WORTECS

**H2020-ICT-2016-2
RIA**

**Project-ID: 761329
WORTECS
Networking research beyond 5G**

Deliverable 4.1

Optical wireless communication prototype

Contractual Date of Delivery:	**2019, August 23rd**
Actual Date of Delivery:	**2019, September 17th**
Editor(s):	**Bastien Béchadergue (Oledcomm)**
Author(s):	**Ravinder Singh, Dominic O'Brien (University of Oxford), Tamas Weszely (pureLiFi), Bastien Béchadergue (Oledcomm)**
Work package:	**WP4**
Security:	**PU**
Nature:	**Deliverable**
Version:	**version 1.5**
Total number of pages:	**16**

Abstract

This deliverable describes the general architecture of the optical wireless communication (OWC) and fiber wireless fiber (FWF) prototypes with a focus on their different building blocks and interfaces. An instructions manual is also provided for both prototypes to give guidelines on how to interface them with the rest of the WORTECS system and how to safely operate them.

Keyword list

virtual reality ; high quality/low latency ; wireless transmission ; OWC ; FWF ; HW implementation ; operating manual.

Executive Summary

In October 2019, first tests of the full WORTECS system will be carried out. This deliverable D4.1 provides on overview of the architecture of both the optical wireless communication (OWC) and fiber wireless fiber (FWF) prototypes developed.

The optical wireless prototype is composed of two core boards: the baseband and the optical front-end. Section 1.1 describes these boards as well as their interfaces whereas Section 1.2 explains how to connect them together and then to the rest of the WORTECS system.

On the other hand, the architecture of the fiber wireless fiber prototype is detailed in Section 2.1 and its operating manual, including its calibration process, is given in Section 2.2.

Impact on the other Work-packages

WP4: This deliverable will serve as a reference manual to anyone willing to install and manipulate the optical communication prototypes during the tasks dedicated to prototypes testing and users' test acceptance.

List of Authors

First name	Last name	Beneficiary	Email address
Bastien	Béchadergue	OLEDCOMM	bastien.bechadergue@oledcomm.net
Dominic	O'Brien	University of Oxford	dominic.obrien@eng.ox.ac.uk
Ravinder	Singh	University of Oxford	ravinder.singh@eng.ox.ac.uk
Tamas	Weszely	pureLiFi	tamas.weszely@purelifi.com

Document History

First name	Last name	Version	Comments
Bastien	Béchadergue	V1.1	First draft
Ravinder	Singh	V1.2	Fiber-Wireless-Fiber Demonstrator Description and User Manual
Dominic	O'Brien	V1.3	Review of Fiber-Wireless-Fiber Demonstrator Description and User Manual
Tamas	Weszely	V1.4	OWC System Description
Bastien	Béchadergue	V1.5	OWC User Manual and editing

List of Acronyms

Acronym	Meaning
FoV	Field of View
FPGA	Field-Programmable Gate Array
FWF	Fiber Wireless Fiber
LED	Light Emitting Diode
OFE	Optical Front-End
OWC	Optical Wireless Communication
PHY	Physical layer
PLC	Power Line Communication
Rx	Receiver
SFP	Small Form-factor Pluggable
SMA	SubMiniature version A
SoTa	State of The Art
Tx	Transmitter
VR	Virtual Reality
WORTECS	Wireless Optical/Radio TErabit CommunicationS

Table of contents

List of Tables

List of Figures

1   Optical wireless communication prototype

1.1 Prototype description

The optical wireless communication (OWC) prototype is composed of two main building blocks: the baseband and the optical front-end (OFE). For the baseband, there are two architectures available for the interim OWC demonstration: a field-programmable gate array (FPGA) based custom physical layer (PHY) developed for the WORTECS project by pureLiFi and a system based on a powerline communications (PLC), off the shelf baseband. During the demonstration in October 2019, two ZCU111 boards and four PLC based boards will be prepared, along with four OFE boards. The following sections describe the general architecture and interfaces of the PLC and OFE boards.

