|  |
| --- |
| login.cs |
| using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Data.SqlClient;  using System.Drawing;  using System.Linq;  using System.Runtime.Remoting.Contexts;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  namespace mine\_project\_1  {  public partial class login : Form  {  game gfm = new game();  regit rfm = new regit();    SqlConnection sqlConn = new SqlConnection();  SqlCommand sqlCommand = new SqlCommand();  List<string> colName = new List<string>();  public login()  {  InitializeComponent();  }  public void Form1\_Load(object sender, EventArgs e)  {  sqlConn.ConnectionString = $"Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=K:\\Data\\[INTEL]AI SW Academy\\project\\mine\_project\_1\\Properties\\mine.mdf;Integrated Security=True;Connect Timeout=30";  sqlConn.Open();  sqlCommand.Connection = sqlConn;  label6.Text = "DB ON";  }  private void bt\_exit\_Click(object sender, EventArgs e)  {  Application.Exit();  }  public void bt\_login\_Click(object sender, EventArgs e)  {      bool db\_id = false;  bool db\_pass = false;    string sql = "select \* from user\_g";  SqlCommand cmd = new SqlCommand(sql, sqlConn);  SqlDataReader dr = cmd.ExecuteReader();  while (dr.Read())  {  if (dr["id"].ToString() == tb\_id.Text)  {  db\_id = true;  if (dr["pass"].ToString() == tb\_pass.Text)  {  db\_pass = true;  }  }  }  dr.Close();  if (db\_id == false)  {  MessageBox.Show("존재 하지 않는 성 명 이 다 !");  }  else if(db\_id == true && db\_pass == false)  {  MessageBox.Show("틀 린 암 구 호 !");  }  else if(db\_id == true && db\_pass == true)  {  MessageBox.Show("립 장 성 공");  gfm.Show();  gfm.label1.Text = tb\_id.Text;  this.Hide();    }  }  public void login\_FormClosing(object sender, FormClosingEventArgs e)  {  sqlConn.Close();  }  public void bt\_regit\_Click\_1(object sender, EventArgs e)  {  sqlConn.Close();  rfm.Show();  }  }  } |

|  |
| --- |
| regit.cs |
| using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Data.SqlClient;  using System.Drawing;  using System.Linq;  using System.Reflection.Emit;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  namespace mine\_project\_1  {  public partial class regit : Form  {  SqlConnection sqlConn = new SqlConnection();  SqlCommand sqlCommand = new SqlCommand();  List<string> colName = new List<string>();  public regit()  {  InitializeComponent();  }  private void bt\_login\_Click(object sender, EventArgs e)  {  if (tb\_id.Text == "")  {  MessageBox.Show("성명을 입력 하라");  return;  }  if (tb\_pass.Text == "")  {  MessageBox.Show("암구호를 입력 하라");  return;  }  string sql = $"insert into user\_g (id, pass) values ('{tb\_id.Text}','{tb\_pass.Text}')";  SqlCommand cmd = new SqlCommand(sql, sqlConn);  SqlDataReader dr = cmd.ExecuteReader();    dr.Close();  sqlConn.Close();  MessageBox.Show("합 류 성 공");  login lfm = new login();  this.Hide();  this.Close();  lfm.Show();  }  private void regit\_Load(object sender, EventArgs e)  {  sqlConn.ConnectionString = $"Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=K:\\Data\\[INTEL]AI SW Academy\\project\\mine\_project\_1\\Properties\\mine.mdf;Integrated Security=True;Connect Timeout=30";  sqlConn.Open();  sqlCommand.Connection = sqlConn;  label6.Text = "DB ON";  }  private void button1\_Click(object sender, EventArgs e)  {  login lfm = new login();  this.Hide();  this.Close();  lfm.ShowDialog();  }  }  } |

