|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное автономное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **Компьютерные системы и сети**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.04.01 Информатика и вычислительная техника**

МАГИСТЕРСКАЯ ПРОГРАММА **09.04.01/05 Современные интеллектуальные**

**программно-аппаратные комплексы**

Программно-аппаратная система «Беспроводная клавиатура»

Исходный текст программных модулей управления клавиатурой

Листов 10

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Студент | ИУ6-41М | |  |  | И.С. Марчук |
|  | (Группа) | |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  | |  |  |  |
| Руководитель ВКРМ | |  |  |  | С.В. Ибрагимов |
|  |  | |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  | |  |  |  |
|  |  | |  |  |  |
|  |  | |  |  |  |
|  |  | |  |  |  |

*2025 г.*

Базовый пример работы с дисплеем приведён в листинге В.1.

Листинг В.1 – Пример программы для работы с дисплеем

|  |
| --- |
| #include "libs/u8g2-master/ch58-u8g2.h"  #include "HidDevice.h"  #include "Keycodes.h"  #include "libs/sx1509/sx1509.h"  // данные дисплея  u8g2\_t u8g2;  uint8\_t TxBuff[] = "This is a tx example\r\n";  // Инициализация микроконтроллера  void initController()  {  // работа от внутреннего генератора, на плате есть внешний кварц на 32Мгц  SetSysClock(CLK\_SOURCE\_PLL\_60MHz);  // инициализация UART1  DebugInit();  PRINT("start\n");  pEP0\_RAM\_Addr = EP0\_Databuf;  pEP1\_RAM\_Addr = EP1\_Databuf;  pEP2\_RAM\_Addr = EP2\_Databuf;  pEP3\_RAM\_Addr = EP3\_Databuf;  USB\_DeviceInit();  PFIC\_EnableIRQ(USB\_IRQn);  }  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Главная функция  \*/  int main()  {  // Инициализация микроконтроллера  initController();  // Проверьте последовательный порт для отправки строки  UART1\_SendString(TxBuff, sizeof(TxBuff));  // uint8\_t abc1[] = "s1\r\n";  // UART1\_SendString(abc1, sizeof(abc1)); |

Продолжение листинга В.1

|  |
| --- |
| PRINT("----123---\r\n");  // ------------- инициализация переменной дисплея ---------  // Инициализация I2C будет использовать их как SDA/SCL (как в примере).  GPIOB\_ModeCfg(GPIO\_Pin\_13 | GPIO\_Pin\_12, GPIO\_ModeIN\_PU);  // инициализация данных в переменной дисплея  u8g2\_Setup\_sh1106\_i2c\_128x64\_noname\_f(  &u8g2,  U8G2\_R0,  u8x8\_byte\_ch58x\_hw\_i2c,  u8x8\_gpio\_and\_delay\_ch58x  );  // допроставление пинов в переменную дисплея (но они пустые так что может и не нужно)  u8x8\_SetPin\_HW\_I2C(  u8g2\_GetU8x8(&u8g2),  U8X8\_PIN\_NONE,  U8X8\_PIN\_NONE,  U8X8\_PIN\_NONE  );  // адрес дисплея  u8g2\_SetI2CAddress(&u8g2, 0x78); // SH1106 с адресом 0x78 (проверь в даташите)  // кисть инвертирует пиксели при наложении  u8g2\_SetDrawColor(&u8g2, 2);  // инициализация шрифта  u8g2\_SetFont(&u8g2, u8x8\_font\_amstrad\_cpc\_extended\_f);// 8g2\_font\_ncenB08\_tr u8x8\_font\_amstrad\_cpc\_extended\_f  u8g2\_SetFontMode(&u8g2, 1);  // ------------------------ работа с дисплеем ------------------------  начинаем взаимодействовать с дисплеем (отправка на него команд инициализации)  u8g2\_InitDisplay(&u8g2);  // отключение powersave  u8g2\_SetPowerSave(&u8g2, 0);  // установить яркость дисплея 0-255 (работает)  u8g2\_SetContrast(&u8g2, 30);  //u8g2\_ClearDisplay(&u8g2);// отправить на дисплей команды очистки |