1.1.1 PLC-based baseband description

1.1.1.1 Overview

The digital baseband interfaces the OFE and the B-COM video-server. The video server is connected via the small form factor pluggable (SFP) interface on the PLC board. The baseband is required to achieve a data-throughput of at least 860Mbps for the video stream *needs*.
On the OFE side, the baseband has two pairs of subminiature version A (SMA) connectors to interface with Oledcomm's OFE.
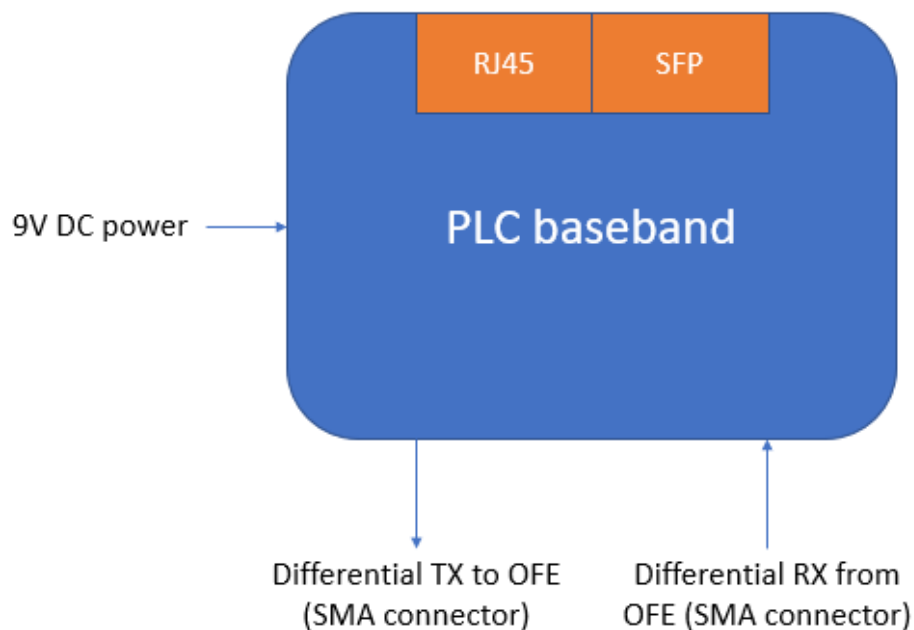Figure 1 below shows the available interfaces on the PLC board.

Figure 1- PLC board interfaces.

1.1.1.2 Features

- 200MHz bandwidth
- Bit loading
- Analogue automatic gain control (AGC) in the receiver
- Maximum throughput of 1700Mbps

1.1.1.3 Interface specification

Interface	Connector type	Notes
Analogue		
OFE Tx	2 x SMA	1V differential signal 50ohm output impedance
OFE Rx	2 x SMA	Maximum 0-10V differential (0-5V on RX+ and RX-) 50ohm input impedance
Power		
Power in	DC jack	9V DC
Data connection		
Data 1	SFP	Interface used with B-COM video server using 10G Ethernet over copper
Data 2	RJ45	Alternative connection for testing using 1G Ethernet

Table 1 – PLC baseband interface specifications.

1.1.2 OFE description

The OFE is to be interfaced with the baseband in order to transmit and received OWC data signals. Figure 2 shows a block diagram view of the OFE.
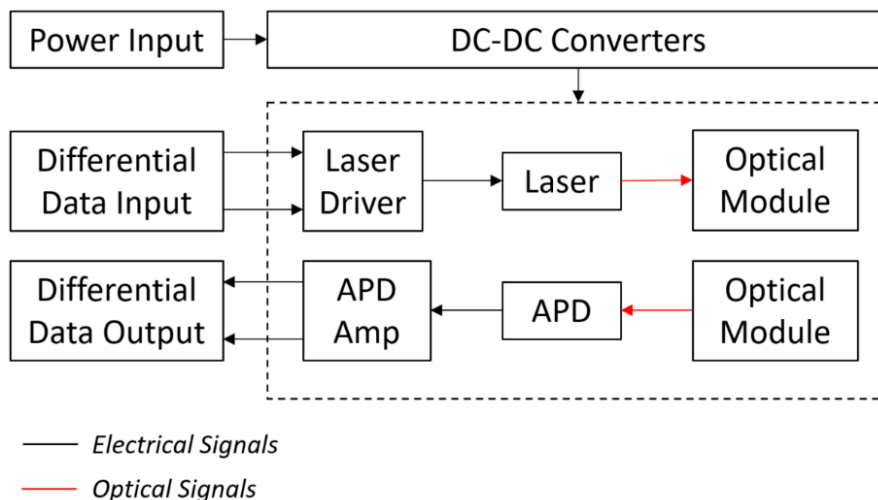

Figure 2 - OFE board architecture and interface description.

Physically, an OFE corresponds to a single electronic and optical board with the following interfaces:

Interface	Connector type	Notes
Analogue		
Baseband Tx	2 x SMA	1V differential signal 50ohm output impedance
Baseband Rx	2 x SMA	Maximum 0-10V differential (0-5V on RX+ and RX-) 50ohm input impedance
Power		
Power in	USB-C	5V DC

Table 2 – OFE interface specifications.

1.2 User Manual

During the October 2019 tests, four OFE will be brought and interfaced with either PLC-based basebands or FPGA-based basebands. In order to install and use an OWC prototype with the PLC baseband, the following steps can be followed.

