Non-conformance Page
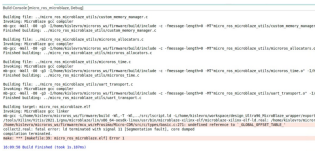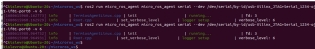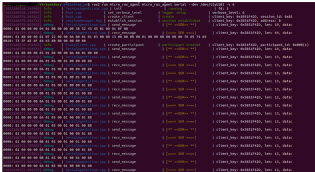
This document contains the non-conformances of our final project with fixes

Problem	Diagram/Photos	Solution	Notes

Project not linking correctly		Add the following args to the colcon.meta file:
		
		``"microcdr": {
		        "cmake-args": [
		            "-DUCDR_PIC=OFF"
		        ]
		    },```
		
		`"microcdr": { "cmake-args": [ "-DUCDR_PIC=OFF" ] },`
		
		And then rebuild the static library
			Link to issue opened in micro-ros github page:
			
			⊙ Building Custom library fails · Issue #619 · micro-ROS/micro_ros_setup

Micro Ros Agen Session not eshtablished (stuck on set verbose level)		Framing in `rmw_uros_set_custom_transport` was disabled, to solve this enable framing:
		
		Add defines:
		
		``#define MICROROS_TRANSPORTS_FRAMING_MODE 1
		#define MICROROS_TRANSPORTS_PACKET_MODE 0
		
		static void microros_thread_custom(void *pvParameters);
		const TickType_t xlsecond = pdMS_TO_TICKS(DELAY_1_SECOND);
		
		/* microRos Prototype */
		bool vitis_transport_open(struct uxrCustomTransport * transport);```
		
		and enable framing:
		
		``74 }
		75 /*--------------------------------------*/
		76 static void microros_thread_custom(void *pvParameters)
		77 {
		78 // Transport
		79 rmw_uros_set_custom_transport(
		80 MICROROS_TRANSPORTS_FRAMING_MODE, // Framing enable here
		81 (void *) NULL, //transport->args
		82 vitis_transport_open,
		83 vitis_transport_close,
		84 vitis_transport_write,
		85 vitis_transport_read);
		86
		87
		88 rcl_allocator_t freeRTOS_allocator = rcutils_get_zero_initialized_all```
			Link to issue:
			
			⊙ Agent run stopped after set_verbose_level · Issue #187 · micro-ROS/micro-ROS-Agent
			
			◉ 4.1. Custom Transports — Vulcanexus 1.0.0 documentation
			
			Important Note:
			
			Run agent first then client

Problem with transport functions
 	
can't see data that was sent
		The XUartPs_Send is non-blocking function so sometimes happened while the UART was sending, XUartPs_Recv was call causing bytes loss. We need to wait until UART finish sending, So we add the following line to the write transport function:
		
		``size_t vitis_transport_write(struct uxrCustomTransport* transport, uint8_t* buf, size_t len, uint8_t * err){
		    int SentCount = 0;
		    u32 LoopCount = 0;
		    /* Block sending the buffer. */
		    // if Sent nothing return 0, otherwise return the length of buf
		    SentCount = XUartPs_Send(&Uart_Ps, buf, len);
		    while(XUartPs_IsSending(&Uart_Ps));
		        LoopCount++;
		    }
		    if (SentCount != len){
		        return 0;
		    }
		    return len;
		}```	