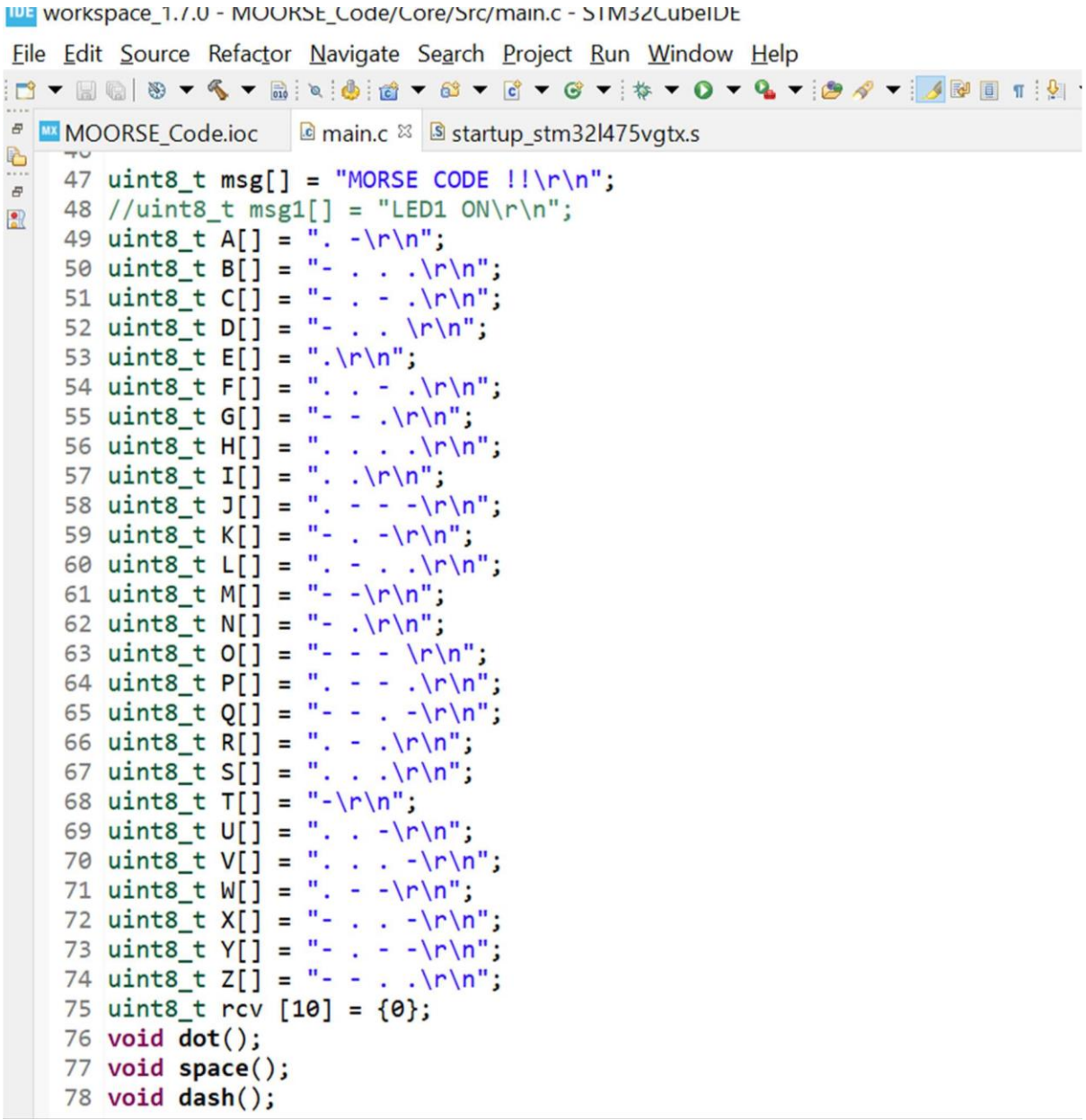
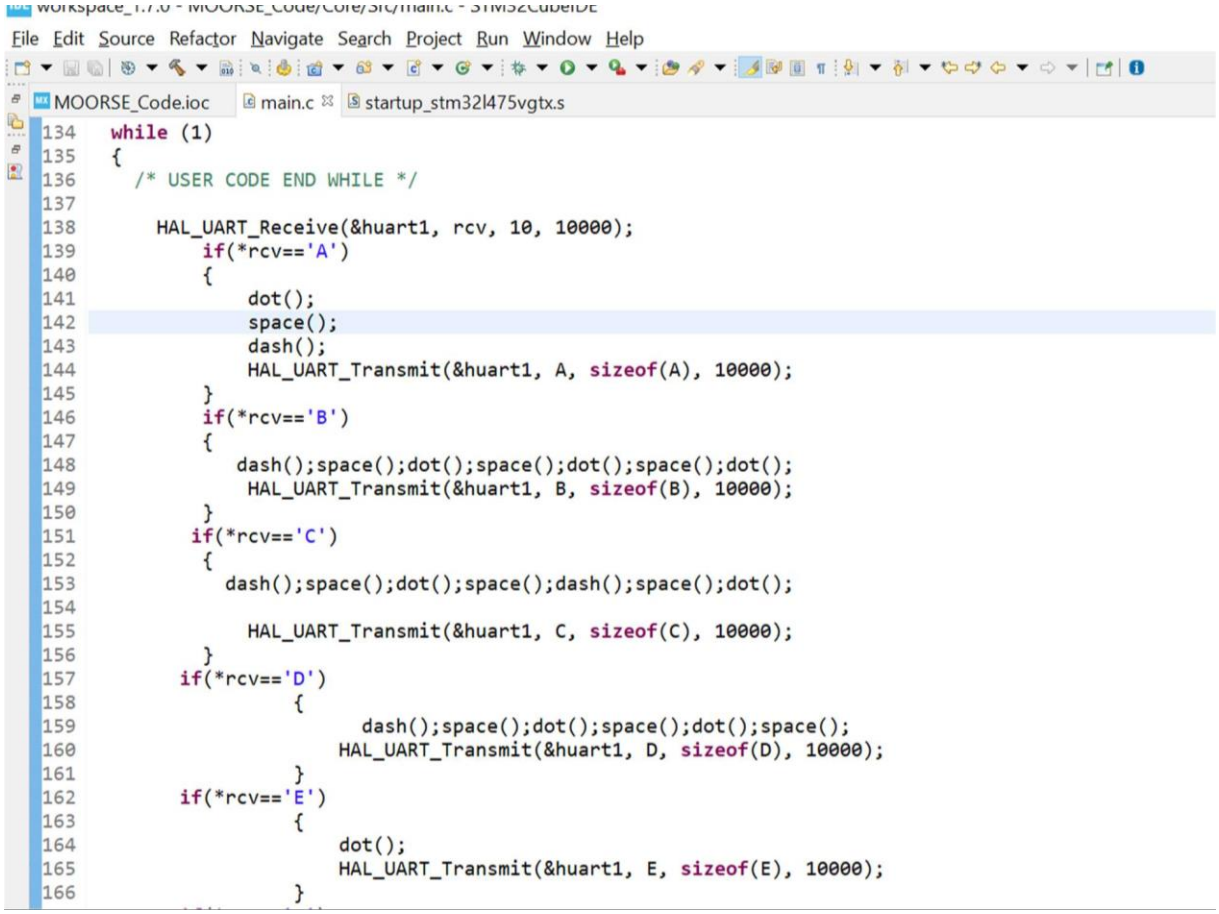
MCU
Assignment-5

Morse Code LED Indicator

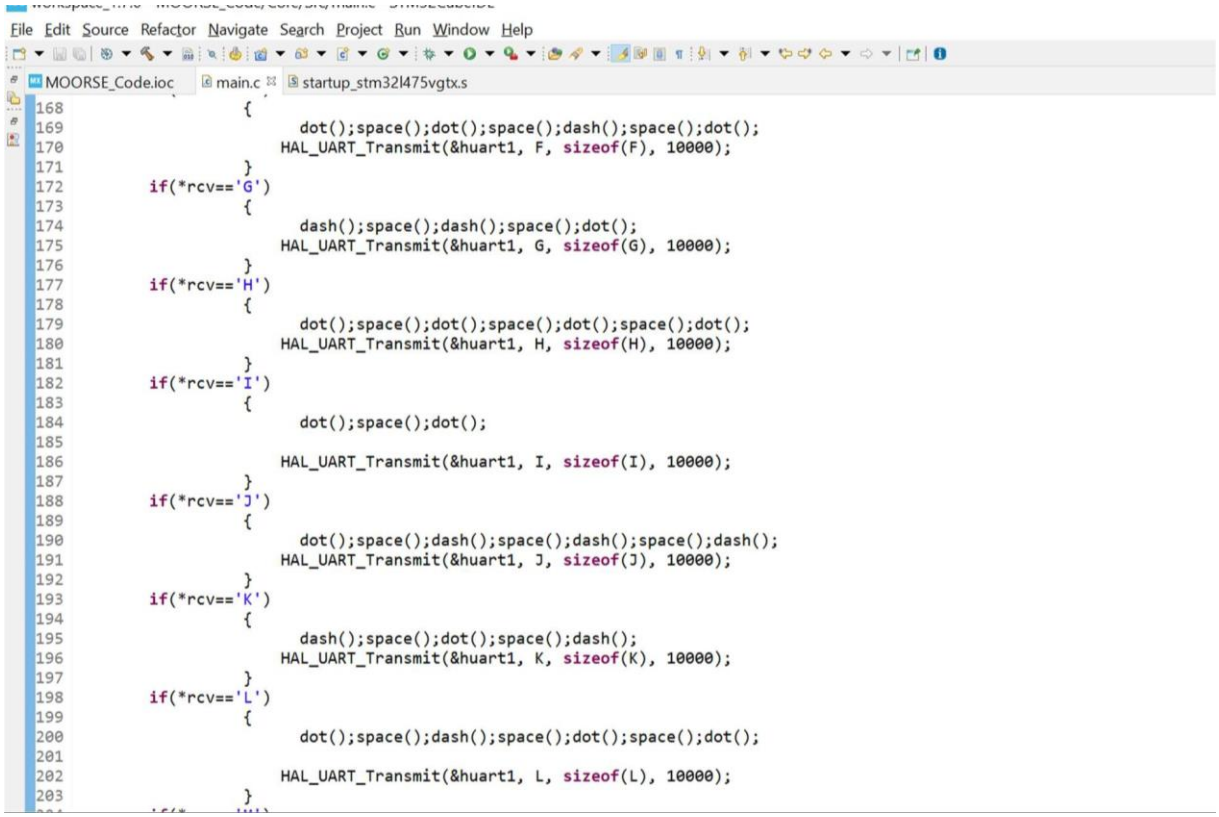
CODE:



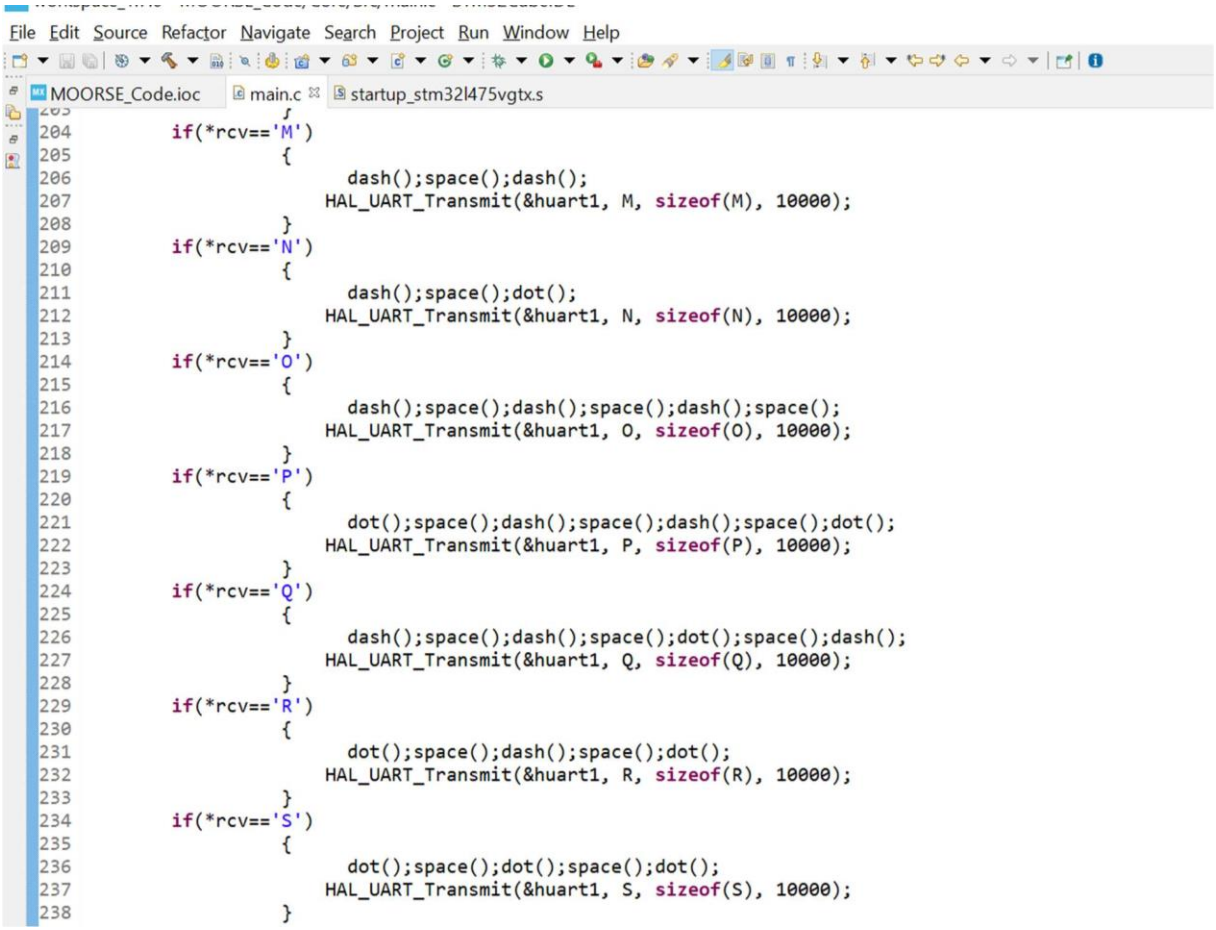

```
47 uint8_t msg[] = "MORSE CODE !!\r\n";
48 //uint8_t msg1[] = "LED1 ON\r\n";
49 uint8_t A[] = ". -\r\n";
50 uint8_t B[] = "- . . .\r\n";
51 uint8_t C[] = "- . - .\r\n";
52 uint8_t D[] = "- . . \r\n";
53 uint8_t E[] = ".\r\n";
54 uint8_t F[] = ". . - .\r\n";
55 uint8_t G[] = "- - .\r\n";
56 uint8_t H[] = ". . . .\r\n";
57 uint8_t I[] = ". .\r\n";
58 uint8_t J[] = ". - - -\r\n";
59 uint8_t K[] = "- . -\r\n";
60 uint8_t L[] = ". - . .\r\n";
61 uint8_t M[] = "- -\r\n";
62 uint8_t N[] = "- .\r\n";
63 uint8_t O[] = "- - - \r\n";
64 uint8_t P[] = ". - - .\r\n";
65 uint8_t Q[] = "- - . -\r\n";
66 uint8_t R[] = ". - .\r\n";
67 uint8_t S[] = ". . .\r\n";
68 uint8_t T[] = "-\r\n";
69 uint8_t U[] = ". . -\r\n";
70 uint8_t V[] = ". . . -\r\n";
71 uint8_t W[] = ". - -\r\n";
72 uint8_t X[] = "- . . -\r\n";