Step 1: Baseband/OFE Assembly

Each OWC prototype is composed of four main components: two baseband boards (named BB_1 and BB_2) and two OFE boards (named OFE_1 and OFE_2). As detailed in Section 1.1.1.1 and shown on Figure 3, the baseband boards have 6 interfaces:

-   Two transmitter outputs named Tx_BB+ and Tx_BB- (male SMA connectors),

-   Two receiver inputs named Rx_BB+ and Rx_BB- (male SMA connectors),

-   One SFP input/output named DATA_BB,

-   One power supply input named PWR_BB.

The baseband boards also embed two status light-emitting diodes (LED):

-   LED_ETH which shows if the connection of the baseband board with the data source (e.g. a data server) is correctly established,

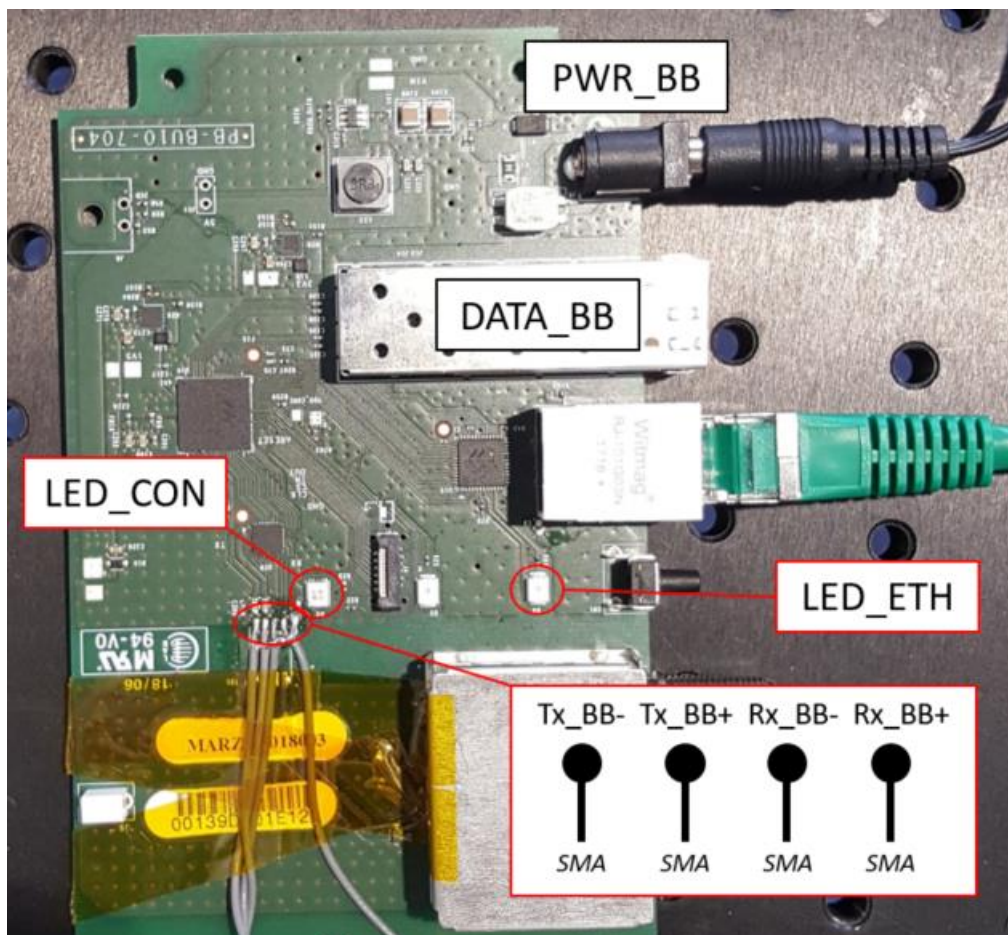-   LED_CON which shows if the OWC link is correctly established.

Figure 3 – Top view of the baseband board with the description of the interfaces.

Similarly, the OFE boards have the 5 interfaces shown on Figure 4:

