**Challenge1-**

ASSIGNMENT 5

**Software user input to hardware action**

**STM32L475 has usart1 interface directly available on ST-Link. You can use this interface for printing output messages and taking user input if you're not using it for advance trace debugging over st-link.**

**Enable and configure UART1 as per experiment completed for UART2. Use polling method in while loop.**

**Write a program to user inputs from UART1. The board functional behavior as per user input should be as following-**

**=> If user sends number 1 from keyboard on serial terminal - LED1 turns HIGH & prints "LED1 ON"**

**=> If user sends number 2 from keyboard on serial terminal - both LED1 & LED2 turns HIGH & prints "LED1 & LED 2 ON"**

**=> If user sends number 3 from keyboard on serial terminal - both LED1 & LED2 switches off & prints "LED1 & LED 2 OFF"**

**=> Any other user input from terminal - both LED1 & LED2 starts blinking & prints "INVALID INPUT"**

**CODE-**

**while** (1)

{

/\* USER CODE END WHILE \*/

/\* USER CODE BEGIN 3 \*/

HAL\_UART\_Transmit(&huart1,msg,10,1000); HAL\_UART\_Receive(&huart1,msg1,10,5000);

**if** (\*msg1=='1'){

HAL\_GPIO\_WritePin(led1\_GPIO\_Port, led1\_Pin, 1); HAL\_UART\_Transmit(&huart1, msg2 ,**sizeof**(msg2),100);

}

**else if** (\*msg1=='2'){

HAL\_GPIO\_WritePin(led1\_GPIO\_Port, led1\_Pin, 1);

HAL\_GPIO\_WritePin(led2\_GPIO\_Port, led2\_Pin, 1); HAL\_UART\_Transmit(&huart1, msg3 ,**sizeof**(msg3),100);

}

**else if** (\*msg1=='3'){

HAL\_GPIO\_WritePin(led1\_GPIO\_Port, led1\_Pin, 0);

HAL\_GPIO\_WritePin(led2\_GPIO\_Port, led2\_Pin, 0); HAL\_UART\_Transmit(&huart1, msg4 ,**sizeof**(msg4),100);

}

**else** {

HAL\_GPIO\_WritePin(led1\_GPIO\_Port, led1\_Pin,1);

HAL\_Delay(10);

HAL\_GPIO\_WritePin(led2\_GPIO\_Port, led2\_Pin,1);

HAL\_Delay(10)

HAL\_GPIO\_WritePin(led1\_GPIO\_Port, led1\_Pin,0); HAL\_Delay(10); HAL\_GPIO\_WritePin(led2\_GPIO\_Port, led2\_Pin,0); HAL\_Delay(10);