|  |
| --- |
| game.cs |
| using mine\_project\_1.Properties;  using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Data.SqlClient;  using System.Drawing;  using System.Linq;  using System.Reflection.Emit;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  using System.IO.Ports;  namespace mine\_project\_1  {  public partial class game : Form  {  rank kfm = new rank();  public class DoubleClickButton : Button  {  public DoubleClickButton() : base()  {  // Set the style so a double click event occurs.  SetStyle(ControlStyles.StandardClick |  ControlStyles.StandardDoubleClick, true);  }  }  public game()  {  InitializeComponent();  Width = 400;  Height = 550;  panel1.Top = 100;  panel1.Left = 0;  panel1.Width = Width;  panel1.Height = Height;    CreatePos();  ButtonCreate();  }  private void game\_FormClosing(object sender, FormClosingEventArgs e)  {  serialPort1.Close();  }  //선언    int[,] array = new int[10, 10];  int[,] Pos = new int[10, 10];  int[,] Visit = new int[10, 10];  int[,] Cnt = new int[10, 10];  DoubleClickButton[,] btnarr = new DoubleClickButton[10, 10];  bool GameOver;  private void CreatePos()  {  //throw new NotImplementedException();  int i, j;  Random r = new Random();  r.Next();  for (i = 0; i < 10; i++)  {  int y = r.Next(1, 10);  int x = r.Next(1, 10);  if (Pos[y, x] == 1)  {  i--;  continue;  }  Pos[y, x] = 1;  }  for (i = 0; i < 10; i++)  {  for (j = 0; j < 10; j++)  {  if (i > 0) Cnt[i, j] += Pos[i - 1, j];  if (i < 9) Cnt[i, j] += Pos[i + 1, j];  if (j > 0) Cnt[i, j] += Pos[i, j - 1];  if (j < 9) Cnt[i, j] += Pos[i, j + 1];  if (i > 0 && j > 0) Cnt[i, j] += Pos[i - 1, j - 1];  if (i > 0 && j < 9) Cnt[i, j] += Pos[i - 1, j + 1];  if (i < 9 && j > 0) Cnt[i, j] += Pos[i + 1, j - 1];  if (i < 9 && j < 9) Cnt[i, j] += Pos[i + 1, j + 1];  if (Pos[i, j] == 1) Cnt[i, j] = -1;  }  }  }  private void ButtonCreate()  {  int i, j;  int num = 1;  for (i = 0; i < 10; i++)  {  for (j = 0; j < 10; j++)  {  //private DoubleClickButton button1;  DoubleClickButton btn = new DoubleClickButton();  btn.Width = 40;  btn.Height = 40;  btn.Top = i \* 40;  btn.Left = j \* 40;  btn.Tag = num;  btn.BackColor = Color.Black;// button1.BackColor;  btn.BackgroundImage = Properties.Resources.pool;  array[i, j] = num;  //btn.Text = Cnt[i, j].ToString();  btn.Click += new EventHandler(btn\_Click);  //this.AllowDrop = true;  btn.DoubleClick += new EventHandler(btn\_DoubleClick);  btn.MouseDown += new MouseEventHandler(btn\_MouseDown);  num++;  btnarr[i, j] = btn;  panel1.Controls.Add(btn);  }  }  }  private void btn\_DoubleClick(object sender, EventArgs e)  {  // throw new NotImplementedException();  //this.FormBorderStyle = initialStyle;  if (GameOver) return;  int x, y;  int res = CheckVisit((sender as Button).Tag, out x, out y);  if (res == -1)  {  GameOver = true;  timer1.Stop();  MessageBox.Show("게임오버");  Sendmsg("RES 1");  }  else if (res == 1) return;  OpenButton(x, y);  if (GameCheck())  {  timer1.Stop();  string id = label1.Text;  int rank = progressBar1.Value;  Sendmsg("RES 2");  if (MessageBox.Show($"지뢰 제거 성공\n랭킹에 등록 하냐?\n아이디 = {id}, 점수 = {rank}", "ㅊㅋㅊㅋ", MessageBoxButtons.YesNo) == DialogResult.Yes)  {  SqlConnection sqlConn = new SqlConnection();  SqlCommand sqlCommand = new SqlCommand();  List<string> colName = new List<string>();  sqlConn.ConnectionString = $"Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=K:\\Data\\[INTEL]AI SW Academy\\project\\mine\_project\_1\\Properties\\mine.mdf;Integrated Security=True;Connect Timeout=30";  sqlConn.Open();  sqlCommand.Connection = sqlConn;  string sql = $"insert into rank (id, time) values ('{id}','{300-rank}')";  SqlCommand cmd = new SqlCommand(sql, sqlConn);  SqlDataReader dr = cmd.ExecuteReader();  dr.Close();  sqlConn.Close();  MessageBox.Show("성공");  }  }  }  private bool GameCheck()  {  int cnt = 0;  int i, j;  for (i = 0; i < 10; i++)  {  for (j = 0; j < 10; j++)  {  if (Visit[i, j] == 2) cnt++;  else if (Visit[i, j] == 0) return false;  }  }  if (cnt == 10) return true;  else return false;  }  private void OpenButton(int x, int y)  {  //throw new NotImplementedException();  if (x < 0 || y < 0 || x > 9 || y > 9) return;  if (Visit[y, x] != 0) return;  Visit[y, x] = 1;  btnarr[y, x].BackColor = Color.White;  btnarr[y, x].BackgroundImage = Properties.Resources.pool;  if (Cnt[y, x].ToString() == "1") btnarr[y, x].BackgroundImage = Properties.Resources.\_1;  else if (Cnt[y, x].ToString() == "2") btnarr[y, x].BackgroundImage = Properties.Resources.\_2;  else if (Cnt[y, x].ToString() == "3") btnarr[y, x].BackgroundImage = Properties.Resources.\_3;  else if (Cnt[y, x].ToString() == "4") btnarr[y, x].BackgroundImage = Properties.Resources.\_4;  else if (Cnt[y, x].ToString() == "0") btnarr[y, x].BackgroundImage = Properties.Resources.he;  else if (Cnt[y, x].ToString() == "-1") btnarr[y, x].BackgroundImage = Properties.Resources.boom;  else btnarr[y, x].BackgroundImage = Properties.Resources.he;  //btnarr[y, x].Text = Cnt[y, x].ToString();  if (Cnt[y, x] > 0) return;  OpenButton(x + 1, y);  OpenButton(x - 1, y);  OpenButton(x, y + 1);  OpenButton(x, y - 1);  OpenButton(x - 1, y - 1);  OpenButton(x - 1, y + 1);  OpenButton(x + 1, y - 1);  OpenButton(x + 1, y + 1);  }  private int CheckVisit(object tag, out int x, out int y)  {  int num = int.Parse(tag.ToString());  int i, j;  y = -1;  x = -1;  for (i = 0; i < 10; i++)  {  for (j = 0; j < 10; j++)  {  if (array[i, j] == num)  {  y = i;  x = j;  if (Cnt[i, j] == -1) return -1;  if (Visit[i, j] == 0) return 0;  else return 1;  }  }  }  return 1;  }  private void btn\_Click(object sender, EventArgs e)  {  //throw new NotImplementedException();  //this.FormBorderStyle = FormBorderStyle.FixedToolWindow;  }  private void Form1\_Resize(object sender, EventArgs e)  {  panel1.Top = 0;  panel1.Left = 0;  panel1.Width = Width;  panel1.Height = Height;  }  private void btn\_MouseDown(object sender, MouseEventArgs e)  {  if (GameOver) return;  //  if (e.Button == MouseButtons.Right)  {  //MessageBox.Show((sender as Button).Tag + "번 우클릭");  if ((sender as Button).BackColor == Color.Red)  {  (sender as Button).BackColor = Color.Black;  (sender as Button).BackgroundImage = Properties.Resources.pool;  }  else  {  (sender as Button).BackColor = Color.Red;  (sender as Button).BackgroundImage = Properties.Resources.flag;  }  RightClick((sender as Button).Tag);  }  }  private void RightClick(object tag)  {  int num = int.Parse(tag.ToString());  int i, j;  for (i = 0; i < 10; i++)  {  for (j = 0; j < 10; j++)  {  if (array[i, j] == num)  {  if (Visit[i, j] == 2) Visit[i, j] = 0;  else Visit[i, j] = 2;  return;  }  }  }  if (GameCheck())  {  timer1.Stop();  int rank = 300 - progressBar1.Value;  if (MessageBox.Show($"지뢰 제거 성공\n랭킹에 등록 하냐?\n점수 = {rank}", "ㅊㅋㅊㅋ", MessageBoxButtons.YesNo) == DialogResult.Yes)  {  MessageBox.Show("성공");  }      }  //MessageBox.Show(num.ToString());  }  private void button1\_Click(object sender, EventArgs e)  {  login lfm = new login();  kfm.Close();  this.Hide();  this.Close();  lfm.ShowDialog();    }  private void game\_Load(object sender, EventArgs e)  {  serialPort1.Parity = (Parity)0;  serialPort1.DataBits = 8;  serialPort1.StopBits = (StopBits)1;  serialPort1.BaudRate = 115200;  string[] Port = System.IO.Ports.SerialPort.GetPortNames();  serialPort1.PortName = Port[2];  string info\_con = $"{serialPort1.PortName} : {serialPort1.BaudRate} {serialPort1.Parity} {serialPort1.DataBits} {serialPort1.StopBits}";  serialPort1.Open();  if (serialPort1.IsOpen)  {  MessageBox.Show($"{serialPort1.PortName} OPEN");  }  kfm.Show();  timer1.Start();  }  private void Sendmsg(string text)  {  text = $"\x02{text}\x03";  serialPort1.Write(text);  }  private void timer1\_Tick(object sender, EventArgs e)  {  if (progressBar1.Value == 1)  {  GameOver = true;  timer1.Stop();  MessageBox.Show("타임 오바");  }  progressBar1.Value --;  }  }  } |

|  |
| --- |
| rank.cs |
| using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Data.SqlClient;  using System.Drawing;  using System.Linq;  using System.Reflection.Emit;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  namespace mine\_project\_1  {  public partial class rank : Form  {  public rank()  {  InitializeComponent();  }  SqlConnection sqlConn = new SqlConnection();  SqlCommand sqlCommand = new SqlCommand();  List<string> colName = new List<string>();  private void rank\_Load(object sender, EventArgs e)  {  textBox1.Text += "ID || RANK \r\n";  