73 uint8_t Y[] = "- . - -\r\n";
74 uint8_t Z[] = "- - . .\r\n";
75 uint8_t rcv [10] = {0};
76 void dot();
77 void space();
78 void dash();
```



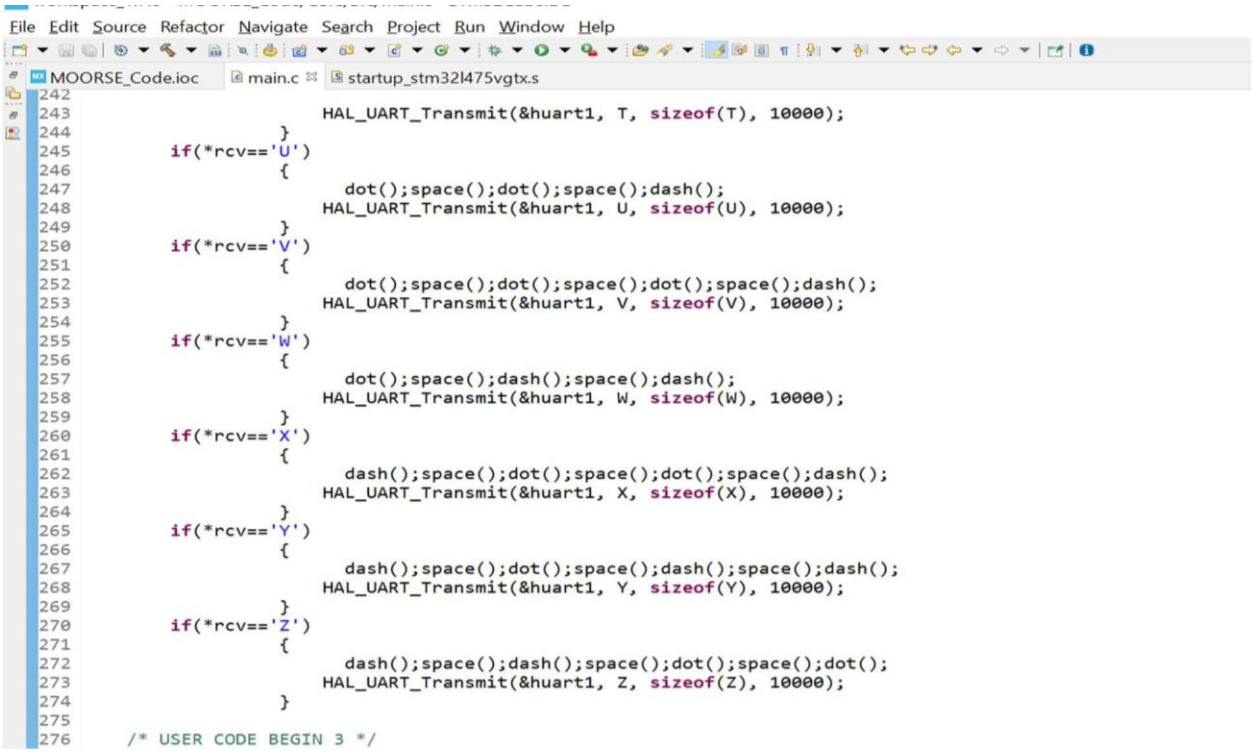

```
134  while (1)
135  {
136    /* USER CODE END WHILE */
137
138      HAL_UART_Receive(&huart1, rcv, 10, 10000);
139          if(*rcv=='A')
140          {
141              dot();
142              space();
143              dash();
144              HAL_UART_Transmit(&huart1, A, sizeof(A), 10000);
145          }
146          if(*rcv=='B')
147          {
148             dash();space();dot();space();dot();space();dot();
149              HAL_UART_Transmit(&huart1, B, sizeof(B), 10000);
150          }
151         if(*rcv=='C')
152          {
153            dash();space();dot();space();dash();space();dot();
154
155              HAL_UART_Transmit(&huart1, C, sizeof(C), 10000);
156          }
157        if(*rcv=='D')
158                  {
159                        dash();space();dot();space();dot();space();
