

Date: 2025-08-13

dscClassic.cpp:105: Potential pointer usage: stream->println(F("ERROR: Failed to initialize ESP32 timer1 after retries"));

dscClassic.cpp:121: Potential pointer usage: if (stream) stream->println(F("ERROR: timer1 is null after initialization"));

dscClassic.cpp:892: Potential pointer usage: stream->print(F("Lights: "));

dscClassic.cpp:894: Potential pointer usage: if (bitRead(panelData[1], 7)) stream->print(F("Ready "));

dscClassic.cpp:895: Potential pointer usage: if (bitRead(panelData[1], 6)) stream->print(F("Armed "));

dscClassic.cpp:896: Potential pointer usage: if (bitRead(panelData[1], 5)) stream->print(F("Memory "));

dscClassic.cpp:897: Potential pointer usage: if (bitRead(panelData[1], 4)) stream->print(F("Bypass "));

dscClassic.cpp:898: Potential pointer usage: if (bitRead(panelData[1], 3)) stream->print(F("Trouble "));

dscClassic.cpp:899: Potential pointer usage: if (bitRead(panelData[1], 2)) stream->print(F("Program "));

dscClassic.cpp:900: Potential pointer usage: if (bitRead(panelData[1], 1)) stream->print(F("Fire "));

dscClassic.cpp:901: Potential pointer usage: if (bitRead(panelData[1], 0)) stream->print(F("Beep "));

dscClassic.cpp:903: Potential pointer usage: else stream->print(F("none "));

dscClassic.cpp:905: Potential pointer usage: stream->print(F("| Status: "));

dscClassic.cpp:907: Potential pointer usage: if (bitRead(pc16Data[1], 7)) stream->print(F("Trouble "));

dscClassic.cpp:908: Potential pointer usage: if (bitRead(pc16Data[1], 6)) stream->print(F("Bypassed zones "));

dscClassic.cpp:909: Potential pointer usage: if (bitRead(pc16Data[1], 5)) stream->print(F("Armed (side A) "));

dscClassic.cpp:910: Potential pointer usage: if (bitRead(pc16Data[1], 4)) stream->print(F("Armed (side B) "));

dscClassic.cpp:911: Potential pointer usage: if (bitRead(pc16Data[1], 3)) stream->print(F("Keypad Panic alarm "));

dscClassic.cpp:912: Potential pointer usage: if (bitRead(pc16Data[1], 2)) stream->print(F("Keypad Aux alarm "));

dscClassic.cpp:913: Potential pointer usage: if (bitRead(pc16Data[1], 1)) stream->print(F("Keypad Fire alarm "));

dscClassic.cpp:914: Potential pointer usage: if (bitRead(pc16Data[1], 0)) stream->print(F("Alarm "));

dscClassic.cpp:916: Potential pointer usage: else stream->print(F("none "));

dscClassic.cpp:918: Potential pointer usage: stream->print(F("| Zones open: "));

dscClassic.cpp:919: Potential pointer usage: if (panelData[0] == 0) stream->print(F("none "));

dscClassic.cpp:923: Potential pointer usage: stream->print(8 - bit);

dscClassic.cpp:924: Potential pointer usage: stream->print(F(" "));

dscClassic.cpp:930: Potential pointer usage: stream->print(F("| Zone alarm: "));

dscClassic.cpp:933: Potential pointer usage: stream->print(8 - bit);

dscClassic.cpp:934: Potential pointer usage: stream->print(F(" "));

dscClassic.cpp:940: Potential pointer usage: stream->print(F("| Fire alarm"));

dscClassic.cpp:949: Potential pointer usage: stream->print(F("[Keypad] "));

dscClassic.cpp:961: Arduino Serial usage: case 0xD7: Serial.print("[Digit]"); break; (replace with ESP_LOGx)

dscClassic.cpp:962: Arduino Serial usage: case 0xB7: Serial.print(""); break; (replace with ESP_LOGx)

dscClassic.cpp:963: Arduino Serial usage: case 0xE7: Serial.print("#"); break; (replace with ESP_LOGx)

dscClassic.cpp:964: Arduino Serial usage: case 0x3F: Serial.print(F("Fire alarm")); break; (replace with ESP_LOGx)

dscClassic.cpp:965: Arduino Serial usage: case 0x5F: Serial.print(F("Aux alarm")); break; (replace with ESP_LOGx)

dscClassic.cpp:966: Arduino Serial usage: case 0x6F: Serial.print(F("Panic alarm")); break; (replace with ESP_LOGx)

dscClassic.cpp:972: Arduino Serial usage: case 0xBE: Serial.print("1"); break; (replace with ESP_LOGx)

dscClassic.cpp:973: Arduino Serial usage: case 0xDE: Serial.print("2"); break; (replace with ESP_LOGx)

dscClassic.cpp:974: Arduino Serial usage: case 0xEE: Serial.print("3"); break; (replace with ESP_LOGx)

dscClassic.cpp:975: Arduino Serial usage: case 0xBD: Serial.print("4"); break; (replace with ESP_LOGx)

dscClassic.cpp:976: Arduino Serial usage: case 0xDD: Serial.print("5"); break; (replace with ESP_LOGx)

dscClassic.cpp:977: Arduino Serial usage: case 0xED: Serial.print("6"); break; (replace with ESP_LOGx)

dscClassic.cpp:978: Arduino Serial usage: case 0xBB: Serial.print("7"); break; (replace with ESP_LOGx)

dscClassic.cpp:979: Arduino Serial usage: case 0xDB: Serial.print("8"); break; (replace with ESP_LOGx)

dscClassic.cpp:980: Arduino Serial usage: case 0xEB: Serial.print("9"); break; (replace with ESP_LOGx)

dscClassic.cpp:981: Arduino Serial usage: case 0xD7: Serial.print("0"); break; (replace with ESP_LOGx)

dscClassic.cpp:982: Arduino Serial usage: case 0xB7: Serial.print(""); break; (replace with ESP_LOGx)

dscClassic.cpp:983: Arduino Serial usage: case 0xE7: Serial.print("#"); break; (replace with ESP_LOGx)

dscClassic.cpp:984: Arduino Serial usage: case 0x3F: Serial.print(F("Fire alarm")); break; (replace with ESP_LOGx)

dscClassic.cpp:985: Arduino Serial usage: case 0x5F: Serial.print(F("Aux alarm")); break; (replace with ESP_LOGx)

dscClassic.cpp:986: Arduino Serial usage: case 0x6F: Serial.print(F("Panic alarm")); break; (replace with ESP_LOGx)

dscClassic.cpp:996: Potential pointer usage: if (mask & panelData[panelByte]) stream->print("1");

dscClassic.cpp:997: Potential pointer usage: else stream->print("0");

dscClassic.cpp:999: Potential pointer usage: if (printSpaces && panelByte != panelByteCount - 1) stream->print(" ");

dscClassic.cpp:1002: Potential pointer usage: if (printSpaces) stream->print(" ");

dscClassic.cpp:1006: Potential pointer usage: if (mask & pc16Data[panelByte]) stream->print("1");

dscClassic.cpp:1007: Potential pointer usage: else stream->print("0");

dscClassic.cpp:1009: Potential pointer usage: if (printSpaces && panelByte != panelByteCount - 1) stream->print(" ");

dscClassic.cpp:1033: Potential pointer usage: if (hideKeypadDigits && keypadDigit && moduleByte == 0) stream->print(".....");

dscClassic.cpp:1036: Potential pointer usage: if (mask & moduleData[moduleByte]) stream->print("1");

dscClassic.cpp:1037: Potential pointer usage: else stream->print("0");

dscClassic.cpp:1040: Potential pointer usage: if (printSpaces && moduleByte != panelByteCount - 1) stream->print(" ");

dscClassic.cpp:1047: Potential pointer usage: stream->print(F("Panel"));

dscClassic.cpp:1220: Potential pointer usage: // Stores new panel data in the panel buffer

dscClassic.cpp:1232: Potential pointer usage: // Stores new keypad and module data - this data is not buffered

dscClassic.cpp:105: Pointer dereference without nearby null check: stream->println(F("ERROR: Failed to initialize ESP32 timer1 after retries"));

dscClassic.cpp:121: Pointer dereference without nearby null check: if (stream) stream->println(F("ERROR: timer1 is null after initialization"));

dscClassic.cpp:892: Pointer dereference without nearby null check: stream->print(F("Lights: "));

dscClassic.cpp:894: Pointer dereference without nearby null check: if (bitRead(panelData[1], 7)) stream->print(F("Ready "));

dscClassic.cpp:895: Pointer dereference without nearby null check: if (bitRead(panelData[1], 6)) stream->print(F("Armed "));

dscClassic.cpp:896: Pointer dereference without nearby null check: if (bitRead(panelData[1], 5)) stream->print(F("Memory "));

dscClassic.cpp:897: Pointer dereference without nearby null check: if (bitRead(panelData[1], 4)) stream->print(F("Bypass "));

dscClassic.cpp:898: Pointer dereference without nearby null check: if (bitRead(panelData[1], 3)) stream->print(F("Trouble "));

dscClassic.cpp:899: Pointer dereference without nearby null check: if (bitRead(panelData[1], 2)) stream->print(F("Program "));

dscClassic.cpp:900: Pointer dereference without nearby null check: if (bitRead(panelData[1], 1)) stream->print(F("Fire "));

dscClassic.cpp:901: Pointer dereference without nearby null check: if (bitRead(panelData[1], 0)) stream->print(F("Beep "));

dscClassic.cpp:903: Pointer dereference without nearby null check: else stream->print(F("none "));

dscClassic.cpp:905: Pointer dereference without nearby null check: stream->print(F("| Status: "));

dscClassic.cpp:907: Pointer dereference without nearby null check: if (bitRead(pc16Data[1], 7)) stream->print(F("Trouble "));

dscClassic.cpp:908: Pointer dereference without nearby null check: if (bitRead(pc16Data[1], 6)) stream->print(F("Bypassed zones "));

dscClassic.cpp:909: Pointer dereference without nearby null check: if (bitRead(pc16Data[1], 5)) stream->print(F("Armed (side A) "));

dscClassic.cpp:910: Pointer dereference without nearby null check: if (bitRead(pc16Data[1], 4)) stream->print(F("Armed (side B) "));

dscClassic.cpp:911: Pointer dereference without nearby null check: if (bitRead(pc16Data[1], 3)) stream->print(F("Keypad Panic alarm "));

dscClassic.cpp:912: Pointer dereference without nearby null check: if (bitRead(pc16Data[1], 2))

```

stream->print(F("Keypad Aux alarm "));
dscClassic.cpp:913: Pointer dereference without nearby null check: if (bitRead(pc16Data[1], 1))
stream->print(F("Keypad Fire alarm "));
dscClassic.cpp:914: Pointer dereference without nearby null check: if (bitRead(pc16Data[1], 0)) stream->print(F("Alarm
"));
dscClassic.cpp:916: Pointer dereference without nearby null check: else stream->print(F("none "));
dscClassic.cpp:918: Pointer dereference without nearby null check: stream->print(F("| Zones open: "));
dscClassic.cpp:919: Pointer dereference without nearby null check: if (panelData[0] == 0) stream->print(F("none "));
dscClassic.cpp:923: Pointer dereference without nearby null check: stream->print(8 - bit);
dscClassic.cpp:924: Pointer dereference without nearby null check: stream->print(F(" "));
dscClassic.cpp:930: Pointer dereference without nearby null check: stream->print(F("| Zone alarm: "));
dscClassic.cpp:933: Pointer dereference without nearby null check: stream->print(8 - bit);
dscClassic.cpp:934: Pointer dereference without nearby null check: stream->print(F(" "));
dscClassic.cpp:940: Pointer dereference without nearby null check: stream->print(F("| Fire alarm"));
dscClassic.cpp:949: Pointer dereference without nearby null check: stream->print(F("[Keypad] "));
dscClassic.cpp:996: Pointer dereference without nearby null check: if (mask & panelData[panelByte]) stream->print("1");
dscClassic.cpp:997: Pointer dereference without nearby null check: else stream->print("0");
dscClassic.cpp:999: Pointer dereference without nearby null check: if (printSpaces && panelByte != panelByteCount - 1)
stream->print(" ");
dscClassic.cpp:1002: Pointer dereference without nearby null check: if (printSpaces) stream->print(" ");
dscClassic.cpp:1006: Pointer dereference without nearby null check: if (mask & pc16Data[panelByte])
stream->print("1");
dscClassic.cpp:1007: Pointer dereference without nearby null check: else stream->print("0");
dscClassic.cpp:1009: Pointer dereference without nearby null check: if (printSpaces && panelByte != panelByteCount -
1) stream->print(" ");
dscClassic.cpp:1033: Pointer dereference without nearby null check: if (hideKeypadDigits && keypadDigit &&
moduleByte == 0) stream->print(".....");
dscClassic.cpp:1036: Pointer dereference without nearby null check: if (mask & moduleData[moduleByte])
stream->print("1");
dscClassic.cpp:1037: Pointer dereference without nearby null check: else stream->print("0");
dscClassic.cpp:1040: Pointer dereference without nearby null check: if (printSpaces && moduleByte != panelByteCount -
1) stream->print(" ");
dscClassic.cpp:1047: Pointer dereference without nearby null check: stream->print(F("Panel"));
dscClassic.h:30: Potential pointer usage: dscClassicInterface(byte setClockPin, byte setReadPin, byte setPC16Pin, byte
setWritePin = 255, const char * setAccessCode = "");
dscClassic.h:33: Arduino Serial usage: void begin(Stream &_stream = Serial); // Initializes the stream output to
Serial by default (replace with ESP_LOGx)
dscClassic.h:146: Potential pointer usage: static hw_timer_t * timer1;
dscClassic.h:162: Potential pointer usage: Stream* stream;
dscClassic.h:163: Potential pointer usage: const char * writeKeysArray;
dscClassic.h:164: Potential pointer usage: const char * accessCodeStay;
dscClassicKeypad.h:12: Arduino Serial usage: void begin(Stream &_stream = Serial); // Initializes the
stream output to Serial by default (replace with ESP_LOGx)
dscClassicKeypad.h:39: Potential pointer usage: Stream* stream;
dscClassicKeypad.h:68: Potential pointer usage: static hw_timer_t * timer1;
dscKeybus.h:33: Arduino Serial usage: void begin(Stream &_stream = Serial); // Initializes the stream output to
Serial by default (replace with ESP_LOGx)
dscKeybus.h:275: Potential pointer usage: static hw_timer_t * timer1;
dscKeybus.h:291: Potential pointer usage: Stream* stream;
dscKeybus.h:292: Potential pointer usage: const char* writeKeysArray;
dscKeybusInterface.cpp:107: Potential pointer usage: stream->println(F("ERROR: Failed to initialize ESP32 timer1 after
retries"));
dscKeybusInterface.cpp:125: Potential pointer usage: if (stream) stream->println(F("ERROR: timer1 is null after

```

initialization"));

dscKeybusInterface.cpp:620: Potential pointer usage: // Stores new panel data in the panel buffer

dscKeybusInterface.cpp:632: Potential pointer usage: // Stores new keypad and module data - this data is not buffered

dscKeybusInterface.cpp:107: Pointer dereference without nearby null check: stream->println(F("ERROR: Failed to initialize ESP32 timer1 after retries"));

dscKeybusInterface.cpp:125: Pointer dereference without nearby null check: if (stream) stream->println(F("ERROR: timer1 is null after initialization"));

dscKeybusInterface_minimal.h:65: Potential pointer usage: extern dscKeybusInterfaceMinimal* dscKeybusInstance;

dscKeybusPrintData.cpp:68: Potential pointer usage: stream->print(F("[CRC Error]"));

dscKeybusPrintData.cpp:120: Potential pointer usage: stream->print("Unknown data");

dscKeybusPrintData.cpp:135: Potential pointer usage: stream->print(F("[Module/0x"]));

dscKeybusPrintData.cpp:136: Potential pointer usage: if (moduleCmd < 16) stream->print("0");

dscKeybusPrintData.cpp:137: Potential pointer usage: stream->print(moduleCmd, HEX);

dscKeybusPrintData.cpp:140: Potential pointer usage: stream->print(".");

dscKeybusPrintData.cpp:141: Potential pointer usage: if (moduleSubCmd < 16) stream->print("0");

dscKeybusPrintData.cpp:142: Potential pointer usage: stream->print(moduleSubCmd, HEX);

dscKeybusPrintData.cpp:144: Potential pointer usage: stream->print(F("] "));

dscKeybusPrintData.cpp:183: Potential pointer usage: default: stream->print("Unknown data");

dscKeybusPrintData.cpp:196: Potential pointer usage: if (panelData[panelByte] == 0) stream->print(F("none "));

dscKeybusPrintData.cpp:198: Potential pointer usage: if (bitRead(panelData[panelByte], 0)) stream->print(F("Ready "));

dscKeybusPrintData.cpp:199: Potential pointer usage: if (bitRead(panelData[panelByte], 1)) stream->print(F("Armed "));

dscKeybusPrintData.cpp:200: Potential pointer usage: if (bitRead(panelData[panelByte], 2)) stream->print(F("Memory "));

dscKeybusPrintData.cpp:201: Potential pointer usage: if (bitRead(panelData[panelByte], 3)) stream->print(F("Bypass "));

dscKeybusPrintData.cpp:202: Potential pointer usage: if (bitRead(panelData[panelByte], 4)) stream->print(F("Trouble "));

dscKeybusPrintData.cpp:203: Potential pointer usage: if (bitRead(panelData[panelByte], 5)) stream->print(F("Program "));

dscKeybusPrintData.cpp:204: Potential pointer usage: if (bitRead(panelData[panelByte], 6)) stream->print(F("Fire "));

dscKeybusPrintData.cpp:205: Potential pointer usage: if (bitRead(panelData[panelByte], 7)) stream->print(F("Backlight "));

dscKeybusPrintData.cpp:209: Potential pointer usage: stream->print(F("- "));

dscKeybusPrintData.cpp:222: Potential pointer usage: case 0x01: stream->print(F("Partition ready")); break;

dscKeybusPrintData.cpp:223: Potential pointer usage: case 0x02: stream->print(F("Stay zones open")); break;

dscKeybusPrintData.cpp:224: Potential pointer usage: case 0x03: stream->print(F("Zones open")); break;

dscKeybusPrintData.cpp:225: Potential pointer usage: case 0x04: stream->print(F("Armed: Stay")); break;

dscKeybusPrintData.cpp:226: Potential pointer usage: case 0x05: stream->print(F("Armed: Away")); break;

dscKeybusPrintData.cpp:227: Potential pointer usage: case 0x06: stream->print(F("Armed: Stay with no entry delay")); break;

dscKeybusPrintData.cpp:228: Potential pointer usage: case 0x07: stream->print(F("Failed to arm")); break;

dscKeybusPrintData.cpp:229: Potential pointer usage: case 0x08: stream->print(F("Exit delay in progress")); break;

dscKeybusPrintData.cpp:230: Potential pointer usage: case 0x09: stream->print(F("Arming: No entry delay")); break;

dscKeybusPrintData.cpp:231: Potential pointer usage: case 0x0B: stream->print(F("Quick exit in progress")); break;

dscKeybusPrintData.cpp:232: Potential pointer usage: case 0x0C: stream->print(F("Entry delay in progress")); break;

dscKeybusPrintData.cpp:233: Potential pointer usage: case 0x0D: stream->print(F("Entry delay after alarm")); break;

dscKeybusPrintData.cpp:234: Potential pointer usage: case 0x0E: stream->print(F("Function not available")); break;

dscKeybusPrintData.cpp:235: Potential pointer usage: case 0x10: stream->print(F("Keypad lockout")); break;

dscKeybusPrintData.cpp:236: Potential pointer usage: case 0x11: stream->print(F("Partition in alarm")); break;

dscKeybusPrintData.cpp:237: Potential pointer usage: case 0x12: stream->print(F("Battery check in progress")); break;

dscKeybusPrintData.cpp:238: Potential pointer usage: case 0x14: stream->print(F("Auto-arm in progress")); break;

dscKeybusPrintData.cpp:239: Potential pointer usage: case 0x15: stream->print(F("Arming with bypassed zones")); break;

dscKeybusPrintData.cpp:240: Potential pointer usage: case 0x16: stream->print(F("Armed: Away with no entry delay")); break;