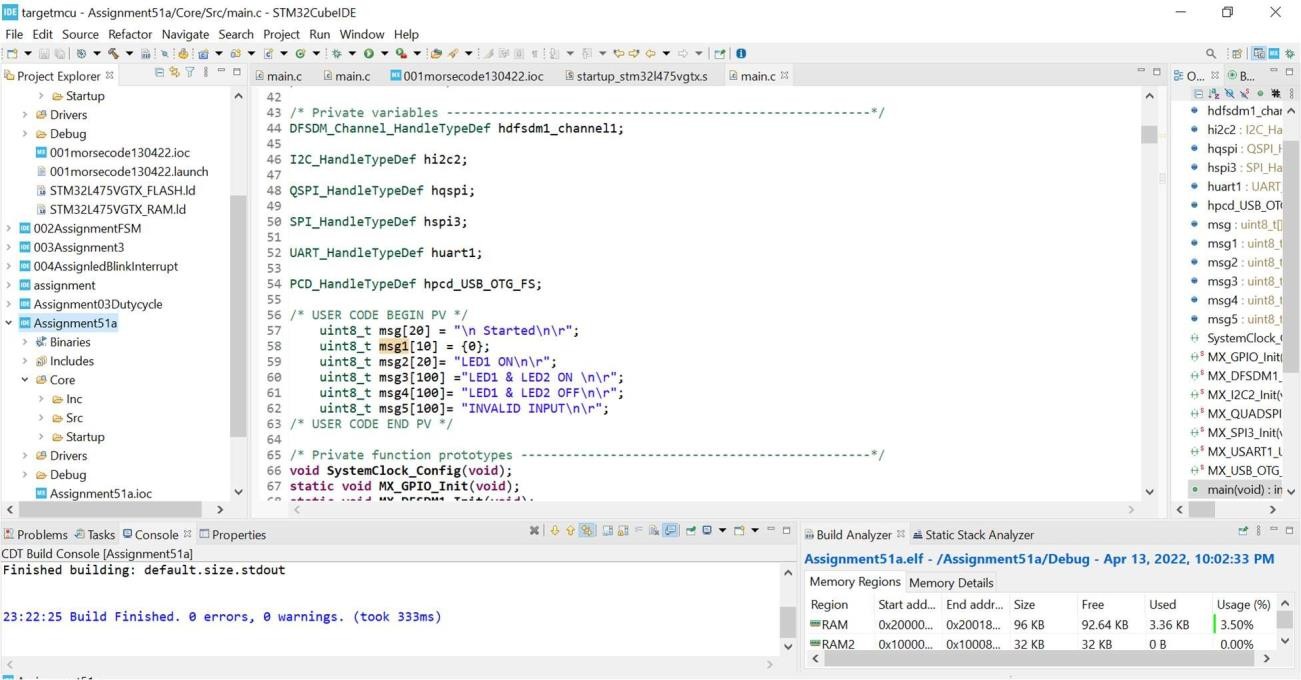
HAL\_UART\_Transmit(&huart1, msg5 ,**sizeof**(msg5),100);

}

}

Graphical user interface, text, application

Description automatically generated









.?00?AUgnmvniñM



*Sic*





F ini shed bui 1di ng : deTauit . s ize . std out



HA L\_GP S0\_Nr j t ePin ( T ed1\_GPS 0\_Po at . 1 ed j\_P in , 1) ;



Text

Description automatically generated

else {

* huartl :



- msg4 : ' '

* msg 5





Ass ig nm ent5 1a.eIf - /Assig nment5 1a/D ebug - Ap r 13, 202 2, 10:02:33 PM

’ 23: 22: 25 8u11d F:t nd shed . 6 errop,s 0 tea nnlngs . (took 333ms ) ’-RAM 0 ?0QQQ 0?Q0 8.. 96X8 3.$0%

else {



HAL cPzo or at ;

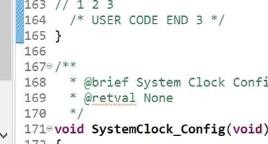




F in T s hed bui Id I ng: de fa uIt . s i ze . std out



not u^xT iran mit(&hvarti, mugs ,sizeo+(m gs),lee);