160                    HAL_UART_Transmit(&huart1, D, sizeof(D), 10000);
161                  }
162        if(*rcv=='E')
163                  {
164                      dot();
165                      HAL_UART_Transmit(&huart1, E, sizeof(E), 10000);
166                  }
```



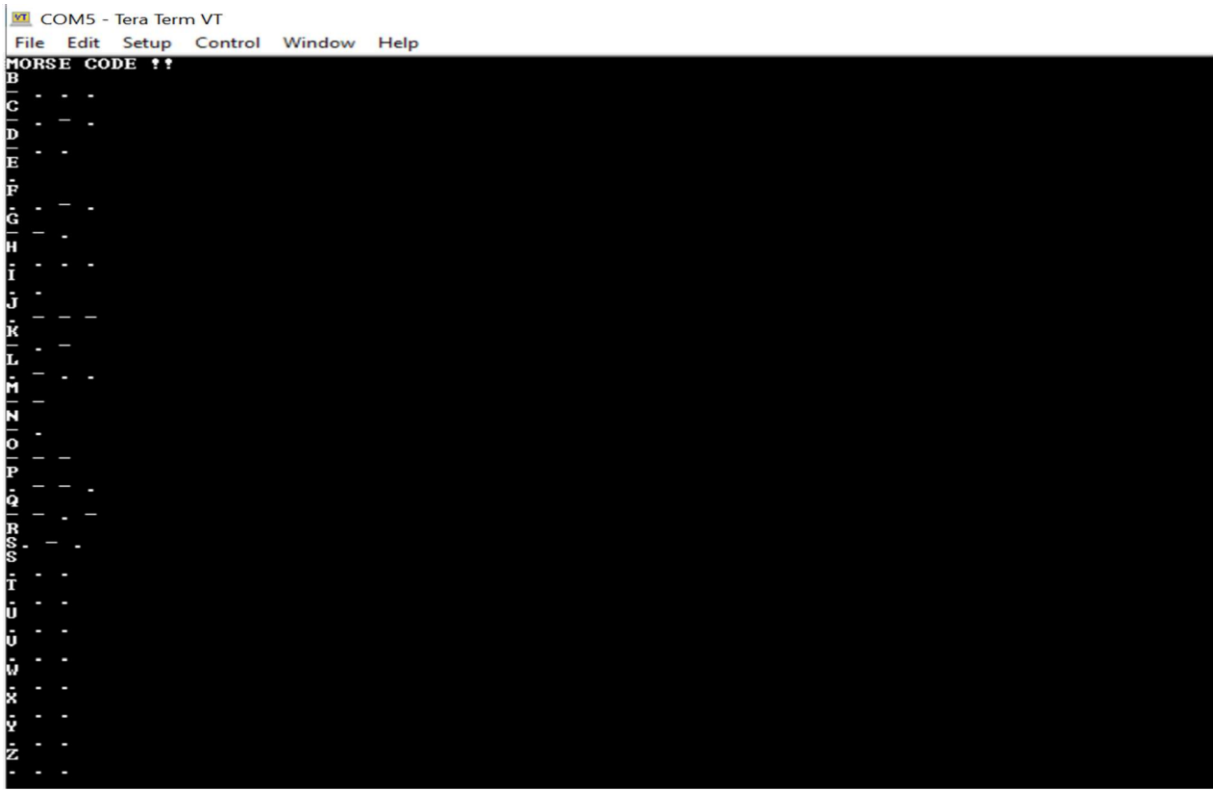

```
168                     {
169                       dot();space();dot();space();dash();space();dot();
170                     HAL_UART_Transmit(&huart1, F, sizeof(F), 10000);
171                     }
172             if(*rcv=='G')
173                     {
174                       dash();space();dash();space();dot();
175                     HAL_UART_Transmit(&huart1, G, sizeof(G), 10000);
176                     }
177             if(*rcv=='H')
178                     {
179                       dot();space();dot();space();dot();space();dot();
180                     HAL_UART_Transmit(&huart1, H, sizeof(H), 10000);
181                     }
182             if(*rcv=='I')
183                     {