dscKeybusPrintData.cpp:241: Potential pointer usage: case 0x19: stream->print(F("Disarmed: Alarm memory")); break;
dscKeybusPrintData.cpp:242: Potential pointer usage: case 0x22: stream->print(F("Disarmed: Recent closing")); break;
dscKeybusPrintData.cpp:243: Potential pointer usage: case 0x2F: stream->print(F("Keypad LCD test")); break;
dscKeybusPrintData.cpp:244: Potential pointer usage: case 0x33: stream->print(F("Command output in progress")); break;
dscKeybusPrintData.cpp:245: Potential pointer usage: case 0x3D: stream->print(F("Disarmed: Alarm memory")); break;
dscKeybusPrintData.cpp:246: Potential pointer usage: case 0x3E: stream->print(F("Partition disarmed")); break;
dscKeybusPrintData.cpp:248: Potential pointer usage: case 0x40: stream->print(F("Keypad blanking")); break;
dscKeybusPrintData.cpp:249: Potential pointer usage: case 0x8A: stream->print(F("Activate stay/away zones")); break;
dscKeybusPrintData.cpp:250: Potential pointer usage: case 0x8B: stream->print(F("Quick exit")); break;
dscKeybusPrintData.cpp:251: Potential pointer usage: case 0x8E: stream->print(F("Function not available")); break;
dscKeybusPrintData.cpp:252: Potential pointer usage: case 0x8F: stream->print(F("Invalid access code")); break;
dscKeybusPrintData.cpp:253: Potential pointer usage: case 0x9E: stream->print(F("Enter * function key")); break;
dscKeybusPrintData.cpp:254: Potential pointer usage: case 0x9F: stream->print(F("Enter access code")); break;
dscKeybusPrintData.cpp:255: Potential pointer usage: case 0xA0: stream->print(F("**1: Zone bypass")); break;
dscKeybusPrintData.cpp:256: Potential pointer usage: case 0xA1: stream->print(F("**2: Trouble")); break;
dscKeybusPrintData.cpp:257: Potential pointer usage: case 0xA2: stream->print(F("**3: Alarm memory")); break;
dscKeybusPrintData.cpp:258: Potential pointer usage: case 0xA3: stream->print(F("Door chime enabled")); break;
dscKeybusPrintData.cpp:259: Potential pointer usage: case 0xA4: stream->print(F("Door chime disabled")); break;
dscKeybusPrintData.cpp:260: Potential pointer usage: case 0xA5: stream->print(F("Enter master code")); break;
dscKeybusPrintData.cpp:261: Potential pointer usage: case 0xA6: stream->print(F("**5: Access codes")); break;
dscKeybusPrintData.cpp:262: Potential pointer usage: case 0xA7: stream->print(F("**5: Enter 4-digit code")); break;
dscKeybusPrintData.cpp:263: Potential pointer usage: case 0xA9: stream->print(F("**6: User functions")); break;
dscKeybusPrintData.cpp:264: Potential pointer usage: case 0xAA: stream->print(F("**6: Time and date")); break;
dscKeybusPrintData.cpp:265: Potential pointer usage: case 0xAB: stream->print(F("**6: Auto-arm time")); break;
dscKeybusPrintData.cpp:266: Potential pointer usage: case 0xAC: stream->print(F("**6: Auto-arm enabled")); break;
dscKeybusPrintData.cpp:267: Potential pointer usage: case 0xAD: stream->print(F("**6: Auto-arm disabled")); break;
dscKeybusPrintData.cpp:268: Potential pointer usage: case 0xAF: stream->print(F("**6: System test")); break;
dscKeybusPrintData.cpp:269: Potential pointer usage: case 0xB0: stream->print(F("**6: Enable DLS")); break;
dscKeybusPrintData.cpp:271: Potential pointer usage: case 0xB3: stream->print(F("**7: Command output")); break;
dscKeybusPrintData.cpp:272: Potential pointer usage: case 0xB7: stream->print(F("Enter installer code")); break;
dscKeybusPrintData.cpp:273: Potential pointer usage: case 0xB8: stream->print(F("Enter * function key while armed")); break;
dscKeybusPrintData.cpp:274: Potential pointer usage: case 0xB9: stream->print(F("**2: Zone tamper menu")); break;
dscKeybusPrintData.cpp:275: Potential pointer usage: case 0xBA: stream->print(F("**2: Zones with low batteries")); break;
dscKeybusPrintData.cpp:276: Potential pointer usage: case 0xBC: stream->print(F("**5: Enter 6-digit code")); break;
dscKeybusPrintData.cpp:277: Potential pointer usage: case 0xBF: stream->print(F("**6: Auto-arm select day")); break;
dscKeybusPrintData.cpp:278: Potential pointer usage: case 0xC6: stream->print(F("**2: Zone fault menu")); break;
dscKeybusPrintData.cpp:279: Potential pointer usage: //case 0xC7: stream->print(F("Partition not available")); break;
dscKeybusPrintData.cpp:280: Potential pointer usage: case 0xC8: stream->print(F("**2: Service required menu")); break;
dscKeybusPrintData.cpp:281: Potential pointer usage: case 0xCD: stream->print(F("Downloading in progress")); break;
dscKeybusPrintData.cpp:282: Potential pointer usage: case 0xCE: stream->print(F("Active camera monitor selection")); break;
dscKeybusPrintData.cpp:283: Potential pointer usage: case 0xD0: stream->print(F("**2: Keypads with low batteries")); break;
dscKeybusPrintData.cpp:284: Potential pointer usage: case 0xD1: stream->print(F("**2: Keyfobs with low batteries")); break;
dscKeybusPrintData.cpp:285: Potential pointer usage: case 0xD4: stream->print(F("**2: Zones with RF Delinquency")); break;
dscKeybusPrintData.cpp:286: Potential pointer usage: case 0xE4: stream->print(F("**8: Installer programming, 3 digits")); decimalInput = false; break;
dscKeybusPrintData.cpp:287: Potential pointer usage: case 0xE5: stream->print(F("Keypad slot assignment")); break;

dscKeybusPrintData.cpp:288: Potential pointer usage: case 0xE6: stream->print(F("Input: 2 digits")); break;

dscKeybusPrintData.cpp:289: Potential pointer usage: case 0xE7: stream->print(F("Input: 3 digits")); decimalInput = true; break;

dscKeybusPrintData.cpp:290: Potential pointer usage: case 0xE8: stream->print(F("Input: 4 digits")); break;

dscKeybusPrintData.cpp:291: Potential pointer usage: case 0xE9: stream->print(F("Input: 5 digits")); break;

dscKeybusPrintData.cpp:292: Potential pointer usage: case 0xEA: stream->print(F("Input HEX: 2 digits")); break;

dscKeybusPrintData.cpp:293: Potential pointer usage: case 0xEB: stream->print(F("Input HEX: 4 digits")); break;

dscKeybusPrintData.cpp:294: Potential pointer usage: case 0xEC: stream->print(F("Input HEX: 6 digits")); break;

dscKeybusPrintData.cpp:295: Potential pointer usage: case 0xED: stream->print(F("Input HEX: 32 digits")); break;

dscKeybusPrintData.cpp:296: Potential pointer usage: case 0xEE: stream->print(F("Input: 1 option per zone")); break;

dscKeybusPrintData.cpp:297: Potential pointer usage: case 0xEF: stream->print(F("Module supervision field")); break;

dscKeybusPrintData.cpp:298: Potential pointer usage: case 0xF0: stream->print(F("Function key 1")); break;

dscKeybusPrintData.cpp:299: Potential pointer usage: case 0xF1: stream->print(F("Function key 2")); break;

dscKeybusPrintData.cpp:300: Potential pointer usage: case 0xF2: stream->print(F("Function key 3")); break;

dscKeybusPrintData.cpp:301: Potential pointer usage: case 0xF3: stream->print(F("Function key 4")); break;

dscKeybusPrintData.cpp:302: Potential pointer usage: case 0xF4: stream->print(F("Function key 5")); break;

dscKeybusPrintData.cpp:303: Potential pointer usage: case 0xF5: stream->print(F("Wireless module placement test")); break;

dscKeybusPrintData.cpp:304: Potential pointer usage: case 0xF6: stream->print(F("Activate device for test")); break;

dscKeybusPrintData.cpp:305: Potential pointer usage: case 0xF7: stream->print(F("**8: Installer programming, 2 digits")); decimalInput = false; break;

dscKeybusPrintData.cpp:306: Potential pointer usage: case 0xF8: stream->print(F("Keypad programming")); break;

dscKeybusPrintData.cpp:307: Potential pointer usage: case 0xFA: stream->print(F("Input: 6 digits")); break;

dscKeybusPrintData.cpp:309: Potential pointer usage: stream->print(F("Unknown data: 0x"));

dscKeybusPrintData.cpp:310: Potential pointer usage: if (panelData[panelByte] < 10) stream->print("0");

dscKeybusPrintData.cpp:311: Potential pointer usage: stream->print(panelData[panelByte], HEX);

dscKeybusPrintData.cpp:362: Potential pointer usage: case 0x49: stream->print(F("Duress alarm")); break;

dscKeybusPrintData.cpp:363: Potential pointer usage: case 0x4A: stream->print(F("Disarmed: Alarm memory")); break;

dscKeybusPrintData.cpp:364: Potential pointer usage: case 0x4B: stream->print(F("Recent closing alarm")); break;

dscKeybusPrintData.cpp:365: Potential pointer usage: case 0x4C: stream->print(F("Zone expander supervisory alarm")); break;

dscKeybusPrintData.cpp:366: Potential pointer usage: case 0x4D: stream->print(F("Zone expander supervisory restored")); break;

dscKeybusPrintData.cpp:367: Potential pointer usage: case 0x4E: stream->print(F("Keypad Fire alarm")); break;

dscKeybusPrintData.cpp:368: Potential pointer usage: case 0x4F: stream->print(F("Keypad Aux alarm")); break;

dscKeybusPrintData.cpp:369: Potential pointer usage: case 0x50: stream->print(F("Keypad Panic alarm")); break;

dscKeybusPrintData.cpp:370: Potential pointer usage: case 0x51: stream->print(F("PGM2 input alarm")); break;

dscKeybusPrintData.cpp:371: Potential pointer usage: case 0x52: stream->print(F("Keypad Fire alarm restored")); break;

dscKeybusPrintData.cpp:372: Potential pointer usage: case 0x53: stream->print(F("Keypad Aux alarm restored")); break;

dscKeybusPrintData.cpp:373: Potential pointer usage: case 0x54: stream->print(F("Keypad Panic alarm restored")); break;

dscKeybusPrintData.cpp:374: Potential pointer usage: case 0x55: stream->print(F("PGM2 input alarm restored")); break;

dscKeybusPrintData.cpp:377: Potential pointer usage: case 0x98: stream->print(F("Keypad lockout")); break;

dscKeybusPrintData.cpp:379: Potential pointer usage: case 0xBE: stream->print(F("Armed: Partial")); break;

dscKeybusPrintData.cpp:380: Potential pointer usage: case 0xBF: stream->print(F("Armed: Special")); break;

dscKeybusPrintData.cpp:382: Potential pointer usage: case 0xE5: stream->print(F("Auto-arm cancelled")); break;

dscKeybusPrintData.cpp:383: Potential pointer usage: case 0xE6: stream->print(F("Disarmed: Special")); break;

dscKeybusPrintData.cpp:384: Potential pointer usage: case 0xE7: stream->print(F("Panel battery trouble")); break;

dscKeybusPrintData.cpp:385: Potential pointer usage: case 0xE8: stream->print(F("Panel AC power trouble")); break;

dscKeybusPrintData.cpp:386: Potential pointer usage: case 0xE9: stream->print(F("Bell trouble")); break;

dscKeybusPrintData.cpp:387: Potential pointer usage: case 0xEA: stream->print(F("Fire zone trouble")); break;

dscKeybusPrintData.cpp:388: Potential pointer usage: case 0xEB: stream->print(F("Panel aux supply trouble")); break;
dscKeybusPrintData.cpp:389: Potential pointer usage: case 0xEC: stream->print(F("Telephone line trouble")); break;
dscKeybusPrintData.cpp:390: Potential pointer usage: case 0xEF: stream->print(F("Panel battery restored")); break;
dscKeybusPrintData.cpp:391: Potential pointer usage: case 0xF0: stream->print(F("Panel AC power restored")); break;
dscKeybusPrintData.cpp:392: Potential pointer usage: case 0xF1: stream->print(F("Bell restored")); break;
dscKeybusPrintData.cpp:393: Potential pointer usage: case 0xF2: stream->print(F("Fire zone trouble restored")); break;
dscKeybusPrintData.cpp:394: Potential pointer usage: case 0xF3: stream->print(F("Panel aux supply restored")); break;
dscKeybusPrintData.cpp:395: Potential pointer usage: case 0xF4: stream->print(F("Telephone line restored")); break;
dscKeybusPrintData.cpp:396: Potential pointer usage: case 0xF7: stream->print(F("Phone 1 FTC")); break;
dscKeybusPrintData.cpp:397: Potential pointer usage: case 0xF8: stream->print(F("Phone 2 FTC")); break;
dscKeybusPrintData.cpp:398: Potential pointer usage: case 0xF9: stream->print(F("Event buffer threshold")); break;
//75% full since last DLS upload
dscKeybusPrintData.cpp:399: Potential pointer usage: case 0xFA: stream->print(F("DLS lead-in")); break;
dscKeybusPrintData.cpp:400: Potential pointer usage: case 0xFB: stream->print(F("DLS lead-out")); break;
dscKeybusPrintData.cpp:401: Potential pointer usage: case 0xFE: stream->print(F("Periodic test transmission")); break;
dscKeybusPrintData.cpp:402: Potential pointer usage: case 0xFF: stream->print(F("System test")); break;
dscKeybusPrintData.cpp:417: Potential pointer usage: stream->print("Zone alarm: ");
dscKeybusPrintData.cpp:432: Potential pointer usage: stream->print("Zone alarm restored: ");
dscKeybusPrintData.cpp:447: Potential pointer usage: stream->print("Zone tamper: ");
dscKeybusPrintData.cpp:461: Potential pointer usage: stream->print("Zone tamper restored: ");
dscKeybusPrintData.cpp:476: Potential pointer usage: stream->print(F("Armed: "));
dscKeybusPrintData.cpp:491: Potential pointer usage: stream->print(F("Disarmed: "));
dscKeybusPrintData.cpp:496: Potential pointer usage: stream->print("Unknown data");
dscKeybusPrintData.cpp:527: Potential pointer usage: case 0x03: stream->print(F("Cross zone alarm")); return;
dscKeybusPrintData.cpp:528: Potential pointer usage: case 0x04: stream->print(F("Delinquency alarm")); return;
dscKeybusPrintData.cpp:529: Potential pointer usage: case 0x05: stream->print(F("Late to close")); return;
dscKeybusPrintData.cpp:531: Potential pointer usage: case 0x29: stream->print(F("Downloading forced answer"));
return;
dscKeybusPrintData.cpp:532: Potential pointer usage: case 0x2B: stream->print(F("Armed: Auto-arm")); return;
dscKeybusPrintData.cpp:537: Potential pointer usage: case 0xAC: stream->print(F("Exit installer programming")); return;
dscKeybusPrintData.cpp:538: Potential pointer usage: case 0xAD: stream->print(F("Enter installer programming"));
return;
dscKeybusPrintData.cpp:539: Potential pointer usage: case 0xAE: stream->print(F("Walk test end")); return;
dscKeybusPrintData.cpp:540: Potential pointer usage: case 0xAF: stream->print(F("Walk test begin")); return;
dscKeybusPrintData.cpp:542: Potential pointer usage: case 0xD0: stream->print(F("Command output 4")); return;
dscKeybusPrintData.cpp:543: Potential pointer usage: case 0xD1: stream->print(F("Exit fault pre-alert")); return;
dscKeybusPrintData.cpp:544: Potential pointer usage: case 0xD2: stream->print(F("Armed: Entry delay")); return;
dscKeybusPrintData.cpp:545: Potential pointer usage: case 0xD3: stream->print(F("Downlook remote trigger")); return;
dscKeybusPrintData.cpp:569: Potential pointer usage: stream->print(F("Zone battery restored: "));
dscKeybusPrintData.cpp:578: Potential pointer usage: stream->print(F("Zone battery low: "));
dscKeybusPrintData.cpp:594: Potential pointer usage: stream->print(F("Zone fault restored: "));
dscKeybusPrintData.cpp:610: Potential pointer usage: stream->print(F("Zone fault: "));
dscKeybusPrintData.cpp:624: Potential pointer usage: stream->print(F("Zone bypassed: "));
dscKeybusPrintData.cpp:629: Potential pointer usage: stream->print("Unknown data");
dscKeybusPrintData.cpp:665: Potential pointer usage: case 0x2A: stream->print(F("Quick exit")); return;
dscKeybusPrintData.cpp:666: Potential pointer usage: case 0x63: stream->print(F("Keybus fault restored")); return;
dscKeybusPrintData.cpp:667: Potential pointer usage: case 0x64: stream->print(F("Keybus fault")); return;
dscKeybusPrintData.cpp:668: Potential pointer usage: case 0x66: stream->print(F("*1: Zone bypass")); return;
dscKeybusPrintData.cpp:670: Potential pointer usage: case 0x8C: stream->print(F("Cold start")); return;
dscKeybusPrintData.cpp:671: Potential pointer usage: case 0x8D: stream->print(F("Warm start")); return;
dscKeybusPrintData.cpp:672: Potential pointer usage: case 0x8E: stream->print(F("Panel factory default")); return;
dscKeybusPrintData.cpp:673: Potential pointer usage: case 0x91: stream->print(F("Swinger shutdown")); return;
dscKeybusPrintData.cpp:674: Potential pointer usage: case 0x93: stream->print(F("Disarmed: Keyswitch")); return;

dscKeybusPrintData.cpp:675: Potential pointer usage: case 0x96: stream->print(F("Armed: Keyswitch")); return;
dscKeybusPrintData.cpp:676: Potential pointer usage: case 0x97: stream->print(F("Armed: Keypad away")); return;
dscKeybusPrintData.cpp:677: Potential pointer usage: case 0x98: stream->print(F("Armed: Quick-arm")); return;
dscKeybusPrintData.cpp:678: Potential pointer usage: case 0x99: stream->print(F("Activate stay/away zones")); return;
dscKeybusPrintData.cpp:679: Potential pointer usage: case 0x9A: stream->print(F("Armed: Stay")); return;
dscKeybusPrintData.cpp:680: Potential pointer usage: case 0x9B: stream->print(F("Armed: Away")); return;
dscKeybusPrintData.cpp:681: Potential pointer usage: case 0x9C: stream->print(F("Armed: No entry delay")); return;
dscKeybusPrintData.cpp:689: Potential pointer usage: case 0xFF: stream->print(F("Zone expander trouble: 1")); return;
dscKeybusPrintData.cpp:696: Potential pointer usage: stream->print(F("Command output: "));
dscKeybusPrintData.cpp:706: Potential pointer usage: stream->print(F("*1: "));
dscKeybusPrintData.cpp:716: Potential pointer usage: stream->print(F("*5: "));
dscKeybusPrintData.cpp:740: Potential pointer usage: stream->print(F("*6: "));
dscKeybusPrintData.cpp:754: Potential pointer usage: stream->print(F("Keypad restored: Slot "));
dscKeybusPrintData.cpp:768: Potential pointer usage: stream->print(F("Keypad trouble: Slot "));
dscKeybusPrintData.cpp:777: Potential pointer usage: stream->print(F("Zone expander restored: "));
dscKeybusPrintData.cpp:782: Potential pointer usage: stream->print("Unknown data");
dscKeybusPrintData.cpp:805: Potential pointer usage: case 0x05: stream->print(F("PC/RF5132: Supervisory restored")); return;
dscKeybusPrintData.cpp:806: Potential pointer usage: case 0x06: stream->print(F("PC/RF5132: Supervisory trouble")); return;
dscKeybusPrintData.cpp:807: Potential pointer usage: case 0x09: stream->print(F("PC5204: Supervisory restored")); return;
dscKeybusPrintData.cpp:808: Potential pointer usage: case 0x0A: stream->print(F("PC5204: Supervisory trouble")); return;
dscKeybusPrintData.cpp:809: Potential pointer usage: case 0x17: stream->print(F("Zone expander restored: 7")); return;
dscKeybusPrintData.cpp:810: Potential pointer usage: case 0x18: stream->print(F("Zone expander trouble: 7")); return;
dscKeybusPrintData.cpp:815: Potential pointer usage: case 0x41: stream->print(F("PC/RF5132: Tamper restored")); return;
dscKeybusPrintData.cpp:816: Potential pointer usage: case 0x42: stream->print(F("PC/RF5132: Tamper")); return;
dscKeybusPrintData.cpp:817: Potential pointer usage: case 0x43: stream->print(F("PC5208: Tamper restored")); return;
dscKeybusPrintData.cpp:818: Potential pointer usage: case 0x44: stream->print(F("PC5208: Tamper")); return;
dscKeybusPrintData.cpp:819: Potential pointer usage: case 0x45: stream->print(F("PC5204: Tamper restored")); return;
dscKeybusPrintData.cpp:820: Potential pointer usage: case 0x46: stream->print(F("PC5204: Tamper")); return;
dscKeybusPrintData.cpp:821: Potential pointer usage: case 0x51: stream->print(F("Zone expander tamper restored: 7")); return;
dscKeybusPrintData.cpp:822: Potential pointer usage: case 0x52: stream->print(F("Zone expander tamper: 7")); return;
dscKeybusPrintData.cpp:823: Potential pointer usage: case 0xB3: stream->print(F("PC5204: Battery restored")); return;
dscKeybusPrintData.cpp:824: Potential pointer usage: case 0xB4: stream->print(F("PC5204: Battery trouble")); return;
dscKeybusPrintData.cpp:825: Potential pointer usage: case 0xB5: stream->print(F("PC5204: Aux supply restored")); return;
dscKeybusPrintData.cpp:826: Potential pointer usage: case 0xB6: stream->print(F("PC5204: Aux supply trouble")); return;
dscKeybusPrintData.cpp:827: Potential pointer usage: case 0xB7: stream->print(F("PC5204: Output 1 restored")); return;
dscKeybusPrintData.cpp:828: Potential pointer usage: case 0xB8: stream->print(F("PC5204: Output 1 trouble")); return;
dscKeybusPrintData.cpp:829: Potential pointer usage: case 0xFF: stream->print(F("Extended status")); return;
dscKeybusPrintData.cpp:836: Potential pointer usage: stream->print(F("Zone expander trouble: "));
dscKeybusPrintData.cpp:845: Potential pointer usage: stream->print(F("Keypad tamper restored: "));
dscKeybusPrintData.cpp:854: Potential pointer usage: stream->print(F("Keypad tamper: "));
dscKeybusPrintData.cpp:869: Potential pointer usage: stream->print(F("Zone expander tamper restored: "));
dscKeybusPrintData.cpp:884: Potential pointer usage: stream->print(F("Zone expander tamper: "));
dscKeybusPrintData.cpp:889: Potential pointer usage: stream->print("Unknown data");
dscKeybusPrintData.cpp:911: Potential pointer usage: case 0x86: stream->print(F("Periodic test with trouble")); return;

