Problem 1 :

|  |
| --- |
| **int** counter = 50;  **int** halfTimeCounter =0;  uint8\_t numberPA = 6;  **const** **int** MAX\_LED = 4;  **int** led\_buffer[4] = {1,2,3,4};  **int** status = 0;  **int** buffer[4] = {1,2,3,0};  /\* \*/  **void** **Exercise1**(){  counter--;  **if**(counter <=0){  counter = 50;  **if**(numberPA == 6){  HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_6, *GPIO\_PIN\_RESET* );  HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_7, *GPIO\_PIN\_SET* );  SevenSegmentBCD(1);  numberPA = 7 ;  }  **else** **if** ( numberPA == 7)  {  HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_6, *GPIO\_PIN\_SET* );  HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_7, *GPIO\_PIN\_RESET* );  SevenSegmentBCD(2);  numberPA = 6 ;  }  }  } |

Problem 2:

|  |
| --- |
| **void** **Exercise2**(){  counter--;  **if**(counter <=0){  counter = 50;  halfTimeCounter++;  **if**(halfTimeCounter > 2){  HAL\_GPIO\_TogglePin(GPIOA, GPIO\_PIN\_4);  HAL\_GPIO\_TogglePin(GPIOA, GPIO\_PIN\_5);  halfTimeCounter = 0;  }  HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_4 | GPIO\_PIN\_5 | GPIO\_PIN\_6 | GPIO\_PIN\_7 | GPIO\_PIN\_8 | GPIO\_PIN\_9 , *GPIO\_PIN\_SET*); // clear all Led  HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_6 << status, *GPIO\_PIN\_RESET*);  SevenSegmentBCD(buffer[status]);  status++;  **if**(status >=4) status = 0;  }  } |

Problem3 , Problem 4 :

|  |
| --- |
| **void** **Exercise4**(){  Ex4Counter--;  **if**(Ex4Counter <= 0){  **if**(Ex4Counter % 25 ==0){  update7SEG(index\_led);  SevenSegmentBCD(buffer[index\_led]);  index\_led++;  **if**(index\_led >=4) index\_led = 0;  }  Ex4Counter=100;  }  } |

Problem 5:

|  |
| --- |
| **void** **Ex5**(){  second ++;  **if** ( second >= 60) {  second = 0;  minute ++;  }  **if**( minute >= 60) {  minute = 0;  hour ++;  }  **if**( hour >=24) {  hour = 0;  }  updateClockBuffer () ;  HAL\_Delay (1000) ;  } **void** **updateClockBuffer**(){  led\_buffer[0] = hour / 10;  led\_buffer[1] = hour % 10;  led\_buffer[2] = minute / 10;  led\_buffer[3] = minute % 10;  } |

Problem 6 7 8 :

|  |
| --- |
| **void** **Exercise7**(){  second ++;  **if** ( second >= 60) {  second = 0;  minute ++;  }  **if**( minute >= 60) {  minute = 0;  hour ++;  }  **if**( hour >=24) {  hour = 0;  }  updateClockBuffer () ;  }  setTimer0(1000);  setTimer1(1000);  setTimer2(1000);  **while** (1)  {  /\* USER CODE END WHILE \*/  /\* USER CODE BEGIN 3 \*/  **if**(timer2\_flag == 1 ){  update7SEG(index\_led);  index\_led++;  **if**(index\_led >=4) index\_led = 0;  updateLEDMatrix(index\_led\_matrix);  setTimer2(1000);  }  } |

Problem 9 ,10 : I the proteus not work with my set up and my code, I don’t know why, I will test it later.

I will answer more questions face to face. I am really sorry for this inconvenience. Because of the lack of time, I cannot make a full details report. I will answer every question face to face. Thanks you.