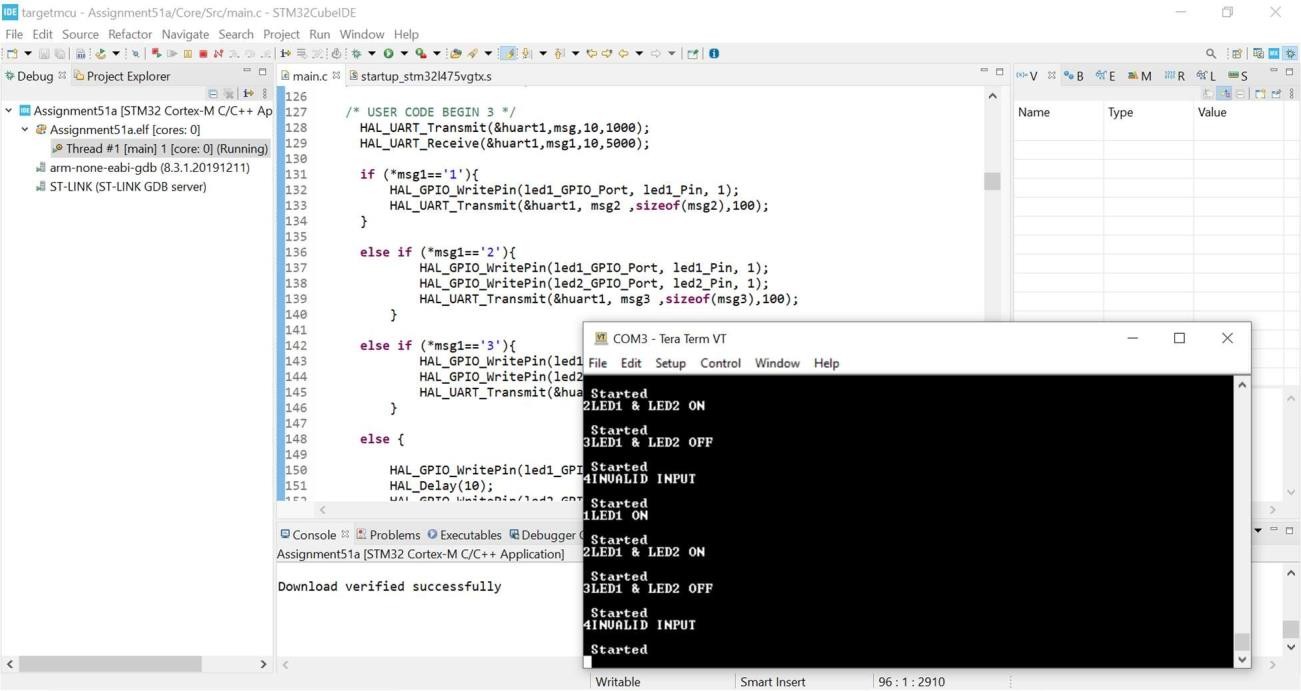
* + msg1 :



-’MXOUADZM







**Challenge 2-**

**Morse Code LED Indicator Configure uart1 in DMA mode.**

**Write the function for taking user input as per morse code valid characters and configure LED1 on STM32L475 to indicate accordingly. Check the demo here: https:/**[**/www.youtube.com/watch?v=g4IUDco9h2I&ab\_channel=SinanKARACA**](http://www.youtube.com/watch?v=g4IUDco9h2I&ab_channel=SinanKARACA)

**You can download the code from given link and take reference of the code to make it working for STM32L475 board**

**CODE:**

uint8\_t msg[] = "MORSE CODE !!\r\n";

//uint8\_t msg1[] = "LED1 ON\r\n"; uint8\_t A[] = ". -\r\n";

uint8\_t B[] = "- . . .\r\n";

uint8\_t C[] = "- . - .\r\n";

uint8\_t D[] = "- . . \r\n";

uint8\_t E[] = ".\r\n"; uint8\_t F[] = ". . - .\r\n";

uint8\_t G[] = "- - .\r\n"; uint8\_t H[] = " \r\n";

uint8\_t I[] = ". .\r\n"; uint8\_t J[] = " \r\n";

uint8\_t K[] = "- . -\r\n";

uint8\_t L[] = ". - . .\r\n";

uint8\_t M[] = "- -\r\n";

uint8\_t N[] = "- .\r\n"; uint8\_t O[] = " \r\n";

uint8\_t P[] = ". - - .\r\n";

uint8\_t Q[] = "- - . -\r\n";

uint8\_t R[] = ". - .\r\n"; uint8\_t S[] = " \r\n";

uint8\_t T[] = "-\r\n";

uint8\_t U[] = ". . -\r\n";

uint8\_t V[] = ". . . -\r\n";

uint8\_t W[] = ". - -\r\n";

uint8\_t X[] = "- . . -\r\n";

uint8\_t Y[] = "- . - -\r\n";

uint8\_t Z[] = "- - . .\r\n"; uint8\_t rcv [10] = {0}; void dot();

void space(); void dash();

/\* USER CODE END PV

/\* USER CODE BEGIN 2 \*/

HAL\_UART\_Transmit(&huart1, msg, sizeof(msg), 5000);