dscKeybusPrintData.cpp:912: Potential pointer usage: case 0x87: stream->print(F("Exit fault")); return;
dscKeybusPrintData.cpp:913: Potential pointer usage: case 0x89: stream->print(F("Alarm cancelled")); return;
dscKeybusPrintData.cpp:917: Potential pointer usage: stream->print("Zone alarm: ");
dscKeybusPrintData.cpp:922: Potential pointer usage: stream->print("Zone alarm restored: ");
dscKeybusPrintData.cpp:927: Potential pointer usage: stream->print("Zone tamper: ");
dscKeybusPrintData.cpp:932: Potential pointer usage: stream->print("Zone tamper restored: ");
dscKeybusPrintData.cpp:936: Potential pointer usage: else stream->print("Unknown data");
dscKeybusPrintData.cpp:958: Potential pointer usage: stream->print(F("Armed: "));
dscKeybusPrintData.cpp:970: Potential pointer usage: stream->print(F("Disarmed: "));
dscKeybusPrintData.cpp:975: Potential pointer usage: stream->print("Unknown data");
dscKeybusPrintData.cpp:991: Potential pointer usage: case 0xC0: stream->print(F("TLink com fault")); return;
dscKeybusPrintData.cpp:992: Potential pointer usage: case 0xC2: stream->print(F("Tlink network fault")); return;
dscKeybusPrintData.cpp:993: Potential pointer usage: case 0xC4: stream->print(F("TLink receiver trouble")); return;
dscKeybusPrintData.cpp:994: Potential pointer usage: case 0xC5: stream->print(F("TLink receiver restored")); return;
dscKeybusPrintData.cpp:1013: Potential pointer usage: case 0x80: stream->print(F("Trouble acknowledged")); return;
dscKeybusPrintData.cpp:1014: Potential pointer usage: case 0x81: stream->print(F("RF delinquency trouble")); return;
dscKeybusPrintData.cpp:1015: Potential pointer usage: case 0x82: stream->print(F("RF delinquency restore")); return;
dscKeybusPrintData.cpp:1039: Potential pointer usage: stream->print(F("**1: "));
dscKeybusPrintData.cpp:1051: Potential pointer usage: stream->print(F("**2: "));
dscKeybusPrintData.cpp:1058: Potential pointer usage: stream->print(F("**2: "));
dscKeybusPrintData.cpp:1070: Potential pointer usage: stream->print(F("**3: "));
dscKeybusPrintData.cpp:1077: Potential pointer usage: stream->print(F("**3: "));
dscKeybusPrintData.cpp:1082: Potential pointer usage: stream->print("Unknown data");
dscKeybusPrintData.cpp:1117: Potential pointer usage: stream->print(F("**5: "));
dscKeybusPrintData.cpp:1130: Potential pointer usage: stream->print(F("**6: "));
dscKeybusPrintData.cpp:1135: Potential pointer usage: stream->print("Unknown data");
dscKeybusPrintData.cpp:1150: Potential pointer usage: case 0xF1: stream->print(F("System reset transmission"));
return;
dscKeybusPrintData.cpp:1193: Potential pointer usage: stream->print(" | ");
dscKeybusPrintData.cpp:1240: Potential pointer usage: if (!zoneLights) stream->print(F("none"));
dscKeybusPrintData.cpp:1268: Potential pointer usage: stream->print(F("Module supervision query"));
dscKeybusPrintData.cpp:1304: Potential pointer usage: stream->print(F("Panel version: v"));
dscKeybusPrintData.cpp:1305: Potential pointer usage: stream->print(panelData[3] >> 4);
dscKeybusPrintData.cpp:1306: Potential pointer usage: stream->print(".");
dscKeybusPrintData.cpp:1307: Potential pointer usage: stream->print(panelData[3] & 0x0F);
dscKeybusPrintData.cpp:1310: Potential pointer usage: stream->print(F(" | Zone wiring: "));
dscKeybusPrintData.cpp:1312: Potential pointer usage: case 0x01: stream->print(F("NC ")); break;
dscKeybusPrintData.cpp:1313: Potential pointer usage: case 0x02: stream->print(F("EOL ")); break;
dscKeybusPrintData.cpp:1314: Potential pointer usage: case 0x03: stream->print(F("DEOL ")); break;
dscKeybusPrintData.cpp:1318: Potential pointer usage: stream->print(F(" | Code length: "));
dscKeybusPrintData.cpp:1319: Potential pointer usage: if (panelData[4] & 0x08) stream->print(F("6"));
dscKeybusPrintData.cpp:1320: Potential pointer usage: else stream->print(F("4"));
dscKeybusPrintData.cpp:1321: Potential pointer usage: stream->print(F(" digits "));
dscKeybusPrintData.cpp:1324: Potential pointer usage: stream->print(F(" | *8 programming: "));
dscKeybusPrintData.cpp:1325: Potential pointer usage: if (panelData[4] & 0x10) stream->print(F("no "));
dscKeybusPrintData.cpp:1326: Potential pointer usage: else stream->print(F("yes "));
dscKeybusPrintData.cpp:1370: Potential pointer usage: stream->print(F("Verify keypad Fire/Auxiliary/Panic"));
dscKeybusPrintData.cpp:1400: Potential pointer usage: stream->print(F("Zone expander query: "));
dscKeybusPrintData.cpp:1401: Potential pointer usage: stream->print(expander);
dscKeybusPrintData.cpp:1441: Potential pointer usage: stream->print(F(" | Zones 1-8 open: "));
dscKeybusPrintData.cpp:1468: Potential pointer usage: stream->print(F(" | Zones 9-16 open: "));
dscKeybusPrintData.cpp:1490: Potential pointer usage: stream->print(F(" | Zones 17-24 open: "));
dscKeybusPrintData.cpp:1512: Potential pointer usage: stream->print(F(" | Zones 25-32 open: "));

dscKeybusPrintData.cpp:1532: Potential pointer usage: stream->print(F("Wireless module query"));
dscKeybusPrintData.cpp:1559: Potential pointer usage: stream->print(F("Module tamper query"));
dscKeybusPrintData.cpp:1577: Potential pointer usage: stream->print(F("Wireless key query"));
dscKeybusPrintData.cpp:1596: Potential pointer usage: stream->print(F("Module status query"));
dscKeybusPrintData.cpp:1637: Potential pointer usage: stream->print(F("| Zones 1-32 flashing: "));
dscKeybusPrintData.cpp:1708: Potential pointer usage: stream->print(F("LCD display: "));
dscKeybusPrintData.cpp:1710: Potential pointer usage: if (panelData[2] <= 0x63) stream->print("0");
dscKeybusPrintData.cpp:1711: Potential pointer usage: if (panelData[2] <= 0x09) stream->print("0");
dscKeybusPrintData.cpp:1712: Potential pointer usage: stream->print(panelData[2], DEC);
dscKeybusPrintData.cpp:1716: Potential pointer usage: stream->print(panelData[panelByte] >> 4, HEX);
dscKeybusPrintData.cpp:1717: Potential pointer usage: stream->print(panelData[panelByte] & 0x0F, HEX);
dscKeybusPrintData.cpp:1736: Potential pointer usage: stream->print(F("LCD keypad data query"));
dscKeybusPrintData.cpp:1857: Potential pointer usage: stream->print(F("PGM outputs enabled: "));
dscKeybusPrintData.cpp:1858: Potential pointer usage: if (panelData[2] == 0 && panelData[3] == 0)
stream->print(F("none "));
dscKeybusPrintData.cpp:1865: Potential pointer usage: if (panelData[3] & 0x04) stream->print(F("| Midnight "));
dscKeybusPrintData.cpp:1866: Potential pointer usage: if (panelData[3] & 0x08) stream->print(F("| Battery check"));
dscKeybusPrintData.cpp:1927: Potential pointer usage: stream->print(F("Module programming entry: "));
dscKeybusPrintData.cpp:1930: Potential pointer usage: stream->print(F("Module programming entry"));
dscKeybusPrintData.cpp:1959: Potential pointer usage: stream->print(F("Module programming request: "));
dscKeybusPrintData.cpp:1962: Potential pointer usage: stream->print(F("Module programming request"));
dscKeybusPrintData.cpp:1979: Potential pointer usage: stream->print(F("DLS query"));
dscKeybusPrintData.cpp:2014: Potential pointer usage: stream->print(F(" | Timestamp"));
dscKeybusPrintData.cpp:2018: Potential pointer usage: stream->print(" |");
dscKeybusPrintData.cpp:2020: Potential pointer usage: case 0x01: printPartition(); printNumberSpace(1);
stream->print("| "); break;
dscKeybusPrintData.cpp:2021: Potential pointer usage: case 0x02: printPartition(); printNumberSpace(2);
stream->print("| "); break;
dscKeybusPrintData.cpp:2069: Potential pointer usage: stream->print(F("Event: "));
dscKeybusPrintData.cpp:2070: Potential pointer usage: if (panelData[7] < 10) stream->print("00");
dscKeybusPrintData.cpp:2071: Potential pointer usage: else if (panelData[7] < 100) stream->print("0");
dscKeybusPrintData.cpp:2072: Potential pointer usage: stream->print(panelData[7]);
dscKeybusPrintData.cpp:2073: Potential pointer usage: stream->print(" |");
dscKeybusPrintData.cpp:2077: Potential pointer usage: stream->print(" |");
dscKeybusPrintData.cpp:2079: Potential pointer usage: case 0x01: printPartition(); printNumberSpace(1);
stream->print("| "); break;
dscKeybusPrintData.cpp:2080: Potential pointer usage: case 0x02: printPartition(); printNumberSpace(2);
stream->print("| "); break;
dscKeybusPrintData.cpp:2117: Potential pointer usage: stream->print(F("Enabled zones 1-32 | Partition 1: "));
dscKeybusPrintData.cpp:2120: Potential pointer usage: stream->print(F("| Partition 2: "));
dscKeybusPrintData.cpp:2145: Potential pointer usage: stream->print(F("Bell: "));
dscKeybusPrintData.cpp:2146: Potential pointer usage: if (bitRead(panelData[2], 5)) stream->print(F("on"));
dscKeybusPrintData.cpp:2147: Potential pointer usage: else stream->print(F("off"));
dscKeybusPrintData.cpp:2175: Potential pointer usage: stream->print(F("TLM: "));
dscKeybusPrintData.cpp:2176: Potential pointer usage: if (panelData[2] & 0x10) stream->print(F("trouble/attempt"));
dscKeybusPrintData.cpp:2177: Potential pointer usage: else stream->print(F("available/disabled"));
dscKeybusPrintData.cpp:2179: Potential pointer usage: if (panelData[2] & 0x08) stream->print(F(" | Dialer call attempt"));
dscKeybusPrintData.cpp:2180: Potential pointer usage: if (panelData[2] & 0x20) stream->print(F(" | Keypad lockout"));
dscKeybusPrintData.cpp:2226: Potential pointer usage: stream->print(F(" [Byte 2/0x"));
dscKeybusPrintData.cpp:2227: Potential pointer usage: if (panelData[2] < 16) stream->print("0");
dscKeybusPrintData.cpp:2228: Potential pointer usage: stream->print(panelData[2], HEX);
dscKeybusPrintData.cpp:2229: Potential pointer usage: stream->print(F("] "));
dscKeybusPrintData.cpp:2248: Potential pointer usage: stream->print(F("Keypad zone query"));

dscKeybusPrintData.cpp:2268: Potential pointer usage: stream->print(F("[CRC Error]"));
dscKeybusPrintData.cpp:2300: Potential pointer usage: default: stream->print("Unknown data");
dscKeybusPrintData.cpp:2384: Potential pointer usage: stream->print(F("Zone expander query: "));
dscKeybusPrintData.cpp:2385: Potential pointer usage: stream->print(expander);
dscKeybusPrintData.cpp:2403: Potential pointer usage: stream->print(F("Zones 33-40 open: "));
dscKeybusPrintData.cpp:2422: Potential pointer usage: stream->print(F("Zones 41-48 open: "));
dscKeybusPrintData.cpp:2441: Potential pointer usage: stream->print(F("Zones 49-56 open: "));
dscKeybusPrintData.cpp:2460: Potential pointer usage: stream->print(F("Zones 57-64 open: "));
dscKeybusPrintData.cpp:2491: Potential pointer usage: stream->print(F("| Zones 1-32 flashing: "));
dscKeybusPrintData.cpp:2524: Potential pointer usage: stream->print(F("| Zones 33-64 flashing: "));
dscKeybusPrintData.cpp:2583: Potential pointer usage: stream->print(F("Partitions in alarm: "));
dscKeybusPrintData.cpp:2586: Potential pointer usage: if (panelData[6] & 0x08) stream->print(F("| Loss of system time
"));
dscKeybusPrintData.cpp:2587: Potential pointer usage: if (panelData[6] & 0x10) stream->print(F("| AC power trouble "));
dscKeybusPrintData.cpp:2588: Potential pointer usage: if (panelData[6] & 0x40) stream->print(F("| Fail to communicate
"));
dscKeybusPrintData.cpp:2589: Potential pointer usage: if (panelData[6] & 0x80) stream->print(F("| Fire alarm "));
dscKeybusPrintData.cpp:2659: Potential pointer usage: stream->print(F("| Enabled zones 1-32: "));
dscKeybusPrintData.cpp:2687: Potential pointer usage: stream->print(F("| Enabled zones 33-64: "));
dscKeybusPrintData.cpp:2748: Potential pointer usage: if (panelData[2] == 0) stream->print(" | ");
dscKeybusPrintData.cpp:2750: Potential pointer usage: stream->print(F(" | Partition "));
dscKeybusPrintData.cpp:2752: Potential pointer usage: stream->print(" | ");
dscKeybusPrintData.cpp:2805: Potential pointer usage: stream->print(F("Event: "));
dscKeybusPrintData.cpp:2806: Potential pointer usage: if (eventNumber < 10) stream->print("00");
dscKeybusPrintData.cpp:2807: Potential pointer usage: else if (eventNumber < 100) stream->print("0");
dscKeybusPrintData.cpp:2808: Potential pointer usage: stream->print(eventNumber);
dscKeybusPrintData.cpp:2809: Potential pointer usage: stream->print(" | ");
dscKeybusPrintData.cpp:2812: Potential pointer usage: stream->print(" | ");
dscKeybusPrintData.cpp:2817: Potential pointer usage: stream->print(" | ");
dscKeybusPrintData.cpp:2832: Potential pointer usage: case 0xFF: stream->print(F("No entry")); return;
dscKeybusPrintData.cpp:2849: Potential pointer usage: stream->print(F("[Keypad] Fire alarm"));
dscKeybusPrintData.cpp:2863: Potential pointer usage: stream->print(F("[Keypad] Auxiliary alarm"));
dscKeybusPrintData.cpp:2877: Potential pointer usage: stream->print(F("[Keypad] Panic alarm"));
dscKeybusPrintData.cpp:2950: Potential pointer usage: if (printedMessage) stream->print(" | ");
dscKeybusPrintData.cpp:2951: Potential pointer usage: stream->print(F("Keypad on partition: "));
dscKeybusPrintData.cpp:2953: Potential pointer usage: stream->print(F("going idle"));
dscKeybusPrintData.cpp:2959: Potential pointer usage: if (printedMessage) stream->print(" | ");
dscKeybusPrintData.cpp:2960: Potential pointer usage: stream->print(F("Zone expander notification: "));
dscKeybusPrintData.cpp:2972: Potential pointer usage: if (printedMessage) stream->print(" | ");
dscKeybusPrintData.cpp:2973: Potential pointer usage: stream->print(F("Module tamper notification "));
dscKeybusPrintData.cpp:2979: Potential pointer usage: if (printedMessage) stream->print(" | ");
dscKeybusPrintData.cpp:2980: Potential pointer usage: stream->print(F("Wireless module battery notification "));
dscKeybusPrintData.cpp:2987: Potential pointer usage: if (printedMessage) stream->print(" | ");
dscKeybusPrintData.cpp:2988: Potential pointer usage: stream->print(F("Wireless notification "));
dscKeybusPrintData.cpp:2994: Potential pointer usage: if (printedMessage) stream->print(" | ");
dscKeybusPrintData.cpp:2995: Potential pointer usage: stream->print(F("Keypad zone notification "));
dscKeybusPrintData.cpp:3001: Potential pointer usage: if (printedMessage) stream->print(" | ");
dscKeybusPrintData.cpp:3002: Potential pointer usage: stream->print(F("Module status notification "));
dscKeybusPrintData.cpp:3008: Potential pointer usage: if (printedMessage) stream->print(" | ");
dscKeybusPrintData.cpp:3009: Potential pointer usage: stream->print(F("Wireless key notification "));
dscKeybusPrintData.cpp:3015: Potential pointer usage: if (printedMessage) stream->print(" | ");
dscKeybusPrintData.cpp:3016: Potential pointer usage: stream->print(F("Keypad notification "));
dscKeybusPrintData.cpp:3021: Potential pointer usage: if (printedMessage) stream->print(" | ");

dscKeybusPrintData.cpp:3022: Potential pointer usage: stream->print(F("Door chime broadcast "));

dscKeybusPrintData.cpp:3027: Potential pointer usage: if (printedMessage) stream->print("| ");

dscKeybusPrintData.cpp:3028: Potential pointer usage: stream->print(F("Zone label broadcast "));

dscKeybusPrintData.cpp:3093: Potential pointer usage: stream->print(F("Keypad slots: "));

dscKeybusPrintData.cpp:3098: Potential pointer usage: stream->print(F("| Zone expander: "));

dscKeybusPrintData.cpp:3112: Potential pointer usage: if ((moduleData[5] & 0x0C) == 0) stream->print(F("| PC/RF5132 "));

dscKeybusPrintData.cpp:3113: Potential pointer usage: if ((moduleData[5] & 0x03) == 0) stream->print(F("| PC5208 "));

dscKeybusPrintData.cpp:3114: Potential pointer usage: if ((moduleData[6] & 0xC0) == 0) stream->print(F("| PC5204 "));

dscKeybusPrintData.cpp:3142: Potential pointer usage: stream->print(F("Wireless module "));

dscKeybusPrintData.cpp:3145: Potential pointer usage: stream->print(F("| Battery low zones: "));

dscKeybusPrintData.cpp:3150: Potential pointer usage: stream->print(F("| Battery restored zones: "));

dscKeybusPrintData.cpp:3228: Potential pointer usage: stream->print(F("Keypad tamper: Slot "));

dscKeybusPrintData.cpp:3234: Potential pointer usage: if (printedMessage) stream->print("| ");

dscKeybusPrintData.cpp:3235: Potential pointer usage: stream->print(F("Keypad tamper restored: Slot "));

dscKeybusPrintData.cpp:3241: Potential pointer usage: if (printedMessage) stream->print("| ");

dscKeybusPrintData.cpp:3242: Potential pointer usage: stream->print(F("Module tamper: Slot "));

dscKeybusPrintData.cpp:3249: Potential pointer usage: if (printedMessage) stream->print("| ");

dscKeybusPrintData.cpp:3250: Potential pointer usage: stream->print(F("Module tamper restored: Slot "));

dscKeybusPrintData.cpp:3257: Potential pointer usage: if (printedMessage) stream->print("| ");

dscKeybusPrintData.cpp:3258: Potential pointer usage: stream->print(F("RF5132: Tamper "));

dscKeybusPrintData.cpp:3263: Potential pointer usage: if (printedMessage) stream->print("| ");

dscKeybusPrintData.cpp:3264: Potential pointer usage: stream->print(F("RF5132: Tamper restored "));

dscKeybusPrintData.cpp:3269: Potential pointer usage: if (printedMessage) stream->print("| ");

dscKeybusPrintData.cpp:3270: Potential pointer usage: stream->print(F("PC5208: Tamper "));

dscKeybusPrintData.cpp:3275: Potential pointer usage: if (printedMessage) stream->print("| ");

dscKeybusPrintData.cpp:3276: Potential pointer usage: stream->print(F("PC5208: Tamper restored "));

dscKeybusPrintData.cpp:3281: Potential pointer usage: if (printedMessage) stream->print("| ");

dscKeybusPrintData.cpp:3282: Potential pointer usage: stream->print(F("PC5204: Tamper "));

dscKeybusPrintData.cpp:3287: Potential pointer usage: if (printedMessage) stream->print("| ");

dscKeybusPrintData.cpp:3288: Potential pointer usage: stream->print(F("PC5204: Tamper restored "));

dscKeybusPrintData.cpp:3327: Potential pointer usage: stream->print(F("Wireless key low battery: "));