sqlConn.ConnectionString = $"Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=K:\\Data\\[INTEL]AI SW Academy\\project\\mine\_project\_1\\Properties\\mine.mdf;Integrated Security=True;Connect Timeout=30";  sqlConn.Open();  sqlCommand.Connection = sqlConn;  string sql = "select \* from rank order by 2";  sqlCommand.CommandText = sql;  SqlDataReader sr = sqlCommand.ExecuteReader();  int m = 0;  int s = 0;  for (int i = 0; sr.Read(); i++)  {  for (int j = 0; j < sr.FieldCount; j++)  {  if (j % 2 == 0)  textBox1.Text += sr.GetValue(j).ToString() + " ";  else  {  int t = Convert.ToInt32(sr.GetValue(j).ToString());  m = t / 60;  s = t % 60;  textBox1.Text += $"{m}분 {s}초";  }  if (j % 2 == 0) textBox1.Text += "|| ";  }  textBox1.Text += "\r\n";  }  sr.Close();  }  }  } |

|  |
| --- |
| stm src |
| /\* USER CODE BEGIN Header \*/  /\*\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* @file : main.c  \* @brief : Main program body  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* @attention  \*  \* Copyright (c) 2023 STMicroelectronics.  \* All rights reserved.  \*  \* This software is licensed under terms that can be found in the LICENSE file  \* in the root directory of this software component.  \* If no LICENSE file comes with this software, it is provided AS-IS.  \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \*/  /\* USER CODE END Header \*/  /\* Includes ------------------------------------------------------------------\*/  #include "main.h"  /\* Private includes ----------------------------------------------------------\*/  /\* USER CODE BEGIN Includes \*/  /\* USER CODE END Includes \*/  /\* Private typedef -----------------------------------------------------------\*/  /\* USER CODE BEGIN PTD \*/  #define RX\_BUF\_SIZE 100  /\* USER CODE END PTD \*/  /\* Private define ------------------------------------------------------------\*/  /\* USER CODE BEGIN PD \*/  char rxBuf2[RX\_BUF\_SIZE], rx; // UART2  char\* oparr[2];  int rxIdx2 = 0, start=0;  int op = -1; // 1 ~ 3 : NULL, CR, CRLF, LFCR  /\* USER CODE END PD \*/  /\* Private macro -------------------------------------------------------------\*/  /\* USER CODE BEGIN PM \*/  /\* USER CODE END PM \*/  /\* Private variables ---------------------------------------------------------\*/  TIM\_HandleTypeDef htim3;  UART\_HandleTypeDef huart2;  /\* USER CODE BEGIN PV \*/  int \_\_io\_putchar(int c)  {  HAL\_UART\_Transmit(&huart2, &c, 1, 10);  return c;  }  void checkoperation(char \*str)  {  printf("%s\r\n",str);  char\* token;  token = strtok(str, " ");  oparr[0] = token;  token = strtok(NULL, " ");  oparr[1] = token;  }  void HAL\_UART\_RxCpltCallback(UART\_HandleTypeDef \*huart)  {  if(rx == 0x02)  start = 1;  else if(rx == 0x03)  {  rxBuf2[rxIdx2++]='\0';  checkoperation(rxBuf2);  start=0;  rxIdx2=0;  }  if(start && rx!