Microcontrollers and Interface

Youshae Bin Mansoor Jafri & Shaheer Baig

Task 0:

```
//Task 0
  HAL_UART_Transmit(&huart2, (uint8_t *)"Hello World", 10, HAL_MAX_DELAY);
```

```
labt1@administrator-OptiPlex-Tower-7020: ~
Hello, World!Hello, World!Hello, World!Hello, World!Hello, World!Hello, World!He
llo, World!Hello, World!Hello, World!Hello, World!Hello, World!Hello, World!Hell
o, World!Hello, World!Hello, World!Hello, World!Hello, World!Hello, World!Hello,
 World!Hello, World!Hello, World!Hello, World!Hello, World!Hello, World!Hello, W
orld!Hello, World!Hello, World!Hello, World!Hello, World!Hello, World!Hello, Wor
ld!Hello, World!Hello, World!Hello, World!Hello, World!Hello, World!Hello, World
!Hello, World!Hello, World!Hello, World!Hello, World!Hello, World!Hello, World!H
ello, World!Hello, World!Hello, World!Hello, World!Hello, World!Hello, World!Hel
lo, World!Hello, World!Hello, World!Hello, World!Hello, World!Hello, World!Hello
, World!Hello, World!Hello, World!Hello, World!Hello, World!Hello, World!Hello,
World!Hello, World!Hello, World!Hello, World!Hello, World!Hello, World!Hello, Wo
rld!Hello, World!Hello, World!Hello, World!Hello, World!Hello, World!Hello, Worl
d!Hello, World!Hello, World!Hello, World!Hello, World!Hello, World!Hello, World!
Hello, World!Hello, World!Hello, World!Hello, World!Hello, World!Hello, World!He
llo, World!Hello, World!Hello, World!Hello, World!Hello, World!Hello, World!Hell
o, World!Hello, World!Hello, World!Hello, World!Hello, World!Hello, World!Hello,
 World!Hello, World!Hello, World!Hello, World!Hello, World!Hello, World!Hello, W
orld!Hello, World!Hello, World!Hello, World!Hello, World!Hello, World!Hello, Wor
ld!Hello, World!Hello, World!Hello, World!Hello, World!Hello, World!Hello, World
!Hello, World!Hello, World!Hello, World!Hello, World!Hello, World!Hello, World!H
ello, World!Hello, World!Hello, World!Hello, World!Hello, World!Hello, World!Hel
lo, World!Hello, World!Hello, World!Hello, World!Hello, World!Hello, World!Hello
, World!Hello, World!Hello, World!Hello, World!Hello, World!Hello, World!Hello,
World!Hello, World!Hello, World!Hello, World!Hello, World!Hello, World!Hello
```

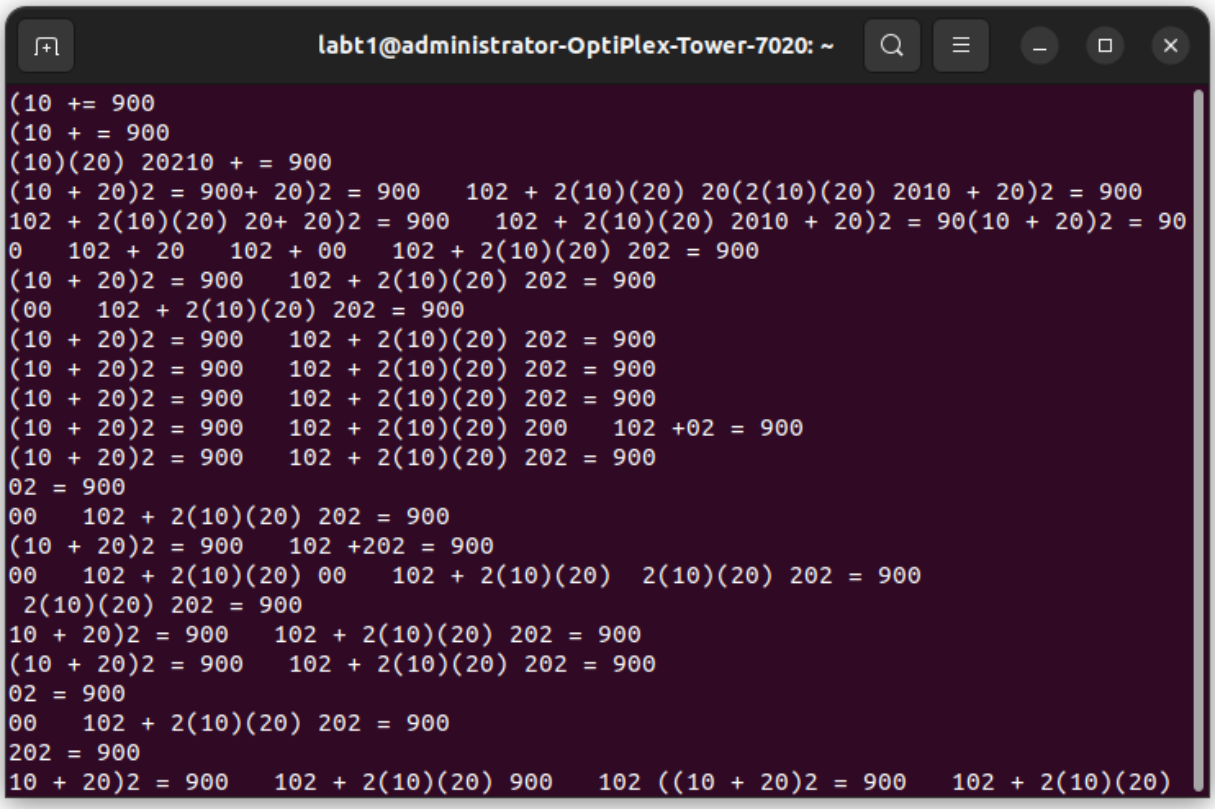
Task 1:

```
void myPrintf(const char *fmt, ...)
{
    char buffer[128];
    va_list args;
    va_start(args, fmt);
    int len = vsnprintf(buffer, sizeof(buffer), fmt, args);
    va_end(args);

        HAL_UART_Transmit(&huart2, (uint8_t *)buffer, len, HAL_MAX_DELAY);

}

    // Task 1
    int x = 42;
    float y = 3.14f;
    myPrintf("Value of x = %d, y = %.2f\r\n", x, y);
```

```
alue of x = 42, y =
```