dscKeybusPrintData.cpp:3333: Potential pointer usage: if (printedMessage) stream->print(F("| "));

dscKeybusPrintData.cpp:3334: Potential pointer usage: stream->print(F("Wireless key battery restored: "));

dscKeybusPrintData.cpp:3384: Potential pointer usage: stream->print(F("PC5204: Battery restored "));

dscKeybusPrintData.cpp:3389: Potential pointer usage: if (printedMessage) stream->print("| ");

dscKeybusPrintData.cpp:3390: Potential pointer usage: stream->print(F("PC5204: Battery trouble "));

dscKeybusPrintData.cpp:3395: Potential pointer usage: if (printedMessage) stream->print("| ");

dscKeybusPrintData.cpp:3396: Potential pointer usage: stream->print(F("PC5204: AC power restored "));

dscKeybusPrintData.cpp:3401: Potential pointer usage: if (printedMessage) stream->print("| ");

dscKeybusPrintData.cpp:3402: Potential pointer usage: stream->print(F("PC5204: AC power trouble "));

dscKeybusPrintData.cpp:3407: Potential pointer usage: if (printedMessage) stream->print("| ");

dscKeybusPrintData.cpp:3408: Potential pointer usage: stream->print(F("PC5204: Output 1 restored "));

dscKeybusPrintData.cpp:3413: Potential pointer usage: if (printedMessage) stream->print("| ");

dscKeybusPrintData.cpp:3414: Potential pointer usage: stream->print(F("PC5204: Output 1 trouble "));

dscKeybusPrintData.cpp:3445: Potential pointer usage: stream->print(F("LCD keypad data entry: "));

dscKeybusPrintData.cpp:3447: Potential pointer usage: if (moduleData[2] <= 0x63) stream->print("0");

dscKeybusPrintData.cpp:3448: Potential pointer usage: if (moduleData[2] <= 0x09) stream->print("0");

dscKeybusPrintData.cpp:3449: Potential pointer usage: stream->print(moduleData[2], DEC);

dscKeybusPrintData.cpp:3453: Potential pointer usage: stream->print(moduleData[moduleByte] >> 4, HEX);

dscKeybusPrintData.cpp:3454: Potential pointer usage: stream->print(moduleData[moduleByte] & 0x0F, HEX);

dscKeybusPrintData.cpp:3466: Potential pointer usage: stream->print(F("Module programming response"));

dscKeybusPrintData.cpp:3495: Potential pointer usage: stream->print(F("Keypad "));

dscKeybusPrintData.cpp:3501: Potential pointer usage: stream->print(F("Slot "));
dscKeybusPrintData.cpp:3504: Potential pointer usage: else stream->print(F(" | Slot "));
dscKeybusPrintData.cpp:3506: Potential pointer usage: stream->print(moduleByte - 1);
dscKeybusPrintData.cpp:3507: Potential pointer usage: if ((slotData & 0x03) == 0x03 && (slotData & 0x30) == 0) stream->print(F(": Zone open"));
dscKeybusPrintData.cpp:3508: Potential pointer usage: if ((slotData & 0x03) == 0 && (slotData & 0x30) == 0x30) stream->print(F(": Zone closed"));
dscKeybusPrintData.cpp:3591: Potential pointer usage: stream->print(F("[Digit] "));
dscKeybusPrintData.cpp:3594: Potential pointer usage: stream->print(F("Partition "));
dscKeybusPrintData.cpp:3596: Potential pointer usage: stream->print(F("Key: "));
dscKeybusPrintData.cpp:3621: Potential pointer usage: case 0x28: stream->print(F("* ")); break;
dscKeybusPrintData.cpp:3622: Potential pointer usage: case 0x2D: stream->print(F("# ")); break;
dscKeybusPrintData.cpp:3623: Potential pointer usage: case 0x46: stream->print(F("Wireless key disarm ")); break;
//isn't send if wls keys uses access codes 17-32 is enabled
dscKeybusPrintData.cpp:3624: Potential pointer usage: case 0x52: stream->print(F("Identified voice prompt help ")); break;
dscKeybusPrintData.cpp:3625: Potential pointer usage: case 0x6E: stream->print(F("Global away arm ")); break;
dscKeybusPrintData.cpp:3626: Potential pointer usage: case 0x70: stream->print(F("Command output 3 ")); break;
dscKeybusPrintData.cpp:3627: Potential pointer usage: case 0x7A: stream->print(F("Time and date programming ")); break;
dscKeybusPrintData.cpp:3628: Potential pointer usage: case 0x75: stream->print(F("Entered *1/*2/*3 ? ")); break;
dscKeybusPrintData.cpp:3629: Potential pointer usage: case 0x82: stream->print(F("Enter ")); break;
dscKeybusPrintData.cpp:3630: Potential pointer usage: case 0x87: stream->print(F("Right arrow ")); break;
dscKeybusPrintData.cpp:3631: Potential pointer usage: case 0x88: stream->print(F("Left arrow ")); break;
dscKeybusPrintData.cpp:3632: Potential pointer usage: case 0x8D: stream->print(F("Bypass recall ")); break;
dscKeybusPrintData.cpp:3633: Potential pointer usage: case 0x93: stream->print(F("Recall bypass group ")); break;
dscKeybusPrintData.cpp:3634: Potential pointer usage: case 0x94: stream->print(F("Global label broadcast ")); break;
dscKeybusPrintData.cpp:3635: Potential pointer usage: case 0x99: stream->print(F("Function key [25] Future Use ")); break;
dscKeybusPrintData.cpp:3636: Potential pointer usage: case 0xA5: stream->print(F("Receive data ")); break;
dscKeybusPrintData.cpp:3637: Potential pointer usage: case 0xAA: stream->print(F("Submit data ")); break;
dscKeybusPrintData.cpp:3638: Potential pointer usage: case 0xAF: stream->print(F("Arm: Stay ")); break;
dscKeybusPrintData.cpp:3639: Potential pointer usage: case 0xB1: stream->print(F("Arm: Away ")); break;
dscKeybusPrintData.cpp:3640: Potential pointer usage: case 0xB6: stream->print(F("Arm: No entry delay ")); break;
dscKeybusPrintData.cpp:3641: Potential pointer usage: case 0xBB: stream->print(F("Door chime configuration ")); break;
dscKeybusPrintData.cpp:3642: Potential pointer usage: case 0xBC: stream->print(F("*6 System test ")); break;
dscKeybusPrintData.cpp:3643: Potential pointer usage: case 0xC3: stream->print(F("*1 Zone bypass programming ")); break;
dscKeybusPrintData.cpp:3644: Potential pointer usage: case 0xC4: stream->print(F("*2 Trouble menu ")); break;
dscKeybusPrintData.cpp:3645: Potential pointer usage: case 0xC9: stream->print(F("*3 Alarm memory display ")); break;
dscKeybusPrintData.cpp:3646: Potential pointer usage: case 0xCE: stream->print(F("*5 Programming ")); break;
dscKeybusPrintData.cpp:3647: Potential pointer usage: case 0xD0: stream->print(F("*6 Programming ")); break;
dscKeybusPrintData.cpp:3648: Potential pointer usage: case 0xD5: stream->print(F("Command output 1 ")); break;
dscKeybusPrintData.cpp:3649: Potential pointer usage: case 0xDA: stream->print(F("Reset / Command output 2 ")); break;
dscKeybusPrintData.cpp:3650: Potential pointer usage: case 0xDF: stream->print(F("Global stay arm ")); break;
dscKeybusPrintData.cpp:3651: Potential pointer usage: case 0xE1: stream->print(F("Quick exit ")); break;
dscKeybusPrintData.cpp:3652: Potential pointer usage: case 0xE6: stream->print(F("Activate stay/away zones ")); break;
dscKeybusPrintData.cpp:3653: Potential pointer usage: case 0xEB: stream->print(F("LCD pixel test ")); break;
dscKeybusPrintData.cpp:3654: Potential pointer usage: case 0xEC: stream->print(F("Command output 4 ")); break;

dscKeybusPrintData.cpp:3655: Potential pointer usage: case 0xF2: stream->print(F("Global disarm ")); break; // Possible overlap with "General voice prompt help"

dscKeybusPrintData.cpp:3656: Potential pointer usage: case 0xF7: stream->print(F("Menu navigation ")); break;

dscKeybusPrintData.cpp:3668: Potential pointer usage: stream->print(F("Zone expander: "));

dscKeybusPrintData.cpp:3687: Potential pointer usage: stream->print(F("| Zones changed: "));

dscKeybusPrintData.cpp:3697: Potential pointer usage: case 0: stream->print(F("open ")); break;

dscKeybusPrintData.cpp:3698: Potential pointer usage: case 1: stream->print(F("closed ")); break;

dscKeybusPrintData.cpp:3699: Potential pointer usage: case 2: stream->print(F("tamper ")); break;

dscKeybusPrintData.cpp:3700: Potential pointer usage: case 3: stream->print(F("open (D/EOL) ")); break;

dscKeybusPrintData.cpp:3748: Potential pointer usage: case 0x11: stream->print(F("RF5132")); break; //section 804 verified on pc1832 and pc5020

dscKeybusPrintData.cpp:3749: Potential pointer usage: case 0x14: stream->print(F("RF5400")); break; //section 801 not verified

dscKeybusPrintData.cpp:3750: Potential pointer usage: case 0x15: stream->print(F("RF5936")); break; //section 802 not verified

dscKeybusPrintData.cpp:3751: Potential pointer usage: case 0x16: stream->print(F("LINKS2X50")); break; //section 803 not verified

dscKeybusPrintData.cpp:3752: Potential pointer usage: case 0x17: stream->print(F("PC5108L")); break; //section 806 not verified

dscKeybusPrintData.cpp:3753: Potential pointer usage: case 0x19: stream->print(F("RF5100")); break; //section 805 not verified

dscKeybusPrintData.cpp:3754: Potential pointer usage: case 0x31: stream->print(F("**5 user")); break; /**5 access codes verified on pc1832 and pc5020

dscKeybusPrintData.cpp:3755: Potential pointer usage: default: stream->print("Unknown data");

dscKeybusPrintData.cpp:3757: Potential pointer usage: stream->print(" | ");

dscKeybusPrintData.cpp:3758: Potential pointer usage: if (panelByte3 < 16) stream->print("0");

dscKeybusPrintData.cpp:3759: Potential pointer usage: stream->print(panelByte3, HEX);

dscKeybusPrintData.cpp:3769: Potential pointer usage: if (partitionCount > startPartition) stream->print(" | ");

dscKeybusPrintData.cpp:3772: Potential pointer usage: stream->print(partitionCount);

dscKeybusPrintData.cpp:3773: Potential pointer usage: stream->print(": ");

dscKeybusPrintData.cpp:3775: Potential pointer usage: if (panelData[statusByte] == 0 || panelData[statusByte] == 0xC7 || panelData[statusByte] == 0xFF) stream->print(F("disabled"));

dscKeybusPrintData.cpp:3798: Potential pointer usage: if (dscYear3 >= 7) stream->print(F("19"));

dscKeybusPrintData.cpp:3799: Potential pointer usage: else stream->print(F("20"));

dscKeybusPrintData.cpp:3800: Potential pointer usage: stream->print(dscYear3);

dscKeybusPrintData.cpp:3801: Potential pointer usage: stream->print(dscYear4, HEX);

dscKeybusPrintData.cpp:3802: Potential pointer usage: stream->print(".");

dscKeybusPrintData.cpp:3803: Potential pointer usage: if (dscMonth < 10) stream->print("0");

dscKeybusPrintData.cpp:3804: Potential pointer usage: stream->print(dscMonth);

dscKeybusPrintData.cpp:3805: Potential pointer usage: stream->print(".");

dscKeybusPrintData.cpp:3806: Potential pointer usage: if (dscDay < 10) stream->print("0");

dscKeybusPrintData.cpp:3807: Potential pointer usage: stream->print(dscDay);

dscKeybusPrintData.cpp:3808: Potential pointer usage: stream->print(" ");

dscKeybusPrintData.cpp:3809: Potential pointer usage: if (dscHour < 10) stream->print("0");

dscKeybusPrintData.cpp:3810: Potential pointer usage: stream->print(dscHour);

dscKeybusPrintData.cpp:3811: Potential pointer usage: stream->print(F(":"));

dscKeybusPrintData.cpp:3812: Potential pointer usage: if (dscMinute < 10) stream->print("0");

dscKeybusPrintData.cpp:3813: Potential pointer usage: stream->print(dscMinute);

dscKeybusPrintData.cpp:3833: Potential pointer usage: case 40: stream->print(F("Master ")); break;

dscKeybusPrintData.cpp:3834: Potential pointer usage: default: stream->print(F("Access ")); break;

dscKeybusPrintData.cpp:3836: Potential pointer usage: stream->print(F("code "));

dscKeybusPrintData.cpp:3837: Potential pointer usage: stream->print(dscCode);

dscKeybusPrintData.cpp:3847: Potential pointer usage: stream->print(F("| Beep: "));

```

dscKeybusPrintData.cpp:3848: Potential pointer usage: stream->print(panelData[panelByte] / 2);
dscKeybusPrintData.cpp:3849: Potential pointer usage: stream->print(F(" beeps"));
dscKeybusPrintData.cpp:3859: Potential pointer usage: stream->print(F("| Tone: "));
dscKeybusPrintData.cpp:3863: Potential pointer usage: stream->print(F("none"));
dscKeybusPrintData.cpp:3868: Potential pointer usage: stream->print(F("constant tone "));
dscKeybusPrintData.cpp:3873: Potential pointer usage: if (printedMessage) stream->print("| ");
dscKeybusPrintData.cpp:3874: Potential pointer usage: stream->print((panelData[panelByte] & 0x70) >> 4);
dscKeybusPrintData.cpp:3875: Potential pointer usage: stream->print(F(" beep "));
dscKeybusPrintData.cpp:3879: Potential pointer usage: stream->print("| ");
dscKeybusPrintData.cpp:3880: Potential pointer usage: stream->print(panelData[panelByte] & 0x0F);
dscKeybusPrintData.cpp:3881: Potential pointer usage: stream->print(F("s interval"));
dscKeybusPrintData.cpp:3892: Potential pointer usage: stream->print(F("| Buzzer: "));
dscKeybusPrintData.cpp:3893: Potential pointer usage: stream->print(panelData[panelByte]);
dscKeybusPrintData.cpp:3894: Potential pointer usage: stream->print("s");
dscKeybusPrintData.cpp:3910: Potential pointer usage: stream->print((zoneBit + startZone) + ((panelByte - inputByte) *
8));
dscKeybusPrintData.cpp:3911: Potential pointer usage: stream->print(" ");
dscKeybusPrintData.cpp:3916: Potential pointer usage: if (!zonesEnabled && panelData[0] != 0x0A && panelData[0] !=
0x0F) stream->print(F("none"));
dscKeybusPrintData.cpp:3923: Potential pointer usage: stream->print(F("Partition "));
dscKeybusPrintData.cpp:3928: Potential pointer usage: stream->print("Unknown data");
dscKeybusPrintData.cpp:3933: Potential pointer usage: stream->print(F("Status lights: "));
dscKeybusPrintData.cpp:3938: Potential pointer usage: if (lowerRange) stream->print(F(" | Zones 1-32 lights: "));
dscKeybusPrintData.cpp:3939: Potential pointer usage: else stream->print(F(" | Zones 33-64 lights: "));
dscKeybusPrintData.cpp:3944: Potential pointer usage: stream->print(F("| Status lights flashing: "));
dscKeybusPrintData.cpp:3949: Potential pointer usage: stream->print(number);
dscKeybusPrintData.cpp:3950: Potential pointer usage: stream->print(" ");
dscKeybusPrintData.cpp:3955: Potential pointer usage: stream->print(panelData[panelByte] + numberOffset);
dscKeybusPrintData.cpp:3961: Potential pointer usage: if (printNone && panelData[panelByte] == 0)
stream->print(F("none "));
dscKeybusPrintData.cpp:3966: Potential pointer usage: stream->print(startNumber + bitCount);
dscKeybusPrintData.cpp:3967: Potential pointer usage: stream->print(" ");
dscKeybusPrintData.cpp:3978: Potential pointer usage: if (panelByte == 1) stream->print(panelData[panelByte]); //
Prints the stop bit
dscKeybusPrintData.cpp:3981: Potential pointer usage: if (mask & panelData[panelByte]) stream->print("1");
dscKeybusPrintData.cpp:3982: Potential pointer usage: else stream->print("0");
dscKeybusPrintData.cpp:3985: Potential pointer usage: if (printSpaces && (panelByte != panelByteCount - 1 ||
displayTrailingBits)) stream->print(" ");
dscKeybusPrintData.cpp:3992: Potential pointer usage: stream->print(bitRead(panelData[panelByteCount], i));
dscKeybusPrintData.cpp:4002: Potential pointer usage: if (moduleByte == 1) stream->print(moduleData[moduleByte]); //
Prints the stop bit
dscKeybusPrintData.cpp:4007: Potential pointer usage: stream->print(F(".....")); // Hides keypad digits
dscKeybusPrintData.cpp:4010: Potential pointer usage: if (mask & moduleData[moduleByte]) stream->print("1");
dscKeybusPrintData.cpp:4011: Potential pointer usage: else stream->print("0");
dscKeybusPrintData.cpp:4014: Potential pointer usage: if (printSpaces && (moduleByte != moduleByteCount - 1 ||
displayTrailingBits)) stream->print(" ");
dscKeybusPrintData.cpp:4021: Potential pointer usage: stream->print(bitRead(moduleData[moduleByteCount], i));
dscKeybusPrintData.cpp:4030: Potential pointer usage: stream->print(F("0x"));
dscKeybusPrintData.cpp:4031: Potential pointer usage: if (panelData[0] < 16) stream->print("0");
dscKeybusPrintData.cpp:4032: Potential pointer usage: stream->print(panelData[0], HEX);
dscKeybusPrintData.cpp:4035: Potential pointer usage: stream->print(".");
dscKeybusPrintData.cpp:4036: Potential pointer usage: if (panelData[2] < 16) stream->print("0");
dscKeybusPrintData.cpp:4037: Potential pointer usage: stream->print(panelData[2], HEX);

```

dscKeybusPrintData.cpp:68: Pointer dereference without nearby null check: stream->print(F("[CRC Error]"));

dscKeybusPrintData.cpp:120: Pointer dereference without nearby null check: stream->print("Unknown data");

dscKeybusPrintData.cpp:135: Pointer dereference without nearby null check: stream->print(F("[Module/0x"]));

dscKeybusPrintData.cpp:136: Pointer dereference without nearby null check: if (moduleCmd < 16) stream->print("0");

dscKeybusPrintData.cpp:137: Pointer dereference without nearby null check: stream->print(moduleCmd, HEX);

dscKeybusPrintData.cpp:140: Pointer dereference without nearby null check: stream->print(".");

dscKeybusPrintData.cpp:141: Pointer dereference without nearby null check: if (moduleSubCmd < 16) stream->print("0");

dscKeybusPrintData.cpp:142: Pointer dereference without nearby null check: stream->print(moduleSubCmd, HEX);

dscKeybusPrintData.cpp:144: Pointer dereference without nearby null check: stream->print(F("] "));

dscKeybusPrintData.cpp:183: Pointer dereference without nearby null check: default: stream->print("Unknown data");

dscKeybusPrintData.cpp:196: Pointer dereference without nearby null check: if (panelData[panelByte] == 0) stream->print(F("none "));

dscKeybusPrintData.cpp:198: Pointer dereference without nearby null check: if (bitRead(panelData[panelByte], 0)) stream->print(F("Ready "));

dscKeybusPrintData.cpp:199: Pointer dereference without nearby null check: if (bitRead(panelData[panelByte], 1)) stream->print(F("Armed "));

dscKeybusPrintData.cpp:200: Pointer dereference without nearby null check: if (bitRead(panelData[panelByte], 2)) stream->print(F("Memory "));

dscKeybusPrintData.cpp:201: Pointer dereference without nearby null check: if (bitRead(panelData[panelByte], 3)) stream->print(F("Bypass "));

dscKeybusPrintData.cpp:202: Pointer dereference without nearby null check: if (bitRead(panelData[panelByte], 4)) stream->print(F("Trouble "));

dscKeybusPrintData.cpp:203: Pointer dereference without nearby null check: if (bitRead(panelData[panelByte], 5)) stream->print(F("Program "));

dscKeybusPrintData.cpp:204: Pointer dereference without nearby null check: if (bitRead(panelData[panelByte], 6)) stream->print(F("Fire "));

dscKeybusPrintData.cpp:205: Pointer dereference without nearby null check: if (bitRead(panelData[panelByte], 7)) stream->print(F("Backlight "));

dscKeybusPrintData.cpp:209: Pointer dereference without nearby null check: stream->print(F("- "));

dscKeybusPrintData.cpp:222: Pointer dereference without nearby null check: case 0x01: stream->print(F("Partition ready")); break;

dscKeybusPrintData.cpp:223: Pointer dereference without nearby null check: case 0x02: stream->print(F("Stay zones open")); break;

dscKeybusPrintData.cpp:224: Pointer dereference without nearby null check: case 0x03: stream->print(F("Zones open")); break;

dscKeybusPrintData.cpp:225: Pointer dereference without nearby null check: case 0x04: stream->print(F("Armed: Stay")); break;

dscKeybusPrintData.cpp:226: Pointer dereference without nearby null check: case 0x05: stream->print(F("Armed: Away")); break;