=0x02)  rxBuf2[rxIdx2++] = rx;  HAL\_UART\_Receive\_IT(&huart2, &rx, 1);  }  /\* USER CODE END PV \*/  /\* Private function prototypes -----------------------------------------------\*/  void SystemClock\_Config(void);  static void MX\_GPIO\_Init(void);  static void MX\_USART2\_UART\_Init(void);  static void MX\_TIM3\_Init(void);  /\* USER CODE BEGIN PFP \*/  /\* USER CODE END PFP \*/  /\* Private user code ---------------------------------------------------------\*/  /\* USER CODE BEGIN 0 \*/  void delay\_us(uint16\_t us)  {  \_\_HAL\_TIM\_SET\_COUNTER(&htim3,0);  while((\_\_HAL\_TIM\_GET\_COUNTER(&htim3)<us));  }  /\* USER CODE END 0 \*/  /\*\*  \* @brief The application entry point.  \* @retval int  \*/  int main(void)  {  /\* USER CODE BEGIN 1 \*/  /\* USER CODE END 1 \*/  /\* MCU Configuration--------------------------------------------------------\*/  /\* Reset of all peripherals, Initializes the Flash interface and the Systick. \*/  HAL\_Init();  /\* USER CODE BEGIN Init \*/  /\* USER CODE END Init \*/  /\* Configure the system clock \*/  SystemClock\_Config();  /\* USER CODE BEGIN SysInit \*/  /\* USER CODE END SysInit \*/  /\* Initialize all configured peripherals \*/  MX\_GPIO\_Init();  MX\_USART2\_UART\_Init();  MX\_TIM3\_Init();  /\* USER CODE BEGIN 2 \*/  HAL\_UART\_Receive\_IT(&huart2, &rx, 1);  HAL\_TIM\_Base\_Start(&htim3);  /\* USER CODE END 2 \*/  /\* Infinite loop \*/  /\* USER CODE BEGIN WHILE \*/  while (1)  {  /\* USER CODE END WHILE \*/  /\* USER CODE BEGIN 3 \*/  if(!HAL\_GPIO\_ReadPin(B1\_GPIO\_Port, B1\_Pin))  {  HAL\_GPIO\_WritePin(BUZ\_GPIO\_Port,BUZ\_Pin, 0);  HAL\_GPIO\_WritePin(fan\_GPIO\_Port,fan\_Pin, 0);  HAL\_GPIO\_WritePin(LED\_R\_GPIO\_Port,LED\_R\_Pin, 0);  HAL\_GPIO\_WritePin(LED\_G\_GPIO\_Port,LED\_G\_Pin, 0);  oparr[1] = "0";  printf("button pushed...............\r\n");  }  op = atoi(oparr[1]);  switch(op)  {  case 1: HAL\_GPIO\_WritePin(BUZ\_GPIO\_Port,BUZ\_Pin, 1);  HAL\_GPIO\_WritePin(fan\_GPIO\_Port, fan\_Pin, 1);  HAL\_GPIO\_WritePin(LED\_R\_GPIO\_Port,LED\_R\_Pin, 1);  break;  case 2: HAL\_GPIO\_WritePin(LED\_G\_GPIO\_Port,LED\_G\_Pin, 1);  HAL\_GPIO\_WritePin(LED\_R\_GPIO\_Port,LED\_R\_Pin, 1);  HAL\_GPIO\_WritePin(LED\_B\_GPIO\_Port,LED\_B\_Pin, 1);  HAL\_Delay(500);  HAL\_GPIO\_WritePin(LED\_G\_GPIO\_Port,LED\_G\_Pin, 0);  HAL\_GPIO\_WritePin(LED\_R\_GPIO\_Port,LED\_R\_Pin, 0);  HAL\_GPIO\_WritePin(LED\_B\_GPIO\_Port,LED\_B\_Pin, 0);  HAL\_Delay(500);  break;  case 3:  HAL\_GPIO\_WritePin(LED\_G\_GPIO\_Port,LED\_G\_Pin, 1);  HAL\_Delay(200);  HAL\_GPIO\_WritePin(LED\_R\_GPIO\_Port,LED\_R\_Pin, 1);  HAL\_Delay(200);  HAL\_GPIO\_WritePin(LED\_B\_GPIO\_Port,LED\_B\_Pin, 1);  HAL\_Delay(200);  HAL\_GPIO\_WritePin(LED\_G\_GPIO\_Port,LED\_G\_Pin, 0);  HAL\_Delay(200);  HAL\_GPIO\_WritePin(LED\_R\_GPIO\_Port,LED\_R\_Pin, 0);  HAL\_Delay(200);  HAL\_GPIO\_WritePin(LED\_B\_GPIO\_Port,LED\_B\_Pin, 0);  HAL\_Delay(200);  break;  default : break;  }  }  /\* USER CODE END 3 \*/  }  /\*\*  \* @brief System Clock Configuration  \* @retval None  \*/  void SystemClock\_Config(void)  {  RCC\_OscInitTypeDef RCC\_OscInitStruct = {0};  RCC\_ClkInitTypeDef RCC\_ClkInitStruct = {0};  /\*\* Configure the main internal regulator output voltage  \*/  \_\_HAL\_RCC\_PWR\_CLK\_ENABLE();  \_\_HAL\_PWR\_VOLTAGESCALING\_CONFIG(PWR\_REGULATOR\_VOLTAGE\_SCALE1);  /\*\* Initializes the RCC Oscillators according to the specified parameters  \* in the RCC\_OscInitTypeDef structure.  \*/  RCC\_OscInitStruct.OscillatorType = RCC\_OSCILLATORTYPE\_HSI;  RCC\_OscInitStruct.HSIState = RCC\_HSI\_ON;  RCC\_OscInitStruct.HSICalibrationValue = RCC\_HSICALIBRATION\_DEFAULT;  RCC\_OscInitStruct.PLL.PLLState = RCC\_PLL\_ON;  RCC\_OscInitStruct.PLL.PLLSource = RCC\_PLLSOURCE\_HSI;  RCC\_OscInitStruct.PLL.PLLM = 16;  RCC\_OscInitStruct.PLL.PLLN = 336;  RCC\_OscInitStruct.PLL.PLLP = RCC\_PLLP\_DIV4;  RCC\_OscInitStruct.PLL.PLLQ = 4;  if (HAL\_RCC\_OscConfig(&RCC\_OscInitStruct) != HAL\_OK)  {  Error\_Handler();  }  /\*\* Initializes the CPU, AHB and APB buses clocks  \*/  RCC\_ClkInitStruct.ClockType = RCC\_CLOCKTYPE\_HCLK|RCC\_CLOCKTYPE\_SYSCLK  |RCC\_CLOCKTYPE\_PCLK1|RCC\_CLOCKTYPE\_PCLK2;  RCC\_ClkInitStruct.SYSCLKSource = RCC\_SYSCLKSOURCE\_PLLCLK;  RCC\_ClkInitStruct.AHBCLKDivider = RCC\_SYSCLK\_DIV1;  RCC\_ClkInitStruct.APB1CLKDivider = RCC\_HCLK\_DIV2;  RCC\_ClkInitStruct.APB2CLKDivider = RCC\_HCLK\_DIV1;  if (HAL\_RCC\_ClockConfig(&RCC\_ClkInitStruct, FLASH\_LATENCY\_2) != HAL\_OK)  {  Error\_Handler();  }  }  /\*\*  \* @brief TIM3 Initialization Function  \* @param None  \* @retval None  \*/  static void MX\_TIM3\_Init(void)  {  /\* USER CODE BEGIN TIM3\_Init 0 \*/  /\* USER CODE END TIM3\_Init 0 \*/  TIM\_ClockConfigTypeDef sClockSourceConfig = {0};  TIM\_MasterConfigTypeDef sMasterConfig = {0};  /\* USER CODE BEGIN TIM3\_Init 1 \*/  /\* USER CODE END TIM3\_Init 1 \*/  htim3.Instance = TIM3;  htim3.Init.Prescaler = 84-1;  htim3.Init.CounterMode = TIM\_COUNTERMODE\_UP;  htim3.Init.Period = 65535;  htim3.Init.ClockDivision = TIM\_CLOCKDIVISION\_DIV1;  htim3.Init.AutoReloadPreload = TIM\_AUTORELOAD\_PRELOAD\_DISABLE;  if (HAL\_TIM\_Base\_Init(&htim3) != HAL\_OK)  {  Error\_Handler();  }  sClockSourceConfig.ClockSource = TIM\_CLOCKSOURCE\_INTERNAL;  if (HAL\_TIM\_ConfigClockSource(&htim3, &sClockSourceConfig) != HAL\_OK)  {  Error\_Handler();  }  sMasterConfig.MasterOutputTrigger = TIM\_TRGO\_RESET;  sMasterConfig.MasterSlaveMode = TIM\_MASTERSLAVEMODE\_DISABLE;  if (HAL\_TIMEx\_MasterConfigSynchronization(&htim3, &sMasterConfig) != HAL\_OK)  {  Error\_Handler();  }  /\* USER CODE BEGIN TIM3\_Init 2 \*/  /\* USER CODE END TIM3\_Init 2 \*/  }  /\*\*  \* @brief USART2 Initialization Function  \* @param None  \* @retval None  \*/  static void MX\_USART2\_UART\_Init(void)  {  /\* USER CODE BEGIN USART2\_Init 0 \*/  /\* USER CODE END USART2\_Init 0 \*/  /\* USER CODE BEGIN USART2\_Init 1 \*/  /\* USER CODE END USART2\_Init 1 \*/  huart2.Instance = USART2;  huart2.Init.BaudRate = 115200;  huart2.Init.WordLength = UART\_WORDLENGTH\_8B;  huart2.Init.StopBits = UART\_STOPBITS\_1;  huart2.Init.Parity = UART\_PARITY\_NONE;  huart2.Init.Mode = UART\_MODE\_TX\_RX;  huart2.Init.HwFlowCtl = UART\_HWCONTROL\_NONE;  huart2.Init.OverSampling = UART\_OVERSAMPLING\_16;  if (HAL\_UART\_Init(&huart2) != HAL\_OK)  {  Error\_Handler();  }  /\* USER CODE BEGIN USART2\_Init 2 \*/  /\* USER CODE END USART2\_Init 2 \*/  }  /\*\*  \* @brief GPIO Initialization Function  \* @param None  \* @retval None  \*/  static void MX\_GPIO\_Init(void)  {  GPIO\_InitTypeDef GPIO\_InitStruct = {0};  /\* USER CODE BEGIN MX\_GPIO\_Init\_1 \*/  /\* USER CODE END MX\_GPIO\_Init\_1 \*/  /\* GPIO Ports Clock Enable \*/  \_\_HAL\_RCC\_GPIOC\_CLK\_ENABLE();  \_\_HAL\_RCC\_GPIOH\_CLK\_ENABLE();  \_\_HAL\_RCC\_GPIOA\_CLK\_ENABLE();  \_\_HAL\_RCC\_GPIOB\_CLK\_ENABLE();  /\*Configure GPIO pin Output Level \*/  HAL\_GPIO\_WritePin(GPIOA, LD2\_Pin|LED\_B\_Pin|LED\_G\_Pin|LED\_R\_Pin, GPIO\_PIN\_RESET);  /\*Configure GPIO pin Output Level \*/  HAL\_GPIO\_WritePin(GPIOB, BUZ\_Pin|fan\_Pin, GPIO\_PIN\_RESET);  /\*Configure GPIO pin : B1\_Pin \*/  GPIO\_InitStruct.Pin = B1\_Pin;  GPIO\_InitStruct.Mode = GPIO\_MODE\_IT\_FALLING;  GPIO\_InitStruct.Pull = GPIO\_NOPULL;  HAL\_GPIO\_Init(B1\_GPIO\_Port, &GPIO\_InitStruct);  /\*Configure GPIO pins : LD2\_Pin LED\_B\_Pin LED\_G\_Pin LED\_R\_Pin \*/  GPIO\_InitStruct.Pin = LD2\_Pin|LED\_B\_Pin|LED\_G\_Pin|LED\_R\_Pin;  GPIO\_InitStruct.Mode = GPIO\_MODE\_OUTPUT\_PP;  GPIO\_InitStruct.Pull = GPIO\_NOPULL;  GPIO\_InitStruct.Speed = GPIO\_SPEED\_FREQ\_LOW;  HAL\_GPIO\_Init(GPIOA, &GPIO\_InitStruct);  /\*Configure GPIO pins : BUZ\_Pin fan\_Pin \*/  GPIO\_InitStruct.Pin = BUZ\_Pin|fan\_Pin;  GPIO\_InitStruct.Mode = GPIO\_MODE\_OUTPUT\_PP;  GPIO\_InitStruct.Pull = GPIO\_NOPULL;  GPIO\_InitStruct.Speed = GPIO\_SPEED\_FREQ\_LOW;  HAL\_GPIO\_Init(GPIOB, &GPIO\_InitStruct);  /\* USER CODE BEGIN MX\_GPIO\_Init\_2 \*/  /\* USER CODE END MX\_GPIO\_Init\_2 \*/  }  /\* USER CODE BEGIN 4 \*/  /\* USER CODE END 4 \*/  /\*\*  \* @brief This function is executed in case of error occurrence.  \* @retval None  \*/  void Error\_Handler(void)  {  /\* USER CODE BEGIN Error\_Handler\_Debug \*/  /\* User can add his own implementation to report the HAL error return state \*/  \_\_disable\_irq();  while (1)  {  }  /\* USER CODE END Error\_Handler\_Debug \*/  }  #ifdef USE\_FULL\_ASSERT  /\*\*  \* @brief Reports the name of the source file and the source line number  \* where the assert\_param error has occurred.  \* @param file: pointer to the source file name  \* @param line: assert\_param error line source number  \* @retval None  \*/  void assert\_failed(uint8\_t \*file, uint32\_t line)  {  /\* USER CODE BEGIN 6 \*/  /\* User can add his own implementation to report the file name and line number,  ex: printf("Wrong parameters value: file %s on line %d\r\n", file, line) \*/  /\* USER CODE END 6 \*/  }  #endif /\* USE\_FULL\_ASSERT \*/ |