// HAL\_UART\_Receive(&huart1, rcv, 10, 5000);

/\* USER CODE END 2 \*/

/\* Infinite loop \*/

/\* USER CODE BEGIN WHILE \*/

while (1)

{

/\* USER CODE END WHILE \*/

HAL\_UART\_Receive(&huart1, rcv, 10, 5000); if(\*rcv=='A')

{

dot();

space();

dash();

HAL\_UART\_Transmit(&huart1, A, sizeof(A), 5000);

}

if(\*rcv=='B')

{

dash();space();dot();space();dot();space();dot(); HAL\_UART\_Transmit(&huart1, B, sizeof(B), 5000);

}

if(\*rcv=='C')

{

dash();space();dot();space();dash();space();dot();

HAL\_UART\_Transmit(&huart1, C, sizeof(C), 5000);

}

if(\*rcv=='D')

{

dash();space();dot();space();dot();space(); HAL\_UART\_Transmit(&huart1, D, sizeof(D), 5000);

}

if(\*rcv=='E')

{

dot();

HAL\_UART\_Transmit(&huart1, E, sizeof(E), 5000);

}

if(\*rcv=='F')

{

dot();space();dot();space();dash();space();dot();

HAL\_UART\_Transmit(&huart1, F, sizeof(F), 5000);

}

if(\*rcv=='G')

{

dash();space();dash();space();dot(); HAL\_UART\_Transmit(&huart1, G, sizeof(G), 5000);

}

if(\*rcv=='H')

{

dot();space();dot();space();dot();space();dot(); HAL\_UART\_Transmit(&huart1, H, sizeof(H), 5000);

}

if(\*rcv=='I')

{

dot();space();dot();

HAL\_UART\_Transmit(&huart1, I, sizeof(I), 5000);

}

if(\*rcv=='J')

{

dot();space();dash();space();dash();space();dash(); HAL\_UART\_Transmit(&huart1, J, sizeof(J), 5000);

}

if(\*rcv=='K')

{

dash();space();dot();space();dash(); HAL\_UART\_Transmit(&huart1, K, sizeof(K), 5000);

}

if(\*rcv=='L')

{

dot();space();dash();space();dot();space();dot();

HAL\_UART\_Transmit(&huart1, L, sizeof(L), 5000);

}

if(\*rcv=='M')

{

dash();space();dash(); HAL\_UART\_Transmit(&huart1, M, sizeof(M), 5000);

}

if(\*rcv=='N')

{

dash();space();dot(); HAL\_UART\_Transmit(&huart1, N, sizeof(N), 5000);

}

if(\*rcv=='O')

{

dash();space();dash();space();dash();space(); HAL\_UART\_Transmit(&huart1, O, sizeof(O), 5000);

}

if(\*rcv=='P')

{

dot();space();dash();space();dash();space();dot(); HAL\_UART\_Transmit(&huart1, P, sizeof(P), 5000);

}

if(\*rcv=='Q')

{

dash();space();dash();space();dot();space();dash(); HAL\_UART\_Transmit(&huart1, Q, sizeof(Q), 5000);

}

if(\*rcv=='R')

{

dot();space();dash();space();dot(); HAL\_UART\_Transmit(&huart1, R, sizeof(R), 5000);

}

if(\*rcv=='S')

{

dot();space();dot();space();dot(); HAL\_UART\_Transmit(&huart1, S, sizeof(S), 5000);

}

if(\*rcv=='T')

{

dash();

HAL\_UART\_Transmit(&huart1, T, sizeof(T), 5000);

}

if(\*rcv=='U')

{

dot();space();dot();space();dash(); HAL\_UART\_Transmit(&huart1, U, sizeof(U), 5000);

}

if(\*rcv=='V')

{

dot();space();dot();space();dot();space();dash(); HAL\_UART\_Transmit(&huart1, V, sizeof(V), 5000);

}

if(\*rcv=='W')

{

dot();space();dash();space();dash(); HAL\_UART\_Transmit(&huart1, W, sizeof(W), 5000);