dscKeybusPrintData.cpp:227: Pointer dereference without nearby null check: case 0x06: stream->print(F("Armed: Stay with no entry delay")); break;

dscKeybusPrintData.cpp:228: Pointer dereference without nearby null check: case 0x07: stream->print(F("Failed to arm")); break;

dscKeybusPrintData.cpp:229: Pointer dereference without nearby null check: case 0x08: stream->print(F("Exit delay in progress")); break;

dscKeybusPrintData.cpp:230: Pointer dereference without nearby null check: case 0x09: stream->print(F("Arming: No entry delay")); break;

dscKeybusPrintData.cpp:231: Pointer dereference without nearby null check: case 0x0B: stream->print(F("Quick exit in progress")); break;

dscKeybusPrintData.cpp:232: Pointer dereference without nearby null check: case 0x0C: stream->print(F("Entry delay in progress")); break;

dscKeybusPrintData.cpp:233: Pointer dereference without nearby null check: case 0x0D: stream->print(F("Entry delay

after alarm")); break;

dscKeybusPrintData.cpp:234: Pointer dereference without nearby null check: case 0x0E: stream->print(F("Function not available")); break;

dscKeybusPrintData.cpp:235: Pointer dereference without nearby null check: case 0x10: stream->print(F("Keypad lockout")); break;

dscKeybusPrintData.cpp:236: Pointer dereference without nearby null check: case 0x11: stream->print(F("Partition in alarm")); break;

dscKeybusPrintData.cpp:237: Pointer dereference without nearby null check: case 0x12: stream->print(F("Battery check in progress")); break;

dscKeybusPrintData.cpp:238: Pointer dereference without nearby null check: case 0x14: stream->print(F("Auto-arm in progress")); break;

dscKeybusPrintData.cpp:239: Pointer dereference without nearby null check: case 0x15: stream->print(F("Arming with bypassed zones")); break;

dscKeybusPrintData.cpp:240: Pointer dereference without nearby null check: case 0x16: stream->print(F("Armed: Away with no entry delay")); break;

dscKeybusPrintData.cpp:241: Pointer dereference without nearby null check: case 0x19: stream->print(F("Disarmed: Alarm memory")); break;

dscKeybusPrintData.cpp:242: Pointer dereference without nearby null check: case 0x22: stream->print(F("Disarmed: Recent closing")); break;

dscKeybusPrintData.cpp:243: Pointer dereference without nearby null check: case 0x2F: stream->print(F("Keypad LCD test")); break;

dscKeybusPrintData.cpp:244: Pointer dereference without nearby null check: case 0x33: stream->print(F("Command output in progress")); break;

dscKeybusPrintData.cpp:245: Pointer dereference without nearby null check: case 0x3D: stream->print(F("Disarmed: Alarm memory")); break;

dscKeybusPrintData.cpp:246: Pointer dereference without nearby null check: case 0x3E: stream->print(F("Partition disarmed")); break;

dscKeybusPrintData.cpp:248: Pointer dereference without nearby null check: case 0x40: stream->print(F("Keypad blanking")); break;

dscKeybusPrintData.cpp:249: Pointer dereference without nearby null check: case 0x8A: stream->print(F("Activate stay/away zones")); break;

dscKeybusPrintData.cpp:250: Pointer dereference without nearby null check: case 0x8B: stream->print(F("Quick exit")); break;

dscKeybusPrintData.cpp:251: Pointer dereference without nearby null check: case 0x8E: stream->print(F("Function not available")); break;

dscKeybusPrintData.cpp:252: Pointer dereference without nearby null check: case 0x8F: stream->print(F("Invalid access code")); break;

dscKeybusPrintData.cpp:253: Pointer dereference without nearby null check: case 0x9E: stream->print(F("Enter * function key")); break;

dscKeybusPrintData.cpp:254: Pointer dereference without nearby null check: case 0x9F: stream->print(F("Enter access code")); break;

dscKeybusPrintData.cpp:255: Pointer dereference without nearby null check: case 0xA0: stream->print(F("**1: Zone bypass")); break;

dscKeybusPrintData.cpp:256: Pointer dereference without nearby null check: case 0xA1: stream->print(F("**2: Trouble")); break;

dscKeybusPrintData.cpp:257: Pointer dereference without nearby null check: case 0xA2: stream->print(F("**3: Alarm memory")); break;

dscKeybusPrintData.cpp:258: Pointer dereference without nearby null check: case 0xA3: stream->print(F("Door chime enabled")); break;

dscKeybusPrintData.cpp:259: Pointer dereference without nearby null check: case 0xA4: stream->print(F("Door chime disabled")); break;

dscKeybusPrintData.cpp:260: Pointer dereference without nearby null check: case 0xA5: stream->print(F("Enter master code")); break;

dscKeybusPrintData.cpp:261: Pointer dereference without nearby null check: case 0xA6: stream->print(F("**5: Access codes")); break;

dscKeybusPrintData.cpp:262: Pointer dereference without nearby null check: case 0xA7: stream->print(F("**5: Enter 4-digit code")); break;

dscKeybusPrintData.cpp:263: Pointer dereference without nearby null check: case 0xA9: stream->print(F("**6: User functions")); break;

dscKeybusPrintData.cpp:264: Pointer dereference without nearby null check: case 0xAA: stream->print(F("**6: Time and date")); break;

dscKeybusPrintData.cpp:265: Pointer dereference without nearby null check: case 0xAB: stream->print(F("**6: Auto-arm time")); break;

dscKeybusPrintData.cpp:266: Pointer dereference without nearby null check: case 0xAC: stream->print(F("**6: Auto-arm enabled")); break;

dscKeybusPrintData.cpp:267: Pointer dereference without nearby null check: case 0xAD: stream->print(F("**6: Auto-arm disabled")); break;

dscKeybusPrintData.cpp:268: Pointer dereference without nearby null check: case 0xAF: stream->print(F("**6: System test")); break;

dscKeybusPrintData.cpp:269: Pointer dereference without nearby null check: case 0xB0: stream->print(F("**6: Enable DLS")); break;

dscKeybusPrintData.cpp:271: Pointer dereference without nearby null check: case 0xB3: stream->print(F("**7: Command output")); break;

dscKeybusPrintData.cpp:272: Pointer dereference without nearby null check: case 0xB7: stream->print(F("Enter installer code")); break;

dscKeybusPrintData.cpp:273: Pointer dereference without nearby null check: case 0xB8: stream->print(F("Enter * function key while armed")); break;

dscKeybusPrintData.cpp:274: Pointer dereference without nearby null check: case 0xB9: stream->print(F("**2: Zone tamper menu")); break;

dscKeybusPrintData.cpp:275: Pointer dereference without nearby null check: case 0xBA: stream->print(F("**2: Zones with low batteries")); break;

dscKeybusPrintData.cpp:276: Pointer dereference without nearby null check: case 0xBC: stream->print(F("**5: Enter 6-digit code")); break;

dscKeybusPrintData.cpp:277: Pointer dereference without nearby null check: case 0xBF: stream->print(F("**6: Auto-arm select day")); break;

dscKeybusPrintData.cpp:278: Pointer dereference without nearby null check: case 0xC6: stream->print(F("**2: Zone fault menu")); break;

dscKeybusPrintData.cpp:279: Pointer dereference without nearby null check: //case 0xC7: stream->print(F("Partition not available")); break;

dscKeybusPrintData.cpp:280: Pointer dereference without nearby null check: case 0xC8: stream->print(F("**2: Service required menu")); break;

dscKeybusPrintData.cpp:281: Pointer dereference without nearby null check: case 0xCD: stream->print(F("Downloading in progress")); break;

dscKeybusPrintData.cpp:282: Pointer dereference without nearby null check: case 0xCE: stream->print(F("Active camera monitor selection")); break;

dscKeybusPrintData.cpp:283: Pointer dereference without nearby null check: case 0xD0: stream->print(F("**2: Keypads with low batteries")); break;

dscKeybusPrintData.cpp:284: Pointer dereference without nearby null check: case 0xD1: stream->print(F("**2: Keyfobs with low batteries")); break;

dscKeybusPrintData.cpp:285: Pointer dereference without nearby null check: case 0xD4: stream->print(F("**2: Zones with RF Delinquency")); break;

dscKeybusPrintData.cpp:286: Pointer dereference without nearby null check: case 0xE4: stream->print(F("**8: Installer programming, 3 digits")); decimalInput = false; break;

dscKeybusPrintData.cpp:287: Pointer dereference without nearby null check: case 0xE5: stream->print(F("Keypad slot assignment")); break;

dscKeybusPrintData.cpp:288: Pointer dereference without nearby null check: case 0xE6: stream->print(F("Input: 2

digits")); break;

dscKeybusPrintData.cpp:289: Pointer dereference without nearby null check: case 0xE7: stream->print(F("Input: 3 digits")); decimalInput = true; break;

dscKeybusPrintData.cpp:290: Pointer dereference without nearby null check: case 0xE8: stream->print(F("Input: 4 digits")); break;

dscKeybusPrintData.cpp:291: Pointer dereference without nearby null check: case 0xE9: stream->print(F("Input: 5 digits")); break;

dscKeybusPrintData.cpp:292: Pointer dereference without nearby null check: case 0xEA: stream->print(F("Input HEX: 2 digits")); break;

dscKeybusPrintData.cpp:293: Pointer dereference without nearby null check: case 0xEB: stream->print(F("Input HEX: 4 digits")); break;

dscKeybusPrintData.cpp:294: Pointer dereference without nearby null check: case 0xEC: stream->print(F("Input HEX: 6 digits")); break;

dscKeybusPrintData.cpp:295: Pointer dereference without nearby null check: case 0xED: stream->print(F("Input HEX: 32 digits")); break;

dscKeybusPrintData.cpp:296: Pointer dereference without nearby null check: case 0xEE: stream->print(F("Input: 1 option per zone")); break;

dscKeybusPrintData.cpp:297: Pointer dereference without nearby null check: case 0xEF: stream->print(F("Module supervision field")); break;

dscKeybusPrintData.cpp:298: Pointer dereference without nearby null check: case 0xF0: stream->print(F("Function key 1")); break;

dscKeybusPrintData.cpp:299: Pointer dereference without nearby null check: case 0xF1: stream->print(F("Function key 2")); break;

dscKeybusPrintData.cpp:300: Pointer dereference without nearby null check: case 0xF2: stream->print(F("Function key 3")); break;

dscKeybusPrintData.cpp:301: Pointer dereference without nearby null check: case 0xF3: stream->print(F("Function key 4")); break;

dscKeybusPrintData.cpp:302: Pointer dereference without nearby null check: case 0xF4: stream->print(F("Function key 5")); break;

dscKeybusPrintData.cpp:303: Pointer dereference without nearby null check: case 0xF5: stream->print(F("Wireless module placement test")); break;

dscKeybusPrintData.cpp:304: Pointer dereference without nearby null check: case 0xF6: stream->print(F("Activate device for test")); break;

dscKeybusPrintData.cpp:305: Pointer dereference without nearby null check: case 0xF7: stream->print(F("*8: Installer programming, 2 digits")); decimalInput = false; break;

dscKeybusPrintData.cpp:306: Pointer dereference without nearby null check: case 0xF8: stream->print(F("Keypad programming")); break;

dscKeybusPrintData.cpp:307: Pointer dereference without nearby null check: case 0xFA: stream->print(F("Input: 6 digits")); break;

dscKeybusPrintData.cpp:309: Pointer dereference without nearby null check: stream->print(F("Unknown data: 0x"));

dscKeybusPrintData.cpp:310: Pointer dereference without nearby null check: if (panelData[panelByte] < 10) stream->print("0");

dscKeybusPrintData.cpp:311: Pointer dereference without nearby null check: stream->print(panelData[panelByte], HEX);

dscKeybusPrintData.cpp:362: Pointer dereference without nearby null check: case 0x49: stream->print(F("Duress alarm")); break;

dscKeybusPrintData.cpp:363: Pointer dereference without nearby null check: case 0x4A: stream->print(F("Disarmed: Alarm memory")); break;

dscKeybusPrintData.cpp:364: Pointer dereference without nearby null check: case 0x4B: stream->print(F("Recent closing alarm")); break;

dscKeybusPrintData.cpp:365: Pointer dereference without nearby null check: case 0x4C: stream->print(F("Zone expander supervisory alarm")); break;

dscKeybusPrintData.cpp:366: Pointer dereference without nearby null check: case 0x4D: stream->print(F("Zone

expander supervisory restored")); break;

dscKeybusPrintData.cpp:367: Pointer dereference without nearby null check: case 0x4E: stream->print(F("Keypad Fire alarm")); break;

dscKeybusPrintData.cpp:368: Pointer dereference without nearby null check: case 0x4F: stream->print(F("Keypad Aux alarm")); break;

dscKeybusPrintData.cpp:369: Pointer dereference without nearby null check: case 0x50: stream->print(F("Keypad Panic alarm")); break;

dscKeybusPrintData.cpp:370: Pointer dereference without nearby null check: case 0x51: stream->print(F("PGM2 input alarm")); break;

dscKeybusPrintData.cpp:371: Pointer dereference without nearby null check: case 0x52: stream->print(F("Keypad Fire alarm restored")); break;

dscKeybusPrintData.cpp:372: Pointer dereference without nearby null check: case 0x53: stream->print(F("Keypad Aux alarm restored")); break;

dscKeybusPrintData.cpp:373: Pointer dereference without nearby null check: case 0x54: stream->print(F("Keypad Panic alarm restored")); break;

dscKeybusPrintData.cpp:374: Pointer dereference without nearby null check: case 0x55: stream->print(F("PGM2 input alarm restored")); break;

dscKeybusPrintData.cpp:377: Pointer dereference without nearby null check: case 0x98: stream->print(F("Keypad lockout")); break;

dscKeybusPrintData.cpp:379: Pointer dereference without nearby null check: case 0xBE: stream->print(F("Armed: Partial")); break;

dscKeybusPrintData.cpp:380: Pointer dereference without nearby null check: case 0xBF: stream->print(F("Armed: Special")); break;

dscKeybusPrintData.cpp:382: Pointer dereference without nearby null check: case 0xE5: stream->print(F("Auto-arm cancelled")); break;

dscKeybusPrintData.cpp:383: Pointer dereference without nearby null check: case 0xE6: stream->print(F("Disarmed: Special")); break;

dscKeybusPrintData.cpp:384: Pointer dereference without nearby null check: case 0xE7: stream->print(F("Panel battery trouble")); break;

dscKeybusPrintData.cpp:385: Pointer dereference without nearby null check: case 0xE8: stream->print(F("Panel AC power trouble")); break;

dscKeybusPrintData.cpp:386: Pointer dereference without nearby null check: case 0xE9: stream->print(F("Bell trouble")); break;

dscKeybusPrintData.cpp:387: Pointer dereference without nearby null check: case 0xEA: stream->print(F("Fire zone trouble")); break;

dscKeybusPrintData.cpp:388: Pointer dereference without nearby null check: case 0xEB: stream->print(F("Panel aux supply trouble")); break;

dscKeybusPrintData.cpp:389: Pointer dereference without nearby null check: case 0xEC: stream->print(F("Telephone line trouble")); break;

dscKeybusPrintData.cpp:390: Pointer dereference without nearby null check: case 0xEF: stream->print(F("Panel battery restored")); break;

dscKeybusPrintData.cpp:391: Pointer dereference without nearby null check: case 0xF0: stream->print(F("Panel AC power restored")); break;

dscKeybusPrintData.cpp:392: Pointer dereference without nearby null check: case 0xF1: stream->print(F("Bell restored")); break;

dscKeybusPrintData.cpp:393: Pointer dereference without nearby null check: case 0xF2: stream->print(F("Fire zone trouble restored")); break;

dscKeybusPrintData.cpp:394: Pointer dereference without nearby null check: case 0xF3: stream->print(F("Panel aux supply restored")); break;

dscKeybusPrintData.cpp:395: Pointer dereference without nearby null check: case 0xF4: stream->print(F("Telephone line restored")); break;

dscKeybusPrintData.cpp:396: Pointer dereference without nearby null check: case 0xF7: stream->print(F("Phone 1 FTC")); break;

dscKeybusPrintData.cpp:397: Pointer dereference without nearby null check: case 0xF8: stream->print(F("Phone 2 FTC")); break;

dscKeybusPrintData.cpp:398: Pointer dereference without nearby null check: case 0xF9: stream->print(F("Event buffer threshold")); break; //75% full since last DLS upload

dscKeybusPrintData.cpp:399: Pointer dereference without nearby null check: case 0xFA: stream->print(F("DLS lead-in")); break;

dscKeybusPrintData.cpp:400: Pointer dereference without nearby null check: case 0xFB: stream->print(F("DLS lead-out")); break;

dscKeybusPrintData.cpp:401: Pointer dereference without nearby null check: case 0xFE: stream->print(F("Periodic test transmission")); break;

dscKeybusPrintData.cpp:402: Pointer dereference without nearby null check: case 0xFF: stream->print(F("System test")); break;

dscKeybusPrintData.cpp:417: Pointer dereference without nearby null check: stream->print("Zone alarm: ");

dscKeybusPrintData.cpp:432: Pointer dereference without nearby null check: stream->print("Zone alarm restored: ");

dscKeybusPrintData.cpp:447: Pointer dereference without nearby null check: stream->print("Zone tamper: ");

dscKeybusPrintData.cpp:461: Pointer dereference without nearby null check: stream->print("Zone tamper restored: ");

dscKeybusPrintData.cpp:476: Pointer dereference without nearby null check: stream->print(F("Armed: "));

dscKeybusPrintData.cpp:491: Pointer dereference without nearby null check: stream->print(F("Disarmed: "));

dscKeybusPrintData.cpp:496: Pointer dereference without nearby null check: stream->print("Unknown data");

dscKeybusPrintData.cpp:527: Pointer dereference without nearby null check: case 0x03: stream->print(F("Cross zone alarm")); return;

dscKeybusPrintData.cpp:528: Pointer dereference without nearby null check: case 0x04: stream->print(F("Delinquency alarm")); return;

dscKeybusPrintData.cpp:529: Pointer dereference without nearby null check: case 0x05: stream->print(F("Late to close")); return;

dscKeybusPrintData.cpp:531: Pointer dereference without nearby null check: case 0x29: stream->print(F("Downloading forced answer")); return;

dscKeybusPrintData.cpp:532: Pointer dereference without nearby null check: case 0x2B: stream->print(F("Armed: Auto-arm")); return;

dscKeybusPrintData.cpp:537: Pointer dereference without nearby null check: case 0xAC: stream->print(F("Exit installer programming")); return;

dscKeybusPrintData.cpp:538: Pointer dereference without nearby null check: case 0xAD: stream->print(F("Enter installer programming")); return;

dscKeybusPrintData.cpp:539: Pointer dereference without nearby null check: case 0xAE: stream->print(F("Walk test end")); return;

dscKeybusPrintData.cpp:540: Pointer dereference without nearby null check: case 0xAF: stream->print(F("Walk test begin")); return;

dscKeybusPrintData.cpp:542: Pointer dereference without nearby null check: case 0xD0: stream->print(F("Command output 4")); return;

dscKeybusPrintData.cpp:543: Pointer dereference without nearby null check: case 0xD1: stream->print(F("Exit fault pre-alert")); return;

dscKeybusPrintData.cpp:544: Pointer dereference without nearby null check: case 0xD2: stream->print(F("Armed: Entry delay")); return;

dscKeybusPrintData.cpp:545: Pointer dereference without nearby null check: case 0xD3: stream->print(F("Downlook remote trigger")); return;

dscKeybusPrintData.cpp:569: Pointer dereference without nearby null check: stream->print(F("Zone battery restored: "));

dscKeybusPrintData.cpp:578: Pointer dereference without nearby null check: stream->print(F("Zone battery low: "));

dscKeybusPrintData.cpp:594: Pointer dereference without nearby null check: stream->print(F("Zone fault restored: "));

dscKeybusPrintData.cpp:610: Pointer dereference without nearby null check: stream->print(F("Zone fault: "));

dscKeybusPrintData.cpp:624: Pointer dereference without nearby null check: stream->print(F("Zone bypassed: "));

dscKeybusPrintData.cpp:629: Pointer dereference without nearby null check: stream->print("Unknown data");

dscKeybusPrintData.cpp:665: Pointer dereference without nearby null check: case 0x2A: stream->print(F("Quick exit")); return;

dscKeybusPrintData.cpp:666: Pointer dereference without nearby null check: case 0x63: stream->print(F("Keybus fault restored")); return;

dscKeybusPrintData.cpp:667: Pointer dereference without nearby null check: case 0x64: stream->print(F("Keybus fault")); return;

dscKeybusPrintData.cpp:668: Pointer dereference without nearby null check: case 0x66: stream->print(F("*1: Zone bypass")); return;

dscKeybusPrintData.cpp:670: Pointer dereference without nearby null check: case 0x8C: stream->print(F("Cold start")); return;

dscKeybusPrintData.cpp:671: Pointer dereference without nearby null check: case 0x8D: stream->print(F("Warm start")); return;

dscKeybusPrintData.cpp:672: Pointer dereference without nearby null check: case 0x8E: stream->print(F("Panel factory default")); return;

dscKeybusPrintData.cpp:673: Pointer dereference without nearby null check: case 0x91: stream->print(F("Swinger shutdown")); return;