184                       dot();space();dot();
185
186                     HAL_UART_Transmit(&huart1, I, sizeof(I), 10000);
187                     }
188             if(*rcv=='J')
189                     {
190                       dot();space();dash();space();dash();space();dash();
191                     HAL_UART_Transmit(&huart1, J, sizeof(J), 10000);
192                     }
193             if(*rcv=='K')
194                     {
195                       dash();space();dot();space();dash();
196                     HAL_UART_Transmit(&huart1, K, sizeof(K), 10000);
197                     }
198             if(*rcv=='L')
199                     {
200                       dot();space();dash();space();dot();space();dot();
201
202                     HAL_UART_Transmit(&huart1, L, sizeof(L), 10000);
203                     }
```



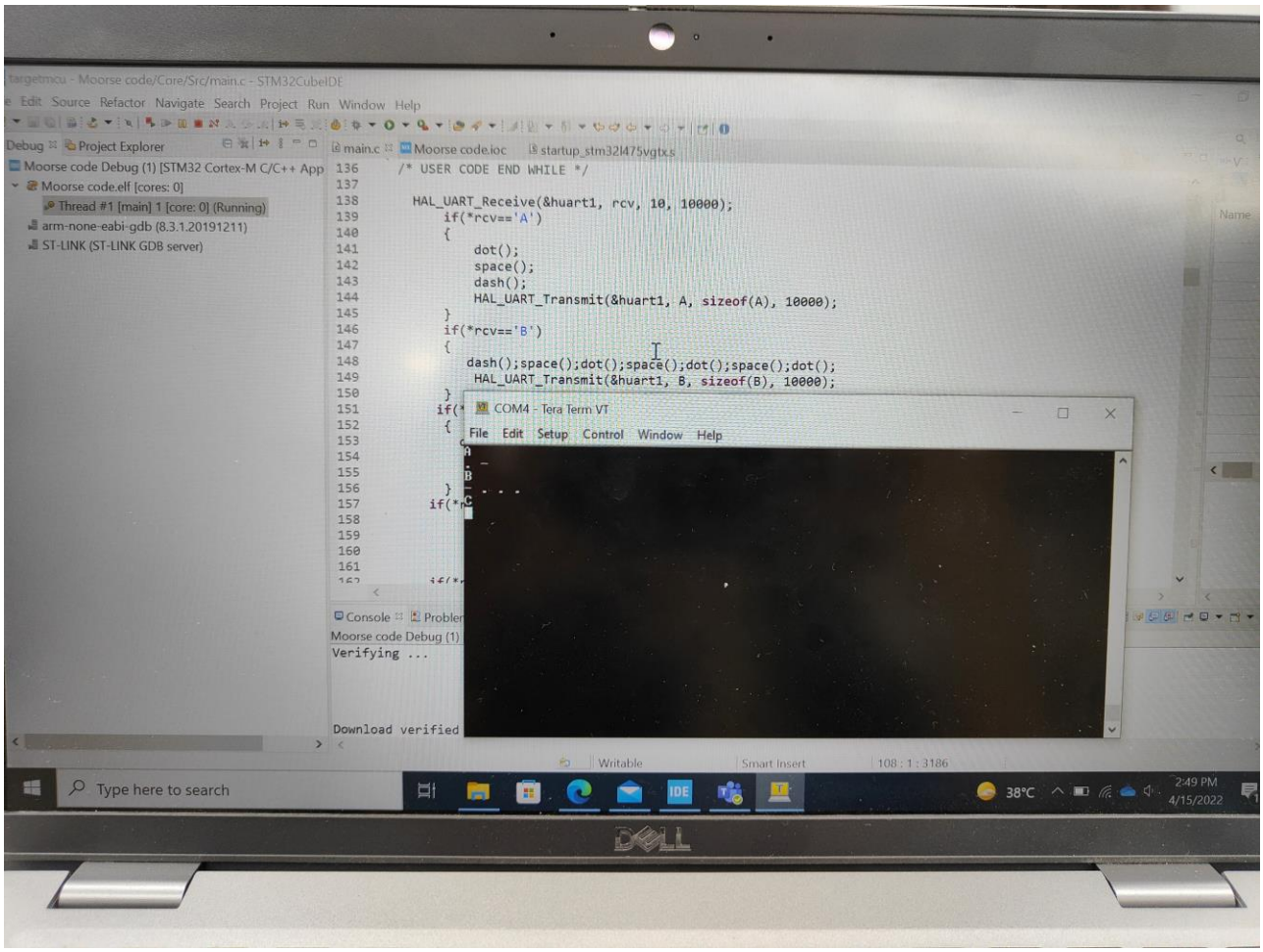

```
204         if(*rcv=='M')
205                 {
206                     dash();space();dash();