}

if(\*rcv=='X')

{

dash();space();dot();space();dot();space();dash(); HAL\_UART\_Transmit(&huart1, X, sizeof(X), 5000);

}

if(\*rcv=='Y')

{

dash();space();dot();space();dash();space();dash(); HAL\_UART\_Transmit(&huart1, Y, sizeof(Y), 5000);

}

if(\*rcv=='Z')

{

dash();space();dash();space();dot();space();dot(); HAL\_UART\_Transmit(&huart1, Z, sizeof(Z), 5000);

}

/\* USER CODE BEGIN 3 \*/

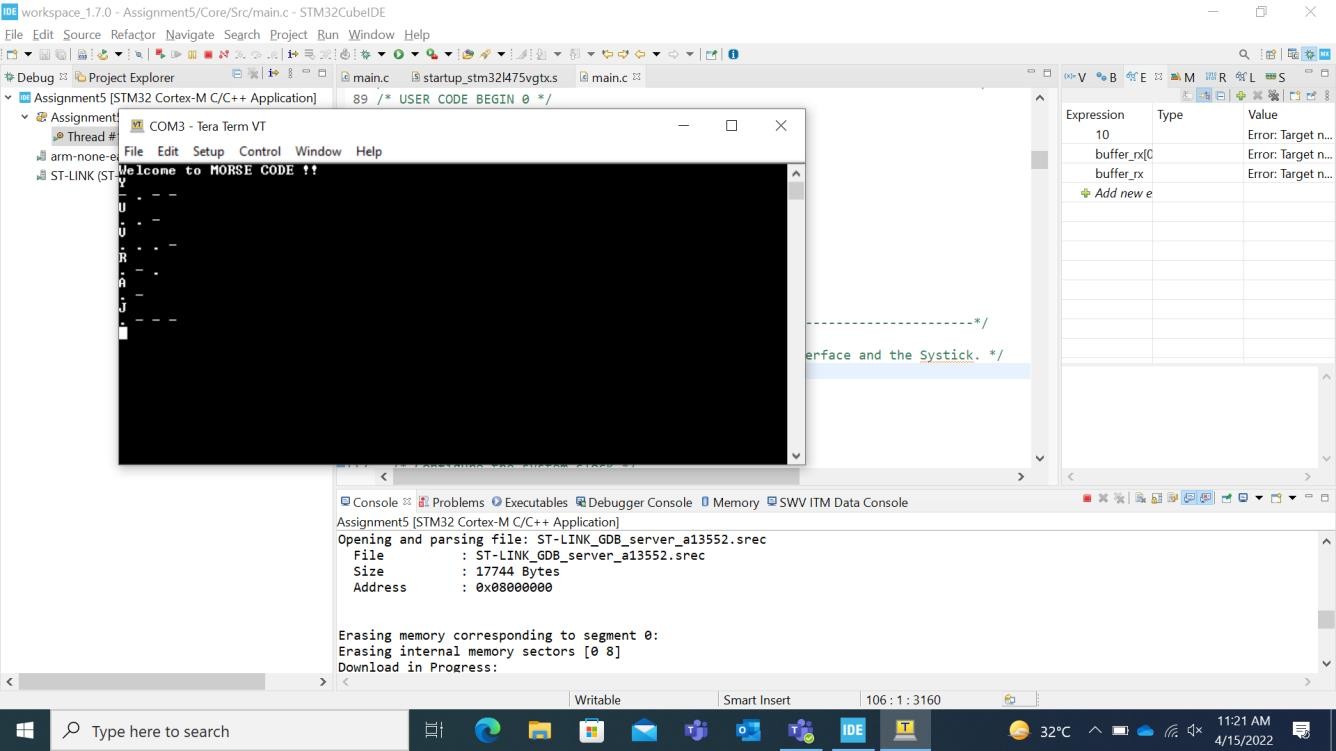
}

/\* USER CODE END 3 \*/

}

Graphical user interface, text, application

Description automatically generated

**Output of Morse code on the Tera Terminal****-**