dscKeybusPrintData.cpp:674: Pointer dereference without nearby null check: case 0x93: stream->print(F("Disarmed: Keyswitch")); return;

dscKeybusPrintData.cpp:675: Pointer dereference without nearby null check: case 0x96: stream->print(F("Armed: Keyswitch")); return;

dscKeybusPrintData.cpp:676: Pointer dereference without nearby null check: case 0x97: stream->print(F("Armed: Keypad away")); return;

dscKeybusPrintData.cpp:677: Pointer dereference without nearby null check: case 0x98: stream->print(F("Armed: Quick-arm")); return;

dscKeybusPrintData.cpp:678: Pointer dereference without nearby null check: case 0x99: stream->print(F("Activate stay/away zones")); return;

dscKeybusPrintData.cpp:679: Pointer dereference without nearby null check: case 0x9A: stream->print(F("Armed: Stay")); return;

dscKeybusPrintData.cpp:680: Pointer dereference without nearby null check: case 0x9B: stream->print(F("Armed: Away")); return;

dscKeybusPrintData.cpp:681: Pointer dereference without nearby null check: case 0x9C: stream->print(F("Armed: No entry delay")); return;

dscKeybusPrintData.cpp:689: Pointer dereference without nearby null check: case 0xFF: stream->print(F("Zone expander trouble: 1")); return;

dscKeybusPrintData.cpp:696: Pointer dereference without nearby null check: stream->print(F("Command output: "));

dscKeybusPrintData.cpp:706: Pointer dereference without nearby null check: stream->print(F("*1: "));

dscKeybusPrintData.cpp:716: Pointer dereference without nearby null check: stream->print(F("*5: "));

dscKeybusPrintData.cpp:740: Pointer dereference without nearby null check: stream->print(F("*6: "));

dscKeybusPrintData.cpp:754: Pointer dereference without nearby null check: stream->print(F("Keypad restored: Slot "));

dscKeybusPrintData.cpp:768: Pointer dereference without nearby null check: stream->print(F("Keypad trouble: Slot "));

dscKeybusPrintData.cpp:777: Pointer dereference without nearby null check: stream->print(F("Zone expander restored: "));

dscKeybusPrintData.cpp:782: Pointer dereference without nearby null check: stream->print("Unknown data");

dscKeybusPrintData.cpp:805: Pointer dereference without nearby null check: case 0x05: stream->print(F("PC/RF5132: Supervisory restored")); return;

dscKeybusPrintData.cpp:806: Pointer dereference without nearby null check: case 0x06: stream->print(F("PC/RF5132: Supervisory trouble")); return;

dscKeybusPrintData.cpp:807: Pointer dereference without nearby null check: case 0x09: stream->print(F("PC5204: Supervisory restored")); return;

dscKeybusPrintData.cpp:808: Pointer dereference without nearby null check: case 0x0A: stream->print(F("PC5204: Supervisory trouble")); return;

dscKeybusPrintData.cpp:809: Pointer dereference without nearby null check: case 0x17: stream->print(F("Zone expander restored: 7")); return;

dscKeybusPrintData.cpp:810: Pointer dereference without nearby null check: case 0x18: stream->print(F("Zone expander trouble: 7")); return;

dscKeybusPrintData.cpp:815: Pointer dereference without nearby null check: case 0x41: stream->print(F("PC/RF5132: Tamper restored")); return;

dscKeybusPrintData.cpp:816: Pointer dereference without nearby null check: case 0x42: stream->print(F("PC/RF5132: Tamper")); return;

dscKeybusPrintData.cpp:817: Pointer dereference without nearby null check: case 0x43: stream->print(F("PC5208: Tamper restored")); return;

dscKeybusPrintData.cpp:818: Pointer dereference without nearby null check: case 0x44: stream->print(F("PC5208: Tamper")); return;

dscKeybusPrintData.cpp:819: Pointer dereference without nearby null check: case 0x45: stream->print(F("PC5204: Tamper restored")); return;

dscKeybusPrintData.cpp:820: Pointer dereference without nearby null check: case 0x46: stream->print(F("PC5204: Tamper")); return;

dscKeybusPrintData.cpp:821: Pointer dereference without nearby null check: case 0x51: stream->print(F("Zone expander tamper restored: 7")); return;

dscKeybusPrintData.cpp:822: Pointer dereference without nearby null check: case 0x52: stream->print(F("Zone expander tamper: 7")); return;

dscKeybusPrintData.cpp:823: Pointer dereference without nearby null check: case 0xB3: stream->print(F("PC5204: Battery restored")); return;

dscKeybusPrintData.cpp:824: Pointer dereference without nearby null check: case 0xB4: stream->print(F("PC5204: Battery trouble")); return;

dscKeybusPrintData.cpp:825: Pointer dereference without nearby null check: case 0xB5: stream->print(F("PC5204: Aux supply restored")); return;

dscKeybusPrintData.cpp:826: Pointer dereference without nearby null check: case 0xB6: stream->print(F("PC5204: Aux supply trouble")); return;

dscKeybusPrintData.cpp:827: Pointer dereference without nearby null check: case 0xB7: stream->print(F("PC5204: Output 1 restored")); return;

dscKeybusPrintData.cpp:828: Pointer dereference without nearby null check: case 0xB8: stream->print(F("PC5204: Output 1 trouble")); return;

dscKeybusPrintData.cpp:829: Pointer dereference without nearby null check: case 0xFF: stream->print(F("Extended status")); return;

dscKeybusPrintData.cpp:836: Pointer dereference without nearby null check: stream->print(F("Zone expander trouble: "));

dscKeybusPrintData.cpp:845: Pointer dereference without nearby null check: stream->print(F("Keypad tamper restored: "));

dscKeybusPrintData.cpp:854: Pointer dereference without nearby null check: stream->print(F("Keypad tamper: "));

dscKeybusPrintData.cpp:869: Pointer dereference without nearby null check: stream->print(F("Zone expander tamper restored: "));

dscKeybusPrintData.cpp:884: Pointer dereference without nearby null check: stream->print(F("Zone expander tamper: "));

dscKeybusPrintData.cpp:889: Pointer dereference without nearby null check: stream->print("Unknown data");

dscKeybusPrintData.cpp:911: Pointer dereference without nearby null check: case 0x86: stream->print(F("Periodic test with trouble")); return;

dscKeybusPrintData.cpp:912: Pointer dereference without nearby null check: case 0x87: stream->print(F("Exit fault")); return;

dscKeybusPrintData.cpp:913: Pointer dereference without nearby null check: case 0x89: stream->print(F("Alarm cancelled")); return;

dscKeybusPrintData.cpp:917: Pointer dereference without nearby null check: stream->print("Zone alarm: ");

dscKeybusPrintData.cpp:922: Pointer dereference without nearby null check: stream->print("Zone alarm restored: ");

dscKeybusPrintData.cpp:927: Pointer dereference without nearby null check: stream->print("Zone tamper: ");

dscKeybusPrintData.cpp:932: Pointer dereference without nearby null check: stream->print("Zone tamper restored: ");

dscKeybusPrintData.cpp:936: Pointer dereference without nearby null check: else stream->print("Unknown data");

dscKeybusPrintData.cpp:958: Pointer dereference without nearby null check: stream->print(F("Armed: "));

dscKeybusPrintData.cpp:970: Pointer dereference without nearby null check: stream->print(F("Disarmed: "));

dscKeybusPrintData.cpp:975: Pointer dereference without nearby null check: stream->print("Unknown data");

dscKeybusPrintData.cpp:991: Pointer dereference without nearby null check: case 0xC0: stream->print(F("TLink com fault")); return;

dscKeybusPrintData.cpp:992: Pointer dereference without nearby null check: case 0xC2: stream->print(F("Tlink network fault")); return;

dscKeybusPrintData.cpp:993: Pointer dereference without nearby null check: case 0xC4: stream->print(F("TLink receiver trouble")); return;

dscKeybusPrintData.cpp:994: Pointer dereference without nearby null check: case 0xC5: stream->print(F("TLink receiver restored")); return;

dscKeybusPrintData.cpp:1013: Pointer dereference without nearby null check: case 0x80: stream->print(F("Trouble acknowledged")); return;

dscKeybusPrintData.cpp:1014: Pointer dereference without nearby null check: case 0x81: stream->print(F("RF delinquency trouble")); return;

dscKeybusPrintData.cpp:1015: Pointer dereference without nearby null check: case 0x82: stream->print(F("RF delinquency restore")); return;

dscKeybusPrintData.cpp:1039: Pointer dereference without nearby null check: stream->print(F("*1: "));

dscKeybusPrintData.cpp:1051: Pointer dereference without nearby null check: stream->print(F("*2: "));

dscKeybusPrintData.cpp:1058: Pointer dereference without nearby null check: stream->print(F("*2: "));

dscKeybusPrintData.cpp:1070: Pointer dereference without nearby null check: stream->print(F("*3: "));

dscKeybusPrintData.cpp:1077: Pointer dereference without nearby null check: stream->print(F("*3: "));

dscKeybusPrintData.cpp:1082: Pointer dereference without nearby null check: stream->print("Unknown data");

dscKeybusPrintData.cpp:1117: Pointer dereference without nearby null check: stream->print(F("*5: "));

dscKeybusPrintData.cpp:1130: Pointer dereference without nearby null check: stream->print(F("*6: "));

dscKeybusPrintData.cpp:1135: Pointer dereference without nearby null check: stream->print("Unknown data");

dscKeybusPrintData.cpp:1150: Pointer dereference without nearby null check: case 0xF1: stream->print(F("System reset transmission")); return;

dscKeybusPrintData.cpp:1193: Pointer dereference without nearby null check: stream->print(" | ");

dscKeybusPrintData.cpp:1240: Pointer dereference without nearby null check: if (!zoneLights) stream->print(F("none"));

dscKeybusPrintData.cpp:1268: Pointer dereference without nearby null check: stream->print(F("Module supervision query"));

dscKeybusPrintData.cpp:1304: Pointer dereference without nearby null check: stream->print(F("Panel version: v"));

dscKeybusPrintData.cpp:1305: Pointer dereference without nearby null check: stream->print(panelData[3] >> 4);

dscKeybusPrintData.cpp:1306: Pointer dereference without nearby null check: stream->print(".");

dscKeybusPrintData.cpp:1307: Pointer dereference without nearby null check: stream->print(panelData[3] & 0x0F);

dscKeybusPrintData.cpp:1310: Pointer dereference without nearby null check: stream->print(F(" | Zone wiring: "));

dscKeybusPrintData.cpp:1312: Pointer dereference without nearby null check: case 0x01: stream->print(F("NC ")); break;

dscKeybusPrintData.cpp:1313: Pointer dereference without nearby null check: case 0x02: stream->print(F("EOL ")); break;

dscKeybusPrintData.cpp:1314: Pointer dereference without nearby null check: case 0x03: stream->print(F("DEOL ")); break;

dscKeybusPrintData.cpp:1318: Pointer dereference without nearby null check: stream->print(F("| Code length: "));

dscKeybusPrintData.cpp:1319: Pointer dereference without nearby null check: if (panelData[4] & 0x08) stream->print(F("6"));

dscKeybusPrintData.cpp:1320: Pointer dereference without nearby null check: else stream->print(F("4"));

dscKeybusPrintData.cpp:1321: Pointer dereference without nearby null check: stream->print(F(" digits "));

dscKeybusPrintData.cpp:1324: Pointer dereference without nearby null check: stream->print(F("| *8 programming: "));

dscKeybusPrintData.cpp:1325: Pointer dereference without nearby null check: if (panelData[4] & 0x10) stream->print(F("no "));

dscKeybusPrintData.cpp:1326: Pointer dereference without nearby null check: else stream->print(F("yes "));

dscKeybusPrintData.cpp:1370: Pointer dereference without nearby null check: stream->print(F("Verify keypad Fire/Auxiliary/Panic"));

dscKeybusPrintData.cpp:1400: Pointer dereference without nearby null check: stream->print(F("Zone expander query: "));


```
");  
dscKeybusPrintData.cpp:1401: Pointer dereference without nearby null check: stream->print(expander);  
dscKeybusPrintData.cpp:1441: Pointer dereference without nearby null check: stream->print(F(" | Zones 1-8 open: "));  
dscKeybusPrintData.cpp:1468: Pointer dereference without nearby null check: stream->print(F(" | Zones 9-16 open: "));  
dscKeybusPrintData.cpp:1490: Pointer dereference without nearby null check: stream->print(F(" | Zones 17-24 open: "));  
dscKeybusPrintData.cpp:1512: Pointer dereference without nearby null check: stream->print(F(" | Zones 25-32 open: "));  
dscKeybusPrintData.cpp:1532: Pointer dereference without nearby null check: stream->print(F("Wireless module  
query"));  
dscKeybusPrintData.cpp:1559: Pointer dereference without nearby null check: stream->print(F("Module tamper query"));  
dscKeybusPrintData.cpp:1577: Pointer dereference without nearby null check: stream->print(F("Wireless key query"));  
dscKeybusPrintData.cpp:1596: Pointer dereference without nearby null check: stream->print(F("Module status query"));  
dscKeybusPrintData.cpp:1637: Pointer dereference without nearby null check: stream->print(F(" | Zones 1-32 flashing:  
"));  
dscKeybusPrintData.cpp:1708: Pointer dereference without nearby null check: stream->print(F("LCD display: "));  
dscKeybusPrintData.cpp:1710: Pointer dereference without nearby null check: if (panelData[2] <= 0x63)  
stream->print("0");  
dscKeybusPrintData.cpp:1711: Pointer dereference without nearby null check: if (panelData[2] <= 0x09)  
stream->print("0");  
dscKeybusPrintData.cpp:1712: Pointer dereference without nearby null check: stream->print(panelData[2], DEC);  
dscKeybusPrintData.cpp:1716: Pointer dereference without nearby null check: stream->print(panelData[panelByte] >> 4,  
HEX);  
dscKeybusPrintData.cpp:1717: Pointer dereference without nearby null check: stream->print(panelData[panelByte] &  
0x0F, HEX);  
dscKeybusPrintData.cpp:1736: Pointer dereference without nearby null check: stream->print(F("LCD keypad data  
query"));  
dscKeybusPrintData.cpp:1857: Pointer dereference without nearby null check: stream->print(F("PGM outputs enabled:  
"));  
dscKeybusPrintData.cpp:1858: Pointer dereference without nearby null check: if (panelData[2] == 0 && panelData[3] ==  
0) stream->print(F("none "));  
dscKeybusPrintData.cpp:1865: Pointer dereference without nearby null check: if (panelData[3] & 0x04)  
stream->print(F(" | Midnight "));  
dscKeybusPrintData.cpp:1866: Pointer dereference without nearby null check: if (panelData[3] & 0x08)  
stream->print(F(" | Battery check"));  
dscKeybusPrintData.cpp:1927: Pointer dereference without nearby null check: stream->print(F("Module programming  
entry: "));  
dscKeybusPrintData.cpp:1930: Pointer dereference without nearby null check: stream->print(F("Module programming  
entry"));  
dscKeybusPrintData.cpp:1959: Pointer dereference without nearby null check: stream->print(F("Module programming  
request: "));  
dscKeybusPrintData.cpp:1962: Pointer dereference without nearby null check: stream->print(F("Module programming  
request"));  
dscKeybusPrintData.cpp:1979: Pointer dereference without nearby null check: stream->print(F("DLS query"));  
dscKeybusPrintData.cpp:2014: Pointer dereference without nearby null check: stream->print(F(" | Timestamp"));  
dscKeybusPrintData.cpp:2018: Pointer dereference without nearby null check: stream->print(" | ");  
dscKeybusPrintData.cpp:2020: Pointer dereference without nearby null check: case 0x01: printPartition();  
printNumberSpace(1); stream->print("| "); break;  
dscKeybusPrintData.cpp:2021: Pointer dereference without nearby null check: case 0x02: printPartition();  
printNumberSpace(2); stream->print("| "); break;  
dscKeybusPrintData.cpp:2069: Pointer dereference without nearby null check: stream->print(F("Event: "));  
dscKeybusPrintData.cpp:2070: Pointer dereference without nearby null check: if (panelData[7] < 10) stream->print("00");  
dscKeybusPrintData.cpp:2071: Pointer dereference without nearby null check: else if (panelData[7] < 100)  
stream->print("0");  
dscKeybusPrintData.cpp:2072: Pointer dereference without nearby null check: stream->print(panelData[7]);
```

dscKeybusPrintData.cpp:2073: Pointer dereference without nearby null check: stream->print(" | ");

dscKeybusPrintData.cpp:2077: Pointer dereference without nearby null check: stream->print(" | ");

dscKeybusPrintData.cpp:2079: Pointer dereference without nearby null check: case 0x01: printPartition(); printNumberSpace(1); stream->print(" | "); break;

dscKeybusPrintData.cpp:2080: Pointer dereference without nearby null check: case 0x02: printPartition(); printNumberSpace(2); stream->print(" | "); break;

dscKeybusPrintData.cpp:2117: Pointer dereference without nearby null check: stream->print(F("Enabled zones 1-32 | Partition 1: "));

dscKeybusPrintData.cpp:2120: Pointer dereference without nearby null check: stream->print(F(" | Partition 2: "));

dscKeybusPrintData.cpp:2145: Pointer dereference without nearby null check: stream->print(F("Bell: "));

dscKeybusPrintData.cpp:2146: Pointer dereference without nearby null check: if (bitRead(panelData[2], 5)) stream->print(F("on"));

dscKeybusPrintData.cpp:2147: Pointer dereference without nearby null check: else stream->print(F("off"));

dscKeybusPrintData.cpp:2175: Pointer dereference without nearby null check: stream->print(F("TLM: "));

dscKeybusPrintData.cpp:2176: Pointer dereference without nearby null check: if (panelData[2] & 0x10) stream->print(F("trouble/attempt"));

dscKeybusPrintData.cpp:2177: Pointer dereference without nearby null check: else stream->print(F("available/disabled"));

dscKeybusPrintData.cpp:2179: Pointer dereference without nearby null check: if (panelData[2] & 0x08) stream->print(F(" | Dialer call attempt"));

dscKeybusPrintData.cpp:2180: Pointer dereference without nearby null check: if (panelData[2] & 0x20) stream->print(F(" | Keypad lockout"));

dscKeybusPrintData.cpp:2226: Pointer dereference without nearby null check: stream->print(F(" [Byte 2/0x"));

dscKeybusPrintData.cpp:2227: Pointer dereference without nearby null check: if (panelData[2] < 16) stream->print("0");

dscKeybusPrintData.cpp:2228: Pointer dereference without nearby null check: stream->print(panelData[2], HEX);

dscKeybusPrintData.cpp:2229: Pointer dereference without nearby null check: stream->print(F("] "));

dscKeybusPrintData.cpp:2248: Pointer dereference without nearby null check: stream->print(F("Keypad zone query"));

dscKeybusPrintData.cpp:2268: Pointer dereference without nearby null check: stream->print(F("[CRC Error]);

dscKeybusPrintData.cpp:2300: Pointer dereference without nearby null check: default: stream->print("Unknown data");

dscKeybusPrintData.cpp:2384: Pointer dereference without nearby null check: stream->print(F("Zone expander query: "));

dscKeybusPrintData.cpp:2385: Pointer dereference without nearby null check: stream->print(expander);

dscKeybusPrintData.cpp:2403: Pointer dereference without nearby null check: stream->print(F("Zones 33-40 open: "));

dscKeybusPrintData.cpp:2422: Pointer dereference without nearby null check: stream->print(F("Zones 41-48 open: "));

dscKeybusPrintData.cpp:2441: Pointer dereference without nearby null check: stream->print(F("Zones 49-56 open: "));

dscKeybusPrintData.cpp:2460: Pointer dereference without nearby null check: stream->print(F("Zones 57-64 open: "));

dscKeybusPrintData.cpp:2491: Pointer dereference without nearby null check: stream->print(F(" | Zones 1-32 flashing: "));

dscKeybusPrintData.cpp:2524: Pointer dereference without nearby null check: stream->print(F(" | Zones 33-64 flashing: "));

dscKeybusPrintData.cpp:2583: Pointer dereference without nearby null check: stream->print(F("Partitions in alarm: "));

dscKeybusPrintData.cpp:2586: Pointer dereference without nearby null check: if (panelData[6] & 0x08) stream->print(F(" | Loss of system time "));

dscKeybusPrintData.cpp:2587: Pointer dereference without nearby null check: if (panelData[6] & 0x10) stream->print(F(" | AC power trouble "));

dscKeybusPrintData.cpp:2588: Pointer dereference without nearby null check: if (panelData[6] & 0x40) stream->print(F(" | Fail to communicate "));

dscKeybusPrintData.cpp:2589: Pointer dereference without nearby null check: if (panelData[6] & 0x80) stream->print(F(" | Fire alarm "));

dscKeybusPrintData.cpp:2659: Pointer dereference without nearby null check: stream->print(F(" | Enabled zones 1-32: "));

dscKeybusPrintData.cpp:2687: Pointer dereference without nearby null check: stream->print(F(" | Enabled zones 33-64: "));

dscKeybusPrintData.cpp:2748: Pointer dereference without nearby null check: if (panelData[2] == 0) stream->print(" | ");

dscKeybusPrintData.cpp:2750: Pointer dereference without nearby null check: stream->print(F(" | Partition "));

dscKeybusPrintData.cpp:2752: Pointer dereference without nearby null check: stream->print("| ");