207                   HAL_UART_Transmit(&huart1, M, sizeof(M), 10000);
208                 }
209         if(*rcv=='N')
210                 {
211                     dash();space();dot();
212                   HAL_UART_Transmit(&huart1, N, sizeof(N), 10000);
213                 }
214         if(*rcv=='O')
215                 {
216                     dash();space();dash();space();dash();space();
217                   HAL_UART_Transmit(&huart1, O, sizeof(O), 10000);
218                 }
219         if(*rcv=='P')
220                 {
221                     dot();space();dash();space();dash();space();dot();
222                   HAL_UART_Transmit(&huart1, P, sizeof(P), 10000);
223                 }
224         if(*rcv=='Q')
225                 {
226                     dash();space();dash();space();dot();space();dash();
227                   HAL_UART_Transmit(&huart1, Q, sizeof(Q), 10000);
228                 }
229         if(*rcv=='R')
230                 {
231                     dot();space();dash();space();dot();
232                   HAL_UART_Transmit(&huart1, R, sizeof(R), 10000);
233                 }
234         if(*rcv=='S')
235                 {
236                     dot();space();dot();space();dot();
237                   HAL_UART_Transmit(&huart1, S, sizeof(S), 10000);
238                 }
```

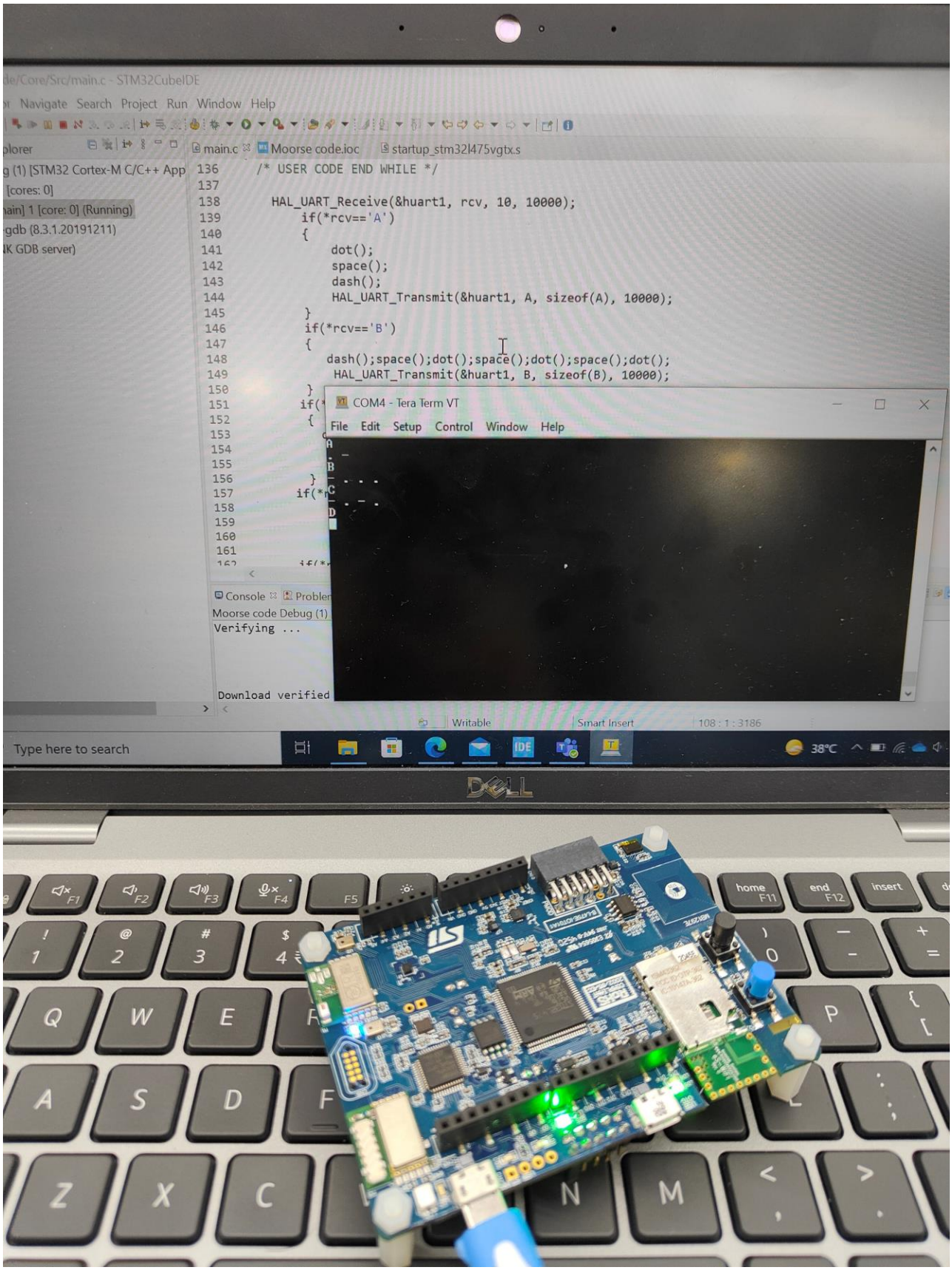

File Edit Source Refactor Navigate Search Project Run Window Help
MOORSE_Code.ioc
main.c
startup_stm32l475vgtx.s
242
243 HAL_UART_Transmit(&huart1, T, sizeof(T), 10000);
244 }
245 if(*rcv=='U')
246 {
247 dot();space();dot();space();dash();
248 HAL_UART_Transmit(&huart1, U, sizeof(U), 10000);
249 }
250 if(*rcv=='V')
251 {
252 dot();space();dot();space();dot();space();dash();
253 HAL_UART_Transmit(&huart1, V, sizeof(V), 10000);
254 }
255 if(*rcv=='W')
256 {
257 dot();space();dash();space();dash();
258 HAL_UART_Transmit(&huart1, W, sizeof(W), 10000);
259 }
260 if(*rcv=='X')
261 {
262 dash();space();dot();space();dot();space();dash();
263 HAL_UART_Transmit(&huart1, X, sizeof(X), 10000);