Task 2:

```
// Task 2
int a = 10;
int b = 20;

int lhs = (a + b) * (a + b);
int rhs = a*a + 2*a*b + b*b;
myPrintf("(%d + %d)2 = %d", a, b, lhs);
myPrintf("   %d2 + 2(%d)(%d) %d2 = %d\n\r", a, a, b, b, rhs);
```




```
10 + 20)2 = 900   102 + 2(10)(20(10 + 20)2 = 900
```

Task 3:

```c
// Task 3
char str[] = "Microcontrollers";
int key = 10059;
myPrintf("\tInitial String :%s", str);
for (int i = 0; i < strlen(str); i++) {
    str[i] = (char)((int)str[i] + (key % 256));
}

//HAL_Delay(2000);
HAL_Delay(200);
myPrintf("\tEncrypted String :%s", str);

HAL_Delay(200);
for (int i = 0; i < strlen(str); i++) {
    str[i] = (char)((int)str[i] - (key % 256));
}
//HAL_Delay(2000);
myPrintf("\tDecrypted String :%s\r\n", str);
```

```
        Initial String :Microcontrollerspted String :◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆Initial Str
ing :Microcontrol :◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆     Decrypted String :Microcontrollers
```

Task 4:

```
//Task4

int A[2][2] = {{1, 2}, {3, 4}};
int B[2][2] = {{5, 6}, {7, 8}};
myPrintf("Marix A:\n\r");


for(int i=0;i<2;i++){
  for(int j=0;j<2;j++){
    myPrintf("%d ",A[i][j]);
  }
  myPrintf("\r\n");
}

myPrintf("Marix B:\n\r");
for(int i=0;i<2;i++){
  for(int j=0;j<2;j++){
    myPrintf("%d ",B[i][j]);
  }
  myPrintf("\r\n");
}
myPrintf("Matrix Multiplication Result:\r\n");

for (int i = 0; i < 2; i++) {
  for (int j = 0; j < 2; j++) {
      int sum = 0;
      for (int k = 0; k < 2; k++) {
          sum += A[i][k] * B[k][j];
      }
      myPrintf("%d \t", sum);
  }
  myPrintf("\r\n");
}
```

```
Matrix Multiplicatarix A:
1 2
3 4
Marix B:
5 6
7 8
Matrix Multiplication Result:
19      22
43      50
```

Task 5:

```c
// Task 5
int s = 100;
int e = 999;

myPrintf("Armstrong numbers between %d and %d are:\n\r", s, e);
for (int i = s; i <= e; i++) {
  int k = i;
    int fnum = k % 10;
    k = k / 10;
    int snum = k % 10;
    k = k / 10;
    int tnum = k;

    int armstrong = (tnum * tnum * tnum) + (snum * snum * snum) + (fnum * fnum * fnum);

    if (armstrong == i) {
        myPrintf("%d\n\r", i);
    }
}
```

```
Armstrong numbers between 100 and 999 aArmstrong numbers between 100 and 999 are
:
153
370
371
407
```

Evaluation

a. USART2 is essentially a translator within the STM32F3 board that takes internal data from the computer and sends it out one bit at a time. To make this work with an external USB-UART module, the wires must be 'crossed.' That is because the TX pin is where the data is leaving the board, so it needs to go to the RX pin where the other device is receiving it. Because if you connect TX to TX then both devices are trying to transmit the data on the same wire and nothing will get through. It is this 'TX-to-RX' setup that provides a clear path for correct data to flow back and forth.