dscKeybusPrintData.cpp:2805: Pointer dereference without nearby null check: stream->print(F("Event: "));

dscKeybusPrintData.cpp:2806: Pointer dereference without nearby null check: if (eventNumber < 10) stream->print("00");

dscKeybusPrintData.cpp:2807: Pointer dereference without nearby null check: else if (eventNumber < 100) stream->print("0");

dscKeybusPrintData.cpp:2808: Pointer dereference without nearby null check: stream->print(eventNumber);

dscKeybusPrintData.cpp:2809: Pointer dereference without nearby null check: stream->print(" | ");

dscKeybusPrintData.cpp:2812: Pointer dereference without nearby null check: stream->print(" | ");

dscKeybusPrintData.cpp:2817: Pointer dereference without nearby null check: stream->print("| ");

dscKeybusPrintData.cpp:2832: Pointer dereference without nearby null check: case 0xFF: stream->print(F("No entry"));

return;

dscKeybusPrintData.cpp:2849: Pointer dereference without nearby null check: stream->print(F("[Keypad] Fire alarm"));

dscKeybusPrintData.cpp:2863: Pointer dereference without nearby null check: stream->print(F("[Keypad] Auxiliary alarm"));

dscKeybusPrintData.cpp:2877: Pointer dereference without nearby null check: stream->print(F("[Keypad] Panic alarm"));

dscKeybusPrintData.cpp:2950: Pointer dereference without nearby null check: if (printedMessage) stream->print("| ");

dscKeybusPrintData.cpp:2951: Pointer dereference without nearby null check: stream->print(F("Keypad on partition: "));

dscKeybusPrintData.cpp:2953: Pointer dereference without nearby null check: stream->print(F("going idle"));

dscKeybusPrintData.cpp:2959: Pointer dereference without nearby null check: if (printedMessage) stream->print("| ");

dscKeybusPrintData.cpp:2960: Pointer dereference without nearby null check: stream->print(F("Zone expander notification: "));

dscKeybusPrintData.cpp:2972: Pointer dereference without nearby null check: if (printedMessage) stream->print("| ");

dscKeybusPrintData.cpp:2973: Pointer dereference without nearby null check: stream->print(F("Module tamper notification "));

dscKeybusPrintData.cpp:2979: Pointer dereference without nearby null check: if (printedMessage) stream->print("| ");

dscKeybusPrintData.cpp:2980: Pointer dereference without nearby null check: stream->print(F("Wireless module battery notification "));

dscKeybusPrintData.cpp:2987: Pointer dereference without nearby null check: if (printedMessage) stream->print("| ");

dscKeybusPrintData.cpp:2988: Pointer dereference without nearby null check: stream->print(F("Wireless notification "));

dscKeybusPrintData.cpp:2994: Pointer dereference without nearby null check: if (printedMessage) stream->print("| ");

dscKeybusPrintData.cpp:2995: Pointer dereference without nearby null check: stream->print(F("Keypad zone notification "));

dscKeybusPrintData.cpp:3001: Pointer dereference without nearby null check: if (printedMessage) stream->print("| ");

dscKeybusPrintData.cpp:3002: Pointer dereference without nearby null check: stream->print(F("Module status notification "));

dscKeybusPrintData.cpp:3008: Pointer dereference without nearby null check: if (printedMessage) stream->print("| ");

dscKeybusPrintData.cpp:3009: Pointer dereference without nearby null check: stream->print(F("Wireless key notification "));

dscKeybusPrintData.cpp:3015: Pointer dereference without nearby null check: if (printedMessage) stream->print("| ");

dscKeybusPrintData.cpp:3016: Pointer dereference without nearby null check: stream->print(F("Keypad notification "));

dscKeybusPrintData.cpp:3021: Pointer dereference without nearby null check: if (printedMessage) stream->print("| ");

dscKeybusPrintData.cpp:3022: Pointer dereference without nearby null check: stream->print(F("Door chime broadcast "));

dscKeybusPrintData.cpp:3027: Pointer dereference without nearby null check: if (printedMessage) stream->print("| ");

dscKeybusPrintData.cpp:3028: Pointer dereference without nearby null check: stream->print(F("Zone label broadcast "));

dscKeybusPrintData.cpp:3093: Pointer dereference without nearby null check: stream->print(F("Keypad slots: "));

dscKeybusPrintData.cpp:3098: Pointer dereference without nearby null check: stream->print(F("| Zone expander: "));

dscKeybusPrintData.cpp:3112: Pointer dereference without nearby null check: if ((moduleData[5] & 0x0C) == 0) stream->print(F("| PC/RF5132 "));

dscKeybusPrintData.cpp:3113: Pointer dereference without nearby null check: if ((moduleData[5] & 0x03) == 0) stream->print(F("| PC5208 "));

dscKeybusPrintData.cpp:3114: Pointer dereference without nearby null check: if ((moduleData[6] & 0xC0) == 0) stream->print(F("| PC5204 "));

dscKeybusPrintData.cpp:3142: Pointer dereference without nearby null check: stream->print(F("Wireless module "));

dscKeybusPrintData.cpp:3145: Pointer dereference without nearby null check: stream->print(F("| Battery low zones: "));

dscKeybusPrintData.cpp:3150: Pointer dereference without nearby null check: stream->print(F("| Battery restored zones: "));

dscKeybusPrintData.cpp:3228: Pointer dereference without nearby null check: stream->print(F("Keypad tamper: Slot "));

dscKeybusPrintData.cpp:3234: Pointer dereference without nearby null check: if (printedMessage) stream->print("| ");

dscKeybusPrintData.cpp:3235: Pointer dereference without nearby null check: stream->print(F("Keypad tamper restored: Slot "));

dscKeybusPrintData.cpp:3241: Pointer dereference without nearby null check: if (printedMessage) stream->print("| ");

dscKeybusPrintData.cpp:3242: Pointer dereference without nearby null check: stream->print(F("Module tamper: Slot "));

dscKeybusPrintData.cpp:3249: Pointer dereference without nearby null check: if (printedMessage) stream->print("| ");

dscKeybusPrintData.cpp:3250: Pointer dereference without nearby null check: stream->print(F("Module tamper restored: Slot "));

dscKeybusPrintData.cpp:3257: Pointer dereference without nearby null check: if (printedMessage) stream->print("| ");

dscKeybusPrintData.cpp:3258: Pointer dereference without nearby null check: stream->print(F("RF5132: Tamper "));

dscKeybusPrintData.cpp:3263: Pointer dereference without nearby null check: if (printedMessage) stream->print("| ");

dscKeybusPrintData.cpp:3264: Pointer dereference without nearby null check: stream->print(F("RF5132: Tamper restored "));

dscKeybusPrintData.cpp:3269: Pointer dereference without nearby null check: if (printedMessage) stream->print("| ");

dscKeybusPrintData.cpp:3270: Pointer dereference without nearby null check: stream->print(F("PC5208: Tamper "));

dscKeybusPrintData.cpp:3275: Pointer dereference without nearby null check: if (printedMessage) stream->print("| ");

dscKeybusPrintData.cpp:3276: Pointer dereference without nearby null check: stream->print(F("PC5208: Tamper restored "));

dscKeybusPrintData.cpp:3281: Pointer dereference without nearby null check: if (printedMessage) stream->print("| ");

dscKeybusPrintData.cpp:3282: Pointer dereference without nearby null check: stream->print(F("PC5204: Tamper "));

dscKeybusPrintData.cpp:3287: Pointer dereference without nearby null check: if (printedMessage) stream->print("| ");

dscKeybusPrintData.cpp:3288: Pointer dereference without nearby null check: stream->print(F("PC5204: Tamper restored "));

dscKeybusPrintData.cpp:3327: Pointer dereference without nearby null check: stream->print(F("Wireless key low battery: "));

dscKeybusPrintData.cpp:3333: Pointer dereference without nearby null check: if (printedMessage) stream->print(F("| "));

dscKeybusPrintData.cpp:3334: Pointer dereference without nearby null check: stream->print(F("Wireless key battery restored: "));

dscKeybusPrintData.cpp:3384: Pointer dereference without nearby null check: stream->print(F("PC5204: Battery restored "));

dscKeybusPrintData.cpp:3389: Pointer dereference without nearby null check: if (printedMessage) stream->print("| ");

dscKeybusPrintData.cpp:3390: Pointer dereference without nearby null check: stream->print(F("PC5204: Battery trouble "));

dscKeybusPrintData.cpp:3395: Pointer dereference without nearby null check: if (printedMessage) stream->print("| ");

dscKeybusPrintData.cpp:3396: Pointer dereference without nearby null check: stream->print(F("PC5204: AC power restored "));

dscKeybusPrintData.cpp:3401: Pointer dereference without nearby null check: if (printedMessage) stream->print("| ");

dscKeybusPrintData.cpp:3402: Pointer dereference without nearby null check: stream->print(F("PC5204: AC power trouble "));

dscKeybusPrintData.cpp:3407: Pointer dereference without nearby null check: if (printedMessage) stream->print("| ");

dscKeybusPrintData.cpp:3408: Pointer dereference without nearby null check: stream->print(F("PC5204: Output 1 restored "));

dscKeybusPrintData.cpp:3413: Pointer dereference without nearby null check: if (printedMessage) stream->print("| ");

dscKeybusPrintData.cpp:3414: Pointer dereference without nearby null check: stream->print(F("PC5204: Output 1

```
trouble ");
dscKeybusPrintData.cpp:3445: Pointer dereference without nearby null check: stream->print(F("LCD keypad data entry:
"));
dscKeybusPrintData.cpp:3447: Pointer dereference without nearby null check: if (moduleData[2] <= 0x63)
stream->print("0");
dscKeybusPrintData.cpp:3448: Pointer dereference without nearby null check: if (moduleData[2] <= 0x09)
stream->print("0");
dscKeybusPrintData.cpp:3449: Pointer dereference without nearby null check: stream->print(moduleData[2], DEC);
dscKeybusPrintData.cpp:3453: Pointer dereference without nearby null check: stream->print(moduleData[moduleByte]
>> 4, HEX);
dscKeybusPrintData.cpp:3454: Pointer dereference without nearby null check: stream->print(moduleData[moduleByte] &
0x0F, HEX);
dscKeybusPrintData.cpp:3466: Pointer dereference without nearby null check: stream->print(F("Module programming
response"));
dscKeybusPrintData.cpp:3495: Pointer dereference without nearby null check: stream->print(F("Keypad "));
dscKeybusPrintData.cpp:3501: Pointer dereference without nearby null check: stream->print(F("Slot "));
dscKeybusPrintData.cpp:3504: Pointer dereference without nearby null check: else stream->print(F(" | Slot "));
dscKeybusPrintData.cpp:3506: Pointer dereference without nearby null check: stream->print(moduleByte - 1);
dscKeybusPrintData.cpp:3507: Pointer dereference without nearby null check: if ((slotData & 0x03) == 0x03 &&
(slotData & 0x30) == 0) stream->print(F(": Zone open"));
dscKeybusPrintData.cpp:3508: Pointer dereference without nearby null check: if ((slotData & 0x03) == 0 && (slotData &
0x30) == 0x30) stream->print(F(": Zone closed"));
dscKeybusPrintData.cpp:3591: Pointer dereference without nearby null check: stream->print(F("[Digit] "));
dscKeybusPrintData.cpp:3594: Pointer dereference without nearby null check: stream->print(F("Partition "));
dscKeybusPrintData.cpp:3596: Pointer dereference without nearby null check: stream->print(F("Key: "));
dscKeybusPrintData.cpp:3621: Pointer dereference without nearby null check: case 0x28: stream->print(F("* ")); break;
dscKeybusPrintData.cpp:3622: Pointer dereference without nearby null check: case 0x2D: stream->print(F("# ")); break;
dscKeybusPrintData.cpp:3623: Pointer dereference without nearby null check: case 0x46: stream->print(F("Wireless key
disarm ")); break; //isn't send if wls keys uses access codes 17-32 is enabled
dscKeybusPrintData.cpp:3624: Pointer dereference without nearby null check: case 0x52: stream->print(F("Identified
voice prompt help ")); break;
dscKeybusPrintData.cpp:3625: Pointer dereference without nearby null check: case 0x6E: stream->print(F("Global away
arm ")); break;
dscKeybusPrintData.cpp:3626: Pointer dereference without nearby null check: case 0x70: stream->print(F("Command
output 3 ")); break;
dscKeybusPrintData.cpp:3627: Pointer dereference without nearby null check: case 0x7A: stream->print(F("Time and
date programming ")); break;
dscKeybusPrintData.cpp:3628: Pointer dereference without nearby null check: case 0x75: stream->print(F("Entered
*1/*2/*3 ? ")); break;
dscKeybusPrintData.cpp:3629: Pointer dereference without nearby null check: case 0x82: stream->print(F("Enter "));
break;
dscKeybusPrintData.cpp:3630: Pointer dereference without nearby null check: case 0x87: stream->print(F("Right arrow
")); break;
dscKeybusPrintData.cpp:3631: Pointer dereference without nearby null check: case 0x88: stream->print(F("Left arrow
")); break;
dscKeybusPrintData.cpp:3632: Pointer dereference without nearby null check: case 0x8D: stream->print(F("Bypass
recall ")); break;
dscKeybusPrintData.cpp:3633: Pointer dereference without nearby null check: case 0x93: stream->print(F("Recall
bypass group ")); break;
dscKeybusPrintData.cpp:3634: Pointer dereference without nearby null check: case 0x94: stream->print(F("Global label
broadcast ")); break;
dscKeybusPrintData.cpp:3635: Pointer dereference without nearby null check: case 0x99: stream->print(F("Function key
[25] Future Use ")); break;
```

dscKeybusPrintData.cpp:3636: Pointer dereference without nearby null check: case 0xA5: stream->print(F("Receive data ")); break;

dscKeybusPrintData.cpp:3637: Pointer dereference without nearby null check: case 0xAA: stream->print(F("Submit data ")); break;

dscKeybusPrintData.cpp:3638: Pointer dereference without nearby null check: case 0xAF: stream->print(F("Arm: Stay ")); break;

dscKeybusPrintData.cpp:3639: Pointer dereference without nearby null check: case 0xB1: stream->print(F("Arm: Away ")); break;

dscKeybusPrintData.cpp:3640: Pointer dereference without nearby null check: case 0xB6: stream->print(F("Arm: No entry delay ")); break;

dscKeybusPrintData.cpp:3641: Pointer dereference without nearby null check: case 0xBB: stream->print(F("Door chime configuration ")); break;

dscKeybusPrintData.cpp:3642: Pointer dereference without nearby null check: case 0xBC: stream->print(F("**6 System test ")); break;

dscKeybusPrintData.cpp:3643: Pointer dereference without nearby null check: case 0xC3: stream->print(F("**1 Zone bypass programming ")); break;

dscKeybusPrintData.cpp:3644: Pointer dereference without nearby null check: case 0xC4: stream->print(F("**2 Trouble menu ")); break;

dscKeybusPrintData.cpp:3645: Pointer dereference without nearby null check: case 0xC9: stream->print(F("**3 Alarm memory display ")); break;

dscKeybusPrintData.cpp:3646: Pointer dereference without nearby null check: case 0xCE: stream->print(F("**5 Programming ")); break;

dscKeybusPrintData.cpp:3647: Pointer dereference without nearby null check: case 0xD0: stream->print(F("**6 Programming ")); break;

dscKeybusPrintData.cpp:3648: Pointer dereference without nearby null check: case 0xD5: stream->print(F("Command output 1 ")); break;

dscKeybusPrintData.cpp:3649: Pointer dereference without nearby null check: case 0xDA: stream->print(F("Reset / Command output 2 ")); break;

dscKeybusPrintData.cpp:3650: Pointer dereference without nearby null check: case 0xDF: stream->print(F("Global stay arm ")); break;

dscKeybusPrintData.cpp:3651: Pointer dereference without nearby null check: case 0xE1: stream->print(F("Quick exit ")); break;

dscKeybusPrintData.cpp:3652: Pointer dereference without nearby null check: case 0xE6: stream->print(F("Activate stay/away zones ")); break;

dscKeybusPrintData.cpp:3653: Pointer dereference without nearby null check: case 0xEB: stream->print(F("LCD pixel test ")); break;

dscKeybusPrintData.cpp:3654: Pointer dereference without nearby null check: case 0xEC: stream->print(F("Command output 4 ")); break;

dscKeybusPrintData.cpp:3655: Pointer dereference without nearby null check: case 0xF2: stream->print(F("Global disarm ")); break; // Possible overlap with "General voice prompt help"

dscKeybusPrintData.cpp:3656: Pointer dereference without nearby null check: case 0xF7: stream->print(F("Menu navigation ")); break;

dscKeybusPrintData.cpp:3668: Pointer dereference without nearby null check: stream->print(F("Zone expander: "));

dscKeybusPrintData.cpp:3687: Pointer dereference without nearby null check: stream->print(F("| Zones changed: "));

dscKeybusPrintData.cpp:3697: Pointer dereference without nearby null check: case 0: stream->print(F("open ")); break;

dscKeybusPrintData.cpp:3698: Pointer dereference without nearby null check: case 1: stream->print(F("closed ")); break;

dscKeybusPrintData.cpp:3699: Pointer dereference without nearby null check: case 2: stream->print(F("tamper ")); break;

dscKeybusPrintData.cpp:3700: Pointer dereference without nearby null check: case 3: stream->print(F("open (D/EOL) ")); break;

dscKeybusPrintData.cpp:3748: Pointer dereference without nearby null check: case 0x11: stream->print(F("RF5132")); break; //section 804 verified on pc1832 and pc5020

dscKeybusPrintData.cpp:3749: Pointer dereference without nearby null check: case 0x14: stream->print(F("RF5400")); break; //section 801 not verified

dscKeybusPrintData.cpp:3750: Pointer dereference without nearby null check: case 0x15: stream->print(F("RF5936")); break; //section 802 not verified

dscKeybusPrintData.cpp:3751: Pointer dereference without nearby null check: case 0x16: stream->print(F("LINKS2X50")); break; //section 803 not verified

dscKeybusPrintData.cpp:3752: Pointer dereference without nearby null check: case 0x17: stream->print(F("PC5108L")); break; //section 806 not verified

dscKeybusPrintData.cpp:3753: Pointer dereference without nearby null check: case 0x19: stream->print(F("RF5100")); break; //section 805 not verified

dscKeybusPrintData.cpp:3754: Pointer dereference without nearby null check: case 0x31: stream->print(F("*5 user")); break; /*5 access codes verified on pc1832 and pc5020

dscKeybusPrintData.cpp:3755: Pointer dereference without nearby null check: default: stream->print("Unknown data");

dscKeybusPrintData.cpp:3757: Pointer dereference without nearby null check: stream->print(" | ");

dscKeybusPrintData.cpp:3758: Pointer dereference without nearby null check: if (panelByte3 < 16) stream->print("0");

dscKeybusPrintData.cpp:3759: Pointer dereference without nearby null check: stream->print(panelByte3, HEX);

dscKeybusPrintData.cpp:3769: Pointer dereference without nearby null check: if (partitionCount > startPartition) stream->print(" | ");

dscKeybusPrintData.cpp:3772: Pointer dereference without nearby null check: stream->print(partitionCount);

dscKeybusPrintData.cpp:3773: Pointer dereference without nearby null check: stream->print(": ");

dscKeybusPrintData.cpp:3775: Pointer dereference without nearby null check: if (panelData[statusByte] == 0 || panelData[statusByte] == 0xC7 || panelData[statusByte] == 0xFF) stream->print(F("disabled"));

dscKeybusPrintData.cpp:3798: Pointer dereference without nearby null check: if (dscYear3 >= 7) stream->print(F("19"));

dscKeybusPrintData.cpp:3799: Pointer dereference without nearby null check: else stream->print(F("20"));

dscKeybusPrintData.cpp:3800: Pointer dereference without nearby null check: stream->print(dscYear3);

dscKeybusPrintData.cpp:3801: Pointer dereference without nearby null check: stream->print(dscYear4, HEX);

dscKeybusPrintData.cpp:3802: Pointer dereference without nearby null check: stream->print(". ");

dscKeybusPrintData.cpp:3803: Pointer dereference without nearby null check: if (dscMonth < 10) stream->print("0");

dscKeybusPrintData.cpp:3804: Pointer dereference without nearby null check: stream->print(dscMonth);

dscKeybusPrintData.cpp:3805: Pointer dereference without nearby null check: stream->print(". ");

dscKeybusPrintData.cpp:3806: Pointer dereference without nearby null check: if (dscDay < 10) stream->print("0");

dscKeybusPrintData.cpp:3807: Pointer dereference without nearby null check: stream->print(dscDay);

dscKeybusPrintData.cpp:3808: Pointer dereference without nearby null check: stream->print(" ");

dscKeybusPrintData.cpp:3809: Pointer dereference without nearby null check: if (dscHour < 10) stream->print("0");

dscKeybusPrintData.cpp:3810: Pointer dereference without nearby null check: stream->print(dscHour);

dscKeybusPrintData.cpp:3811: Pointer dereference without nearby null check: stream->print(F(":"));

dscKeybusPrintData.cpp:3812: Pointer dereference without nearby null check: if (dscMinute < 10) stream->print("0");

dscKeybusPrintData.cpp:3813: Pointer dereference without nearby null check: stream->print(dscMinute);