-   Two transmitter inputs named Tx_OFE+ and Tx_OFE- (male SMA connectors),

-   Two receiver outputs named Rx_OFE+ and Rx_OFE- (male SMA connectors),

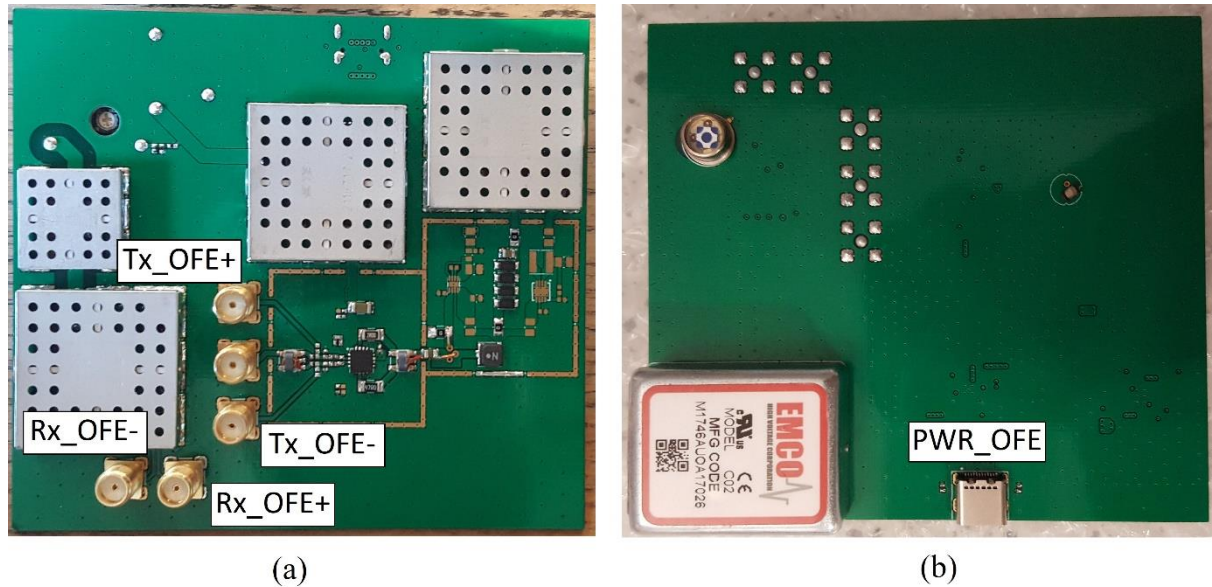- One power supply input named PWR_OFE (female USB-C connector).

(a)   (b)

Figure 4 – (a) Bottom and (b) top view of the OFE board with the description of the interfaces.

Each baseband board is to be assembled with a single OFE board, that is:

- BB_1 is to be connected to OFE_1,

- BB_2 is to be connected to OFE_2.

The connection between a baseband board and an OFE board must respect the pinout table shown on Table 3.

Baseband input/output	Corresponding OFE input/output
Tx_BB+	Tx_OFE+
Tx_BB-	Tx_OFE-
Rx_BB+	Rx_OFE+
Rx_BB-	Rx_OFE-

Table 3 – Baseband/OFE connection pinout.

Step 2: Connection to the Data Server

The BB_1/OFE_1 assembly is to be connected to the data server through the DATA_BB input/output of the BB_1 board. Simply plug an SFP copper cable from the DATA_BB SFP port of BB_1 to the data port of the server. If the connection is correctly established, then the status LED LED_ETH blinks in green.

Step 3: Connection to the Virtual Reality Helmet

Similarly, the BB_2/OFE_2 assembly is to be connected to the virtual reality (VR) helmet through the DATA_BB input/output of the BB_2 board. Simply plug an SFP copper cable from the DATA_BB SFP port of BB_2 to the data port of the VR helmet. If the connection is correctly established, then the status LED LED_ETH blinks in green.

Step 4: Relative Positioning of the Baseband/OFE Pairs

Once the baseband/OFE pairs are connected to the data server and VR helmet, both OFE boards must be positioned face to face at around 10 cm from the other,. **Given this low distance of communication, the OWC access point should probably not be installed on the ceiling but on a dedicated demonstration bench as, for example, shown on Figure 5.**

Step 5: Powering Up

Once the whole setup is installed, it can be powered up. The four boards must power up separately following this order: OFE_1, BB_1, OFE_2, BB_2:

- In order to power-up an OFE board, a male USB-C cable, connected on its other end to an USB charger, must be plugged to the PWR_OFE port of the board.

- In order to power-up a baseband board, the power cable provided with the board must be connected to the power network on its first end and to the PWR_BB port of the board on its other end.

Step 6: Checking Connectivity

Once the whole setup is installed and powered-up, the VR data transmission can be launched. The connectivity between both ends of the OWC prototype can be verified by checking the colour of the STATUS_LED (see Figure 3) on both baseband boards:

- If both LED are off, there is no connectivity between both ends of the OWC prototype.

- If both LED are blinking in orange, the connection between both ends of the OWC prototype is established but the link quality is degraded. In this case, it is recommended to align the OFE boards and move them closer to each other.

- If both LED are blinking in green, the connection between both ends of the OWC prototype is established with a good quality.

Figure 5 – View of the OWC prototype setup.

2   Fiber wireless fiber prototype