264 }
265 if(*rcv=='Y')
266 {
267 dash();space();dot();space();dash();space();dash();
268 HAL_UART_Transmit(&huart1, Y, sizeof(Y), 10000);
269 }
270 if(*rcv=='Z')
271 {
272 dash();space();dash();space();dot();space();dot();
273 HAL_UART_Transmit(&huart1, Z, sizeof(Z), 10000);
274 }
275
276 /* USER CODE BEGIN 3 */

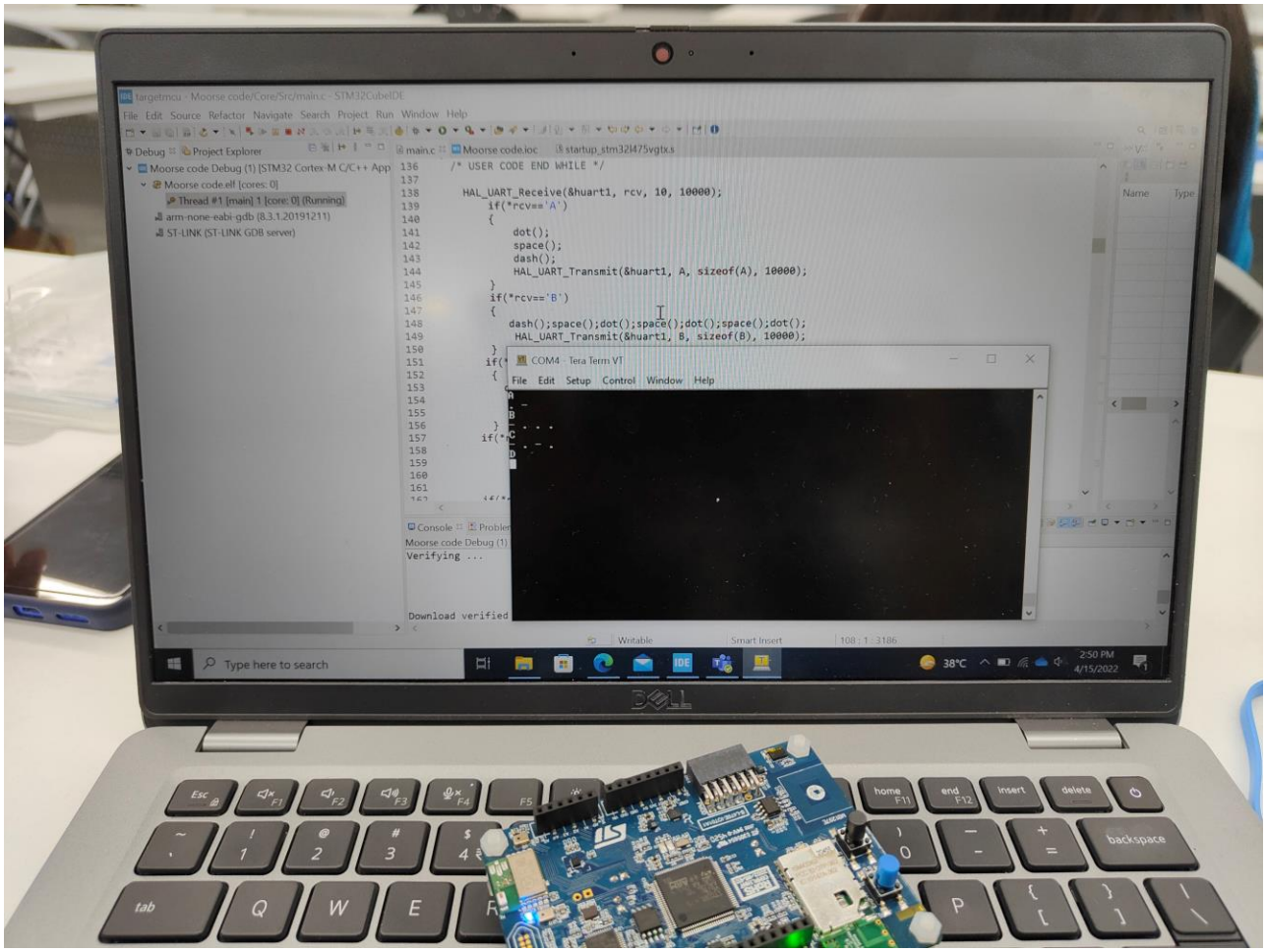
OUTPUT :


COM5 - Tera Term VT
File Edit Setup Control Window Help
MORSE CODE !!
B - - -
C - _ -
D - -
E .
F . _ -
G _ -
H . . . .
I . .
J . _ _ _
K _ . _
L . _ . .
M _ _
N _ .
O _ _ _
P . _ _ .
Q _ _ . _
R . _ .
S . . .
T _ -
U . . _
V . . . _
W . _ _
X _ . . _
Y _ . _ _
Z _ _ . .

HARDWARE OUTPUT:




main.c
Moorse code.ioc
startup_stm32l475vgtx.s
/* USER CODE END WHILE */
HAL_UART_Receive(&huart1, rcv, 10, 10000);
if(*rcv=='A')
dot();
space();
dash();
HAL_UART_Transmit(&huart1, A, sizeof(A), 10000);
if(*rcv=='B')
dash();space();dot();space();dot();space();dot();
HAL_UART_Transmit(&huart1, B, sizeof(B), 10000);
COM4 - Tera Term VT
File Edit Setup Control Window Help