dscKeybusPrintData.cpp:3833: Pointer dereference without nearby null check: case 40: stream->print(F("Master ")); break;

dscKeybusPrintData.cpp:3834: Pointer dereference without nearby null check: default: stream->print(F("Access ")); break;

dscKeybusPrintData.cpp:3836: Pointer dereference without nearby null check: stream->print(F("code "));

dscKeybusPrintData.cpp:3837: Pointer dereference without nearby null check: stream->print(dscCode);

dscKeybusPrintData.cpp:3847: Pointer dereference without nearby null check: stream->print(F("| Beep: "));

dscKeybusPrintData.cpp:3848: Pointer dereference without nearby null check: stream->print(panelData[panelByte] / 2);

dscKeybusPrintData.cpp:3849: Pointer dereference without nearby null check: stream->print(F(" beeps"));

dscKeybusPrintData.cpp:3859: Pointer dereference without nearby null check: stream->print(F("| Tone: "));

dscKeybusPrintData.cpp:3863: Pointer dereference without nearby null check: stream->print(F("none"));

dscKeybusPrintData.cpp:3868: Pointer dereference without nearby null check: stream->print(F("constant tone "));

dscKeybusPrintData.cpp:3873: Pointer dereference without nearby null check: if (printedMessage) stream->print("| ");

dscKeybusPrintData.cpp:3874: Pointer dereference without nearby null check: stream->print((panelData[panelByte] & 0x70) >> 4);

dscKeybusPrintData.cpp:3875: Pointer dereference without nearby null check: stream->print(F(" beep "));

dscKeybusPrintData.cpp:3879: Pointer dereference without nearby null check: stream->print("| ");

dscKeybusPrintData.cpp:3880: Pointer dereference without nearby null check: stream->print(panelData[panelByte] & 0x0F);

dscKeybusPrintData.cpp:3881: Pointer dereference without nearby null check: stream->print(F("s interval"));

dscKeybusPrintData.cpp:3892: Pointer dereference without nearby null check: stream->print(F("| Buzzer: "));

dscKeybusPrintData.cpp:3893: Pointer dereference without nearby null check: stream->print(panelData[panelByte]);

dscKeybusPrintData.cpp:3894: Pointer dereference without nearby null check: stream->print("s");

dscKeybusPrintData.cpp:3910: Pointer dereference without nearby null check: stream->print((zoneBit + startZone) + ((panelByte - inputByte) * 8));

dscKeybusPrintData.cpp:3911: Pointer dereference without nearby null check: stream->print(" ");

dscKeybusPrintData.cpp:3916: Pointer dereference without nearby null check: if (!zonesEnabled && panelData[0] != 0x0A && panelData[0] != 0x0F) stream->print(F("none"));

dscKeybusPrintData.cpp:3923: Pointer dereference without nearby null check: stream->print(F("Partition "));

dscKeybusPrintData.cpp:3928: Pointer dereference without nearby null check: stream->print("Unknown data");

dscKeybusPrintData.cpp:3933: Pointer dereference without nearby null check: stream->print(F("Status lights: "));

dscKeybusPrintData.cpp:3938: Pointer dereference without nearby null check: if (lowerRange) stream->print(F(" | Zones 1-32 lights: "));

dscKeybusPrintData.cpp:3939: Pointer dereference without nearby null check: else stream->print(F(" | Zones 33-64 lights: "));

dscKeybusPrintData.cpp:3944: Pointer dereference without nearby null check: stream->print(F("| Status lights flashing: "));

dscKeybusPrintData.cpp:3949: Pointer dereference without nearby null check: stream->print(number);

dscKeybusPrintData.cpp:3950: Pointer dereference without nearby null check: stream->print(" ");

dscKeybusPrintData.cpp:3955: Pointer dereference without nearby null check: stream->print(panelData[panelByte] + numberOffset);

dscKeybusPrintData.cpp:3961: Pointer dereference without nearby null check: if (printNone && panelData[panelByte] == 0) stream->print(F("none "));

dscKeybusPrintData.cpp:3966: Pointer dereference without nearby null check: stream->print(startNumber + bitCount);

dscKeybusPrintData.cpp:3967: Pointer dereference without nearby null check: stream->print(" ");

dscKeybusPrintData.cpp:3978: Pointer dereference without nearby null check: if (panelByte == 1) stream->print(panelData[panelByte]); // Prints the stop bit

dscKeybusPrintData.cpp:3981: Pointer dereference without nearby null check: if (mask & panelData[panelByte]) stream->print("1");

dscKeybusPrintData.cpp:3982: Pointer dereference without nearby null check: else stream->print("0");

dscKeybusPrintData.cpp:3985: Pointer dereference without nearby null check: if (printSpaces && (panelByte != panelByteCount - 1 || displayTrailingBits)) stream->print(" ");

dscKeybusPrintData.cpp:3992: Pointer dereference without nearby null check: stream->print(bitRead(panelData[panelByteCount], i));

dscKeybusPrintData.cpp:4002: Pointer dereference without nearby null check: if (moduleByte == 1) stream->print(moduleData[moduleByte]); // Prints the stop bit

dscKeybusPrintData.cpp:4007: Pointer dereference without nearby null check: stream->print(F(".....")); // Hides keypad digits

dscKeybusPrintData.cpp:4010: Pointer dereference without nearby null check: if (mask & moduleData[moduleByte]) stream->print("1");

dscKeybusPrintData.cpp:4011: Pointer dereference without nearby null check: else stream->print("0");

dscKeybusPrintData.cpp:4014: Pointer dereference without nearby null check: if (printSpaces && (moduleByte != moduleByteCount - 1 || displayTrailingBits)) stream->print(" ");

dscKeybusPrintData.cpp:4021: Pointer dereference without nearby null check: stream->print(bitRead(moduleData[moduleByteCount], i));

dscKeybusPrintData.cpp:4030: Pointer dereference without nearby null check: stream->print(F("0x"));

dscKeybusPrintData.cpp:4031: Pointer dereference without nearby null check: if (panelData[0] < 16) stream->print("0");

dscKeybusPrintData.cpp:4032: Pointer dereference without nearby null check: stream->print(panelData[0], HEX);

dscKeybusPrintData.cpp:4035: Pointer dereference without nearby null check: stream->print(".");

dscKeybusPrintData.cpp:4036: Pointer dereference without nearby null check: if (panelData[2] < 16) stream->print("0");

dscKeybusPrintData.cpp:4037: Pointer dereference without nearby null check: stream->print(panelData[2], HEX);

dscKeybusProcessData.cpp:949: Potential pointer usage: byte maxZones = dscZones * 8;

dscKeypad.h:30: Potential pointer usage: Stream* stream;

dscKeypad.h:40: Potential pointer usage: static hw_timer_t* timer1;

dsc_arduino_compatibility.h:48: Potential pointer usage: typedef void* hw_timer_t;

dsc_arduino_compatibility.h:139: Arduino Serial usage: // Global Serial object - only if not already defined by Arduino (replace with ESP_LOGx)

dsc_arduino_compatibility.h:142: Arduino Serial usage: static Stream& Serial = _serial_instance; (replace with ESP_LOGx)

dsc_esp_idf_timer_fix.cpp:20: Potential pointer usage: DSCTimer* timer_instance = static_cast<DSCTimer*>(arg);

dsc_esp_idf_timer_fix.cpp:21: Potential pointer usage: if (timer_instance && timer_instance->callback_func) {

dsc_esp_idf_timer_fix.cpp:22: Potential pointer usage: timer_instance->callback_func();

dsc_esp_idf_timer_fix.cpp:171: Potential pointer usage: uint32_t ticks = microseconds * 80;

dsc_esp_idf_timer_fix.cpp:21: Pointer dereference without nearby null check: if (timer_instance && timer_instance->callback_func) {

dsc_esp_idf_timer_fix.cpp:22: Pointer dereference without nearby null check: timer_instance->callback_func();

dsc_esp_idf_timer_fix.h:45: Potential pointer usage: hw_timer_t* hw_timer_handle;

dsc_keybus.cpp:101: Potential pointer usage: getDSC().init(this->clock_pin_, this->read_pin_, this->write_pin_, this->pc16_pin_);

dsc_keybus.cpp:104: Potential pointer usage: for (auto *trigger : this->system_status_triggers_) {

dsc_keybus.cpp:105: Potential pointer usage: trigger->trigger(STATUS_OFFLINE);

dsc_keybus.cpp:108: Potential pointer usage: this->force_disconnect_ = false;

dsc_keybus.cpp:197: Potential pointer usage: if (!this->force_disconnect_ && getDSC().isHardwareInitialized()) {

dsc_keybus.cpp:211: Potential pointer usage: for (auto *trigger : this->system_status_triggers_) {

dsc_keybus.cpp:212: Potential pointer usage: trigger->trigger(STATUS_ONLINE);

dsc_keybus.cpp:216: Potential pointer usage: for (auto *trigger : this->system_status_triggers_) {

dsc_keybus.cpp:217: Potential pointer usage: trigger->trigger(STATUS_OFFLINE);

dsc_keybus.cpp:262: Potential pointer usage: this->force_disconnect_ = true;

dsc_keybus.cpp:286: Potential pointer usage: ESP_LOGCONFIG(TAG, " Debug level: %u", this->debug_level_);

dsc_keybus.cpp:287: Potential pointer usage: ESP_LOGCONFIG(TAG, " Access code configured: %s", this->access_code_.empty() ? "NO" : "YES");

dsc_keybus.cpp:288: Potential pointer usage: ESP_LOGCONFIG(TAG, " Enable 05 messages: %s", YESNO(this->enable_05_messages_));

dsc_keybus.cpp:101: Pointer dereference without nearby null check: getDSC().init(this->clock_pin_, this->read_pin_, this->write_pin_, this->pc16_pin_);

dsc_keybus.cpp:104: Pointer dereference without nearby null check: for (auto *trigger : this->system_status_triggers_) {

dsc_keybus.cpp:105: Pointer dereference without nearby null check: trigger->trigger(STATUS_OFFLINE);

dsc_keybus.cpp:108: Pointer dereference without nearby null check: this->force_disconnect_ = false;

dsc_keybus.cpp:197: Pointer dereference without nearby null check: if (!this->force_disconnect_ && getDSC().isHardwareInitialized()) {

dsc_keybus.cpp:211: Pointer dereference without nearby null check: for (auto *trigger : this->system_status_triggers_) {

dsc_keybus.cpp:212: Pointer dereference without nearby null check: trigger->trigger(STATUS_ONLINE);

dsc_keybus.cpp:216: Pointer dereference without nearby null check: for (auto *trigger : this->system_status_triggers_) {

dsc_keybus.cpp:217: Pointer dereference without nearby null check: trigger->trigger(STATUS_OFFLINE);

dsc_keybus.cpp:262: Pointer dereference without nearby null check: this->force_disconnect_ = true;

dsc_keybus.cpp:286: Pointer dereference without nearby null check: ESP_LOGCONFIG(TAG, " Debug level: %u", this->debug_level_);

dsc_keybus.cpp:287: Pointer dereference without nearby null check: ESP_LOGCONFIG(TAG, " Access code configured: %s", this->access_code_.empty() ? "NO" : "YES");

dsc_keybus.cpp:288: Pointer dereference without nearby null check: ESP_LOGCONFIG(TAG, " Enable 05 messages: %s", YESNO(this->enable_05_messages_));

dsc_keybus.h:51: Potential pointer usage: DSCKeybusComponent *parent_;

dsc_keybus.h:59: Potential pointer usage: DSCKeybusComponent *parent_;

dsc_keybus.h:67: Potential pointer usage: DSCKeybusComponent *parent_;

dsc_keybus.h:75: Potential pointer usage: DSCKeybusComponent *parent_;

dsc_keybus.h:83: Potential pointer usage: DSCKeybusComponent *parent_;

dsc_keybus.h:91: Potential pointer usage: DSCKeybusComponent *parent_;

dsc_keybus.h:99: Potential pointer usage: DSCKeybusComponent *parent_;

dsc_keybus.h:111: Potential pointer usage: void set_access_code(const std::string &code) { this->access_code_ = code; }

dsc_keybus.h:112: Potential pointer usage: void set_debug_level(uint8_t level) { this->debug_level_ = level; }

dsc_keybus.h:113: Potential pointer usage: void set_enable_05_messages(bool enable) { this->enable_05_messages_ = enable; }

dsc_keybus.h:114: Potential pointer usage: void set_clock_pin(uint8_t pin) { this->clock_pin_ = pin; }

dsc_keybus.h:115: Potential pointer usage: void set_read_pin(uint8_t pin) { this->read_pin_ = pin; }

dsc_keybus.h:116: Potential pointer usage: void set_write_pin(uint8_t pin) { this->write_pin_ = pin; }

dsc_keybus.h:117: Potential pointer usage: void set_pc16_pin(uint8_t pin) { this->pc16_pin_ = pin; }

dsc_keybus.h:120: Arduino setup/loop pattern: void setup() override; (use ESPHome lifecycle)

dsc_keybus.h:121: Arduino setup/loop pattern: void loop() override; (use ESPHome lifecycle)

dsc_keybus.h:141: Potential pointer usage: this->system_status_triggers_.push_back(trigger);

dsc_keybus.h:144: Potential pointer usage: this->partition_status_triggers_.push_back(trigger);

dsc_keybus.h:147: Potential pointer usage: this->partition_msg_triggers_.push_back(trigger);

dsc_keybus.h:150: Potential pointer usage: this->trouble_status_triggers_.push_back(trigger);

dsc_keybus.h:153: Potential pointer usage: this->fire_status_triggers_.push_back(trigger);

dsc_keybus.h:156: Potential pointer usage: this->zone_status_triggers_.push_back(trigger);

dsc_keybus.h:159: Potential pointer usage: this->zone_alarm_triggers_.push_back(trigger);

dsc_keybus.h:163: Potential pointer usage: static constexpr const char *STATUS_PENDING = "pending";

dsc_keybus.h:164: Potential pointer usage: static constexpr const char *STATUS_ARM = "armed_away";

dsc_keybus.h:165: Potential pointer usage: static constexpr const char *STATUS_STAY = "armed_home";

dsc_keybus.h:166: Potential pointer usage: static constexpr const char *STATUS_NIGHT = "armed_night";

dsc_keybus.h:167: Potential pointer usage: static constexpr const char *STATUS_OFF = "disarmed";

dsc_keybus.h:168: Potential pointer usage: static constexpr const char *STATUS_ONLINE = "online";

dsc_keybus.h:169: Potential pointer usage: static constexpr const char *STATUS_OFFLINE = "offline";

dsc_keybus.h:170: Potential pointer usage: static constexpr const char *STATUS_TRIGGERED = "triggered";

dsc_keybus.h:171: Potential pointer usage: static constexpr const char *STATUS_READY = "ready";

dsc_keybus.h:172: Potential pointer usage: static constexpr const char *STATUS_NOT_READY = "unavailable";

dsc_keybus.h:173: Potential pointer usage: static constexpr const char *MSG_ZONE_BYPASS = "zone_bypass_entered";

dsc_keybus.h:174: Potential pointer usage: static constexpr const char *MSG_ARMED_BYPASS = "armed_custom_bypass";

dsc_keybus.h:175: Potential pointer usage: static constexpr const char *MSG_NO_ENTRY_DELAY = "no_entry_delay";

dsc_keybus.h:176: Potential pointer usage: static constexpr const char *MSG_NONE = "no_messages";

dsc_keybus.h:111: Pointer dereference without nearby null check: void set_access_code(const std::string &code) { this->access_code_ = code; }

dsc_keybus.h:112: Pointer dereference without nearby null check: void set_debug_level(uint8_t level) { this->debug_level_ = level; }

dsc_keybus.h:113: Pointer dereference without nearby null check: void set_enable_05_messages(bool enable) { this->enable_05_messages_ = enable; }

dsc_keybus.h:114: Pointer dereference without nearby null check: void set_clock_pin(uint8_t pin) { this->clock_pin_ = pin; }

dsc_keybus.h:115: Pointer dereference without nearby null check: void set_read_pin(uint8_t pin) { this->read_pin_ = pin; }

dsc_keybus.h:116: Pointer dereference without nearby null check: void set_write_pin(uint8_t pin) { this->write_pin_ = pin; }

dsc_keybus.h:117: Pointer dereference without nearby null check: void set_pc16_pin(uint8_t pin) { this->pc16_pin_ = pin; }

dsc_keybus.h:141: Pointer dereference without nearby null check: this->system_status_triggers_.push_back(trigger);

dsc_keybus.h:144: Pointer dereference without nearby null check: this->partition_status_triggers_.push_back(trigger);

dsc_keybus.h:147: Pointer dereference without nearby null check: this->partition_msg_triggers_.push_back(trigger);

dsc_keybus.h:150: Pointer dereference without nearby null check: this->trouble_status_triggers_.push_back(trigger);

dsc_keybus.h:153: Pointer dereference without nearby null check: this->fire_status_triggers_.push_back(trigger);

dsc_keybus.h:156: Pointer dereference without nearby null check: this->zone_status_triggers_.push_back(trigger);

dsc_keybus.h:159: Pointer dereference without nearby null check: this->zone_alarm_triggers_.push_back(trigger);

dsc_wrapper.cpp:42: Potential pointer usage: delete dsc_interface_;

dsc_wrapper.cpp:48: Potential pointer usage: dsc_interface_ = new dscClassicInterface(clockPin, readPin, pc16Pin, writePin);

dsc_wrapper.cpp:50: Potential pointer usage: dsc_interface_ = new dscKeybusInterface(clockPin, readPin, writePin);

dsc_wrapper.cpp:75: Potential pointer usage: dsc_interface_->begin();

dsc_wrapper.cpp:97: Potential pointer usage: dsc_interface_->begin(stream);

dsc_wrapper.cpp:105: Potential pointer usage: return dsc_interface_->loop();

dsc_wrapper.cpp:112: Potential pointer usage: dsc_interface_->resetStatus();

dsc_wrapper.cpp:118: Potential pointer usage: dsc_interface_->stop();

dsc_wrapper.cpp:129: Potential pointer usage: dsc_interface_->write(keys);

dsc_wrapper.cpp:135: Potential pointer usage: return dsc_interface_->statusChanged;

dsc_wrapper.cpp:142: Potential pointer usage: dsc_interface_->statusChanged = value;

dsc_wrapper.cpp:148: Potential pointer usage: return dsc_interface_->bufferOverflow;

dsc_wrapper.cpp:155: Potential pointer usage: dsc_interface_->bufferOverflow = value;

dsc_wrapper.cpp:161: Potential pointer usage: return dsc_interface_->keybusChanged;

dsc_wrapper.cpp:168: Potential pointer usage: dsc_interface_->keybusChanged = value;

dsc_wrapper.cpp:174: Potential pointer usage: return dsc_interface_->keybusConnected;

dsc_wrapper.cpp:181: Potential pointer usage: return dsc_interface_->panelData;

dsc_wrapper.cpp:188: Potential pointer usage: return dsc_interface_->armed[partition];

dsc_wrapper.cpp:195: Potential pointer usage: return dsc_interface_->exitDelay[partition];

dsc_wrapper.cpp:75: Pointer dereference without nearby null check: dsc_interface_->begin();

dsc_wrapper.cpp:97: Pointer dereference without nearby null check: dsc_interface_->begin(stream);

dsc_wrapper.cpp:105: Pointer dereference without nearby null check: return dsc_interface_->loop();

dsc_wrapper.cpp:112: Pointer dereference without nearby null check: dsc_interface_->resetStatus();

dsc_wrapper.cpp:118: Pointer dereference without nearby null check: dsc_interface_->stop();

dsc_wrapper.cpp:129: Pointer dereference without nearby null check: dsc_interface_->write(keys);

dsc_wrapper.cpp:135: Pointer dereference without nearby null check: return dsc_interface_->statusChanged;

dsc_wrapper.cpp:142: Pointer dereference without nearby null check: dsc_interface_->statusChanged = value;

dsc_wrapper.cpp:148: Pointer dereference without nearby null check: return dsc_interface_->bufferOverflow;

dsc_wrapper.cpp:155: Pointer dereference without nearby null check: dsc_interface_->bufferOverflow = value;

dsc_wrapper.cpp:161: Pointer dereference without nearby null check: return dsc_interface_->keybusChanged;

dsc_wrapper.cpp:168: Pointer dereference without nearby null check: dsc_interface_->keybusChanged = value;

dsc_wrapper.cpp:174: Pointer dereference without nearby null check: return dsc_interface_->keybusConnected;

dsc_wrapper.cpp:181: Pointer dereference without nearby null check: return dsc_interface_->panelData;

dsc_wrapper.cpp:188: Pointer dereference without nearby null check: return dsc_interface_->armed[partition];

dsc_wrapper.cpp:195: Pointer dereference without nearby null check: return dsc_interface_->exitDelay[partition];

dsc_wrapper.h:88: Potential pointer usage: dscClassicInterface* dsc_interface_;

dsc_wrapper.h:90: Potential pointer usage: dscKeybusInterface* dsc_interface_;

Recommendations:

- Add null pointer checks before dereferencing any pointer.
- Validate all required config fields at startup.
- Replace Arduino Serial with ESP_LOGx macros.

- Refactor setup/loop to use ESPHome lifecycle methods.
- Ensure all dynamic allocations succeed and objects are initialized before use.
- Remove or guard Arduino-only code with `#ifdef ARDUINO` if ESP-IDF compatibility is required.