Продолжение листинга В.1

|  |
| --- |
| mDelaymS(1000);  u8g2\_ClearBuffer(&u8g2);  u8g2\_DrawBox(&u8g2,0,0,32,64);  u8g2\_DrawStr(&u8g2, 20, 20, "Hello, CH582!");  u8g2\_SendBuffer(&u8g2);  // ------------------------ Главный цикл ------------------------  while (1)  {  u8g2\_ClearBuffer(&u8g2); // clear the internal memory  u8g2\_DrawBox(&u8g2,0,0,64,64);//0 0 128 64  u8g2\_DrawStr(&u8g2, 0, 12, "Hello, CH582!");  u8g2\_SendBuffer(&u8g2);//transferinternal memory to the display  }  } |

Базовый пример работы с драйвером клавиатуры SX1509 приведён в листинге В.2.

Листинг В.2 – Пример программы для работы с драйвером клавиатуры

|  |
| --- |
| #include "libs/u8g2-master/ch58-u8g2.h"  #include "HidDevice.h"  #include "Keycodes.h"  #include "libs/sx1509/sx1509.h"  uint8\_t TxBuff[] = "This is a tx example\r\n";  // Инициализация микроконтроллера  void initController()  {  // работа от внутреннего генератора, на плате есть внешний кварц на 32Мгц  SetSysClock(CLK\_SOURCE\_PLL\_60MHz);  // todo проставить для чего каждая строка  // инициализация UART1  DebugInit();  PRINT("start\n");  //PRINT("Start @ChipID=%02x\n", R8\_CHIP\_ID);  pEP0\_RAM\_Addr = EP0\_Databuf;  pEP1\_RAM\_Addr = EP1\_Databuf; |

Продолжение листинга В.2

|  |
| --- |
| pEP2\_RAM\_Addr = EP2\_Databuf;  pEP3\_RAM\_Addr = EP3\_Databuf;  PFIC\_EnableIRQ(USB\_IRQn);  }  /\*главная функция \*/  int main()  {  // Инициализация микроконтроллера  initController();  // PRINT("@ChipID=%02X\n", R8\_CHIP\_ID);    // Проверьте последовательный порт для отправки строки  UART1\_SendString(TxBuff, sizeof(TxBuff));  // переназначить пины I2C PB13/PB12 -> PB21/PB20  //GPIOPinRemap(ENABLE, RB\_PIN\_I2C);  // // Инициализация I2C будет использовать их как SDA/SCL.  GPIOB\_ModeCfg(GPIO\_Pin\_13 | GPIO\_Pin\_12, GPIO\_ModeIN\_PU);  I2C\_Init(  I2C\_Mode\_I2C,  100000, // 100 кГц // 400000, // 400 кГц  I2C\_DutyCycle\_2,  I2C\_Ack\_Disable,  I2C\_AckAddr\_7bit, // в библиотеке адресация 7 бит  0x42 // адрес ведущего  );  // прерывание поставим на B8  GPIOB\_ModeCfg(GPIO\_Pin\_8, GPIO\_ModeIN\_PU);  SX1509\_Init();  // ------------------------ Главный цикл ------------------------  while (1)  {  // если было прерывание  if (!GPIOB\_ReadPortPin(GPIO\_Pin\_8)) |

Продолжение листинга В.2

|  |
| --- |
| {  // Если нет прерываний, можно вручную обновлять:  SX1509\_UpdateKeyMatrix();  // после считывания флаг прерывания сбрасывается  for (uint8\_t i = 0; i < 8; i++)  {  PRINT("%d %d %d %d %d %d %d %d\n",  sx1509\_key\_matrix[i][0],  sx1509\_key\_matrix[i][1],  sx1509\_key\_matrix[i][2],  sx1509\_key\_matrix[i][3],  sx1509\_key\_matrix[i][4],  sx1509\_key\_matrix[i][5],  sx1509\_key\_matrix[i][6],  sx1509\_key\_matrix[i][7]);  }  PRINT("yes -------\n");  }  else  {  PRINT("no -------\n");  }  }  } |

Базовый пример работы со считыванием клавиш на прямую и отправка данных по USB приведены в листинге В.3.

