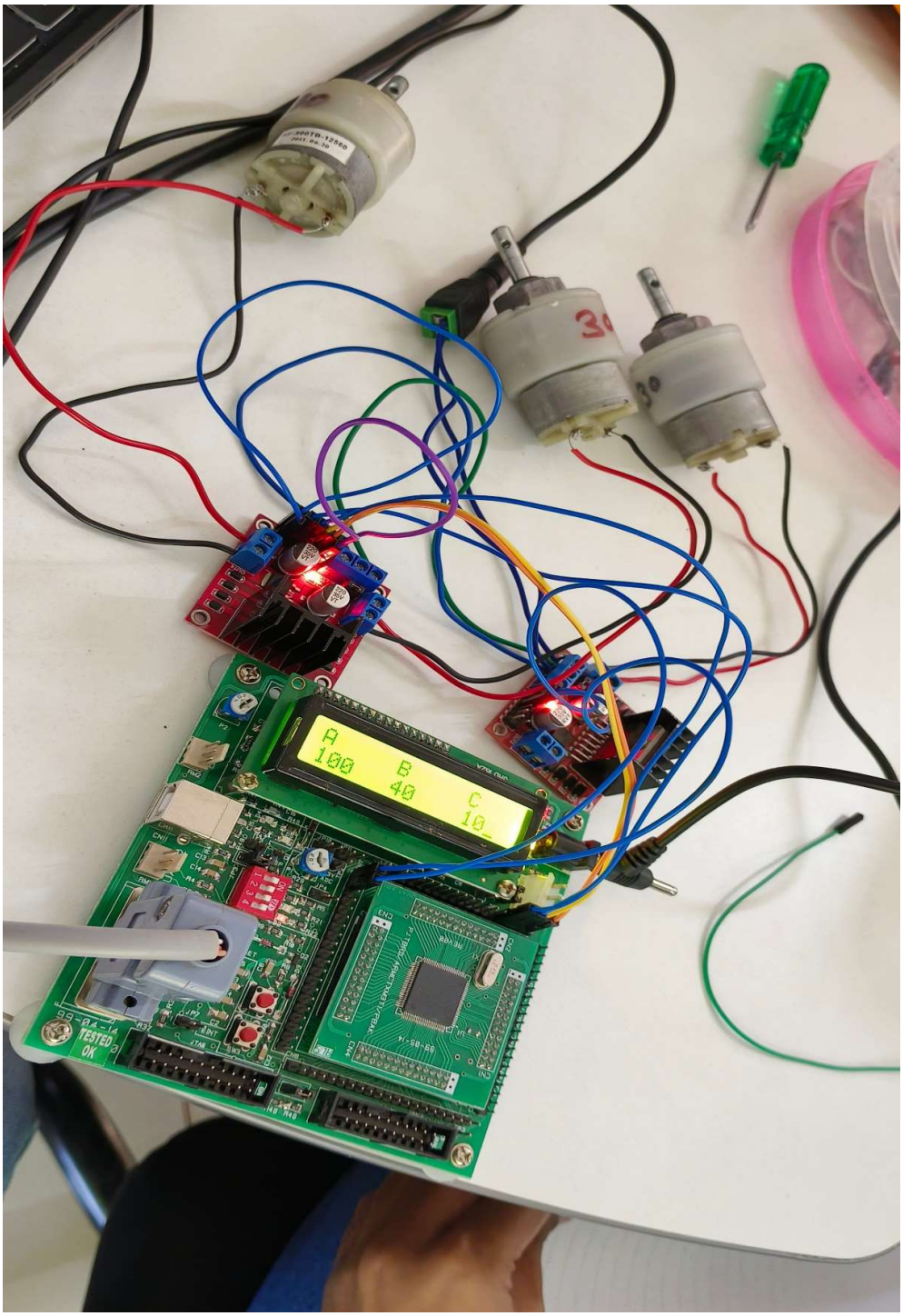
```
 }
   transmit(0X0D);
   transmit(0x0A);
}

unsigned char receive()
{
 while(!(LPC_UART0->LSR & 0x01));
 return(LPC_UART0->RBR);
}
void transmit(unsigned char ch)
{
 while(!(LPC_UART0->LSR & 0x20));
 LPC_UART0->THR = ch;
}

__task void lcd(void)
{
          delay_lcd(800);
                temp1 = 0x80;                    //1st message on LCD 1st line
        lcd_com();
                delay_lcd(800);
          lcd_puts(Msg1);

          temp1 = 0xc0;                    //2nd message on LCD 2nd line
```

```
        lcd_com();
        delay_lcd(800);
        lcd_puts(Msg2);

        os_tsk_create(dispense,10);
        os_tsk_create(order,10);
        os_tsk_delete_self();
}

void delay_ms(uint32_t ms) {
   LPC_TIM0->TCR = 0x02;  // Reset Timer
   LPC_TIM0->PR  = 0x00;  // Set prescaler to 0
   LPC_TIM0->MR0 = ms * (12000000 / 1000);  // Set match value for the
desired delay
   LPC_TIM0->TCR = 0x01;  // Enable Timer
   while (LPC_TIM0->TC < LPC_TIM0->MR0);
   LPC_TIM0->TCR = 0x00;  // Disable Timer
}

int main(void)
{
        SystemInit();
        SystemCoreClockUpdate();
  _init_box(mpool, sizeof(mpool), sizeof(U32)); // initialize the 'mpool'
memory for the membox dynamic allocation
```

```
 _init_box(mpool2, sizeof(mpool2), sizeof(U32)); // initialize the 'mpool'
memory for the membox dynamic allocation
        lcd_init();
        UART0_Init();
 os_sys_init(lcd);
 return 0;
}
```

HARDWARE IMPLEMENTATION




A
100
B
40
C
10
TESTED
OK

CONCLUSION

In conclusion, our dispensing machine, powered by a DC motor, featuring an LCD display, and utilizing UART communication in LPC1768, represents a successful integration of hardware components. The efficient design ensures precise dispensing, user-friendly interaction through the LCD, and seamless communication via UART. This project exemplifies a reliable and versatile automated system with potential applications across various industries.