A . _
B _ . . .
C _ . _ .
Moorse code Debug (1)
Verifying ...
Download verified
Writable
Smart Insert
108 : 1 : 3186
Type here to search
38°C


targetmcu - Moorse code/Core/Src/main.c - STM32CubeIDE
File Edit Source Refactor Navigate Search Project Run Window Help
Debug
Project Explorer
Moorse code Debug (1) [STM32 Cortex-M C/C++ App
Moorse code.elf [cores: 0]
Thread #1 [main] 1 [core: 0] (Running)
arm-none-eabi-gdb (8.3.1.20191211)
ST-LINK (ST-LINK GDB server)
main.c
Moorse code.ioc
startup_stm32l475vgtx.s
/* USER CODE END WHILE */
HAL_UART_Receive(&huart1, rcv, 10, 10000);
if(*rcv=='A')
{
dot();
space();
dash();
HAL_UART_Transmit(&huart1, A, sizeof(A), 10000);
}
if(*rcv=='B')
{
dash();space();dot();space();dot();space();dot();
HAL_UART_Transmit(&huart1, B, sizeof(B), 10000);
}
COM4 - Tera Term VT
File Edit Setup Control Window Help
A
. _
B
_ . . .
C
_ . _ .
D
Console
Moorse code Debug (1)
Verifying ...
Download verified
Writable
Smart Insert
Type here to search
38°C
2:50 PM
4/15/2022