Листинг В.3 – Код работы с клавиатурной матрицей напрямую и отправка нажатий клавиш по USB

|  |
| --- |
| #include "libs/u8g2-master/ch58-u8g2.h"  #include "HidDevice.h"  #include "Keycodes.h"  uint8\_t keysState[] = {  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}; |

Продолжение листинга В.3

|  |
| --- |
| uint8\_t TxBuff[] = "This is a tx example\r\n";  // Инициализация микроконтроллера  void initController()  {  // работа от внутреннего генератора, на плате есть внешний кварц на 32Мгц  SetSysClock(CLK\_SOURCE\_PLL\_60MHz);    // инициализация UART1  DebugInit();  PRINT("start\n");  //PRINT("Start @ChipID=%02x\n", R8\_CHIP\_ID);  pEP0\_RAM\_Addr = EP0\_Databuf;  pEP1\_RAM\_Addr = EP1\_Databuf;  pEP2\_RAM\_Addr = EP2\_Databuf;  pEP3\_RAM\_Addr = EP3\_Databuf;  USB\_DeviceInit();  PFIC\_EnableIRQ(USB\_IRQn);  }  /\* главная функция \*/  int main()  {  // Инициализация микроконтроллера  initController();  // ------------- инициализация direct клавиатуры -----  // чтение данных с клавиатуры  GPIOB\_ModeCfg(GPIO\_Pin\_0, GPIO\_ModeIN\_PU);  GPIOB\_ModeCfg(GPIO\_Pin\_1, GPIO\_ModeIN\_PU);  GPIOB\_ModeCfg(GPIO\_Pin\_2, GPIO\_ModeIN\_PU);  GPIOB\_ModeCfg(GPIO\_Pin\_3, GPIO\_ModeIN\_PU);  выход на матрицу клавиатуры  GPIOB\_ModeCfg(GPIO\_Pin\_4, GPIO\_ModeOut\_PP\_5mA);  GPIOB\_ModeCfg(GPIO\_Pin\_5, GPIO\_ModeOut\_PP\_5mA);  GPIOB\_ModeCfg(GPIO\_Pin\_6, GPIO\_ModeOut\_PP\_5mA);  GPIOB\_ModeCfg(GPIO\_Pin\_7, GPIO\_ModeOut\_PP\_5mA);  GPIOB\_ResetBits(GPIO\_Pin\_4);  GPIOB\_ResetBits(GPIO\_Pin\_5);  GPIOB\_ResetBits(GPIO\_Pin\_6);  GPIOB\_ResetBits(GPIO\_Pin\_7); |

Продолжение листинга В.3

|  |
| --- |
| // GPIOB\_SetBits(GPIO\_Pin\_4);  // mDelaymS(1000);  // GPIOB\_ResetBits(GPIO\_Pin\_4);  // mDelaymS(1000);  while (1)  {  for (uint8\_t row = 0; row < 3; row++)  {  switch (row)  {  case 0:  GPIOB\_SetBits(GPIO\_Pin\_4);  GPIOB\_ResetBits(GPIO\_Pin\_7);  break;  case 1:  GPIOB\_SetBits(GPIO\_Pin\_5);  GPIOB\_ResetBits(GPIO\_Pin\_4);  break;  case 2:  GPIOB\_SetBits(GPIO\_Pin\_6);  GPIOB\_ResetBits(GPIO\_Pin\_5);  break;  case 3:  GPIOB\_SetBits(GPIO\_Pin\_7);  GPIOB\_ResetBits(GPIO\_Pin\_6);  break;  }  mDelaymS(10);  //row << 2 +  if(row == 0){  keysState[ 0] = GPIOB\_ReadPortPin(GPIO\_Pin\_0);  keysState[ 1] = GPIOB\_ReadPortPin(GPIO\_Pin\_1);  keysState[ 2] = GPIOB\_ReadPortPin(GPIO\_Pin\_2);  keysState[ 3] = GPIOB\_ReadPortPin(GPIO\_Pin\_3);  }  }  if (keysState[0] == 0)  {DevHIDKeyReport(KEY\_W);  }  else if (keysState[1] == 0)  {  DevHIDKeyReport(KEY\_C);  } else if (keysState[2] == 0)  {  DevHIDKeyReport(KEY\_H); } |

Продолжение листинга В.3

|  |
| --- |
| else if (keysState[2] == 0)  {  DevHIDKeyReport(KEY\_H);  }  else if (keysState[3] == 0)  {  DevHIDKeyReport(KEY\_MINUS);  }  else  { DevHIDKeyReport(KEY\_RESERVED)}  //Левая кнопка мыши  DevHIDMouseReport(0x01);// нажать  //mDelaymS(100);  DevHIDMouseReport(0x00); // отпустить  //mDelaymS(200);  // Нажатие клавиш “wch”  DevHIDKeyReport(KEY\_W);  mDelaymS(100);  DevHIDKeyReport(KEY\_RESERVED);  mDelaymS(200);  DevHIDKeyReport(KEY\_C);  mDelaymS(100);  DevHIDKeyReport(KEY\_RESERVED);  mDelaymS(200);  DevHIDKeyReport(KEY\_MINUS);  mDelaymS(100);  DevHIDKeyReport(KEY\_RESERVED);  }  } |

Базовый пример работы с HID по Bluetooth приведен в листинге В.4.

Листинг В.4 – Код работы BLE HID (главная функция)

|  |
| --- |
| /\* Заголовочные файлы \*/  #include "config.h"  #include "HAL.h"  #include "hiddev.h"  #include "hidkbd.h"  #include "CH58xBLE\_LIB.h"  //#include "CH58xBLE\_ROM.h" |

Продолжение листинга В.4

|  |
| --- |
| /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* GLOBAL TYPEDEFS  \*/  \_\_attribute\_\_((aligned(4))) uint32\_t MEM\_BUF[BLE\_MEMHEAP\_SIZE / 4];  #if(defined(BLE\_MAC)) && (BLE\_MAC == TRUE)  const uint8\_t MacAddr[6] = {0x84, 0xC2, 0xE4, 0x03, 0x02, 0x02};  #endif  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* @fn Main\_Circulation  \*  \* @brief Основной цикл  \*/  \_\_HIGH\_CODE  \_\_attribute\_\_((noinline))  void Main\_Circulation()  {  while(1)  {  TMOS\_SystemProcess();  }  }  int main(void)  {  #if(defined(DCDC\_ENABLE)) && (DCDC\_ENABLE == TRUE)  PWR\_DCDCCfg(ENABLE);  #endif  SetSysClock(CLK\_SOURCE\_PLL\_60MHz);  #if(defined(HAL\_SLEEP)) && (HAL\_SLEEP == TRUE)  GPIOA\_ModeCfg(GPIO\_Pin\_All, GPIO\_ModeIN\_PU);  GPIOB\_ModeCfg(GPIO\_Pin\_All, GPIO\_ModeIN\_PU);  #endif  #ifdef DEBUG  GPIOA\_SetBits(bTXD1);  GPIOA\_ModeCfg(bTXD1, GPIO\_ModeOut\_PP\_5mA);  UART1\_DefInit();  #endif  //PRINT("%s\n", VER\_LIB);  CH58X\_BLEInit();  HAL\_Init();  GAPRole\_PeripheralInit();  HidDev\_Init();  HidEmu\_Init();  Main\_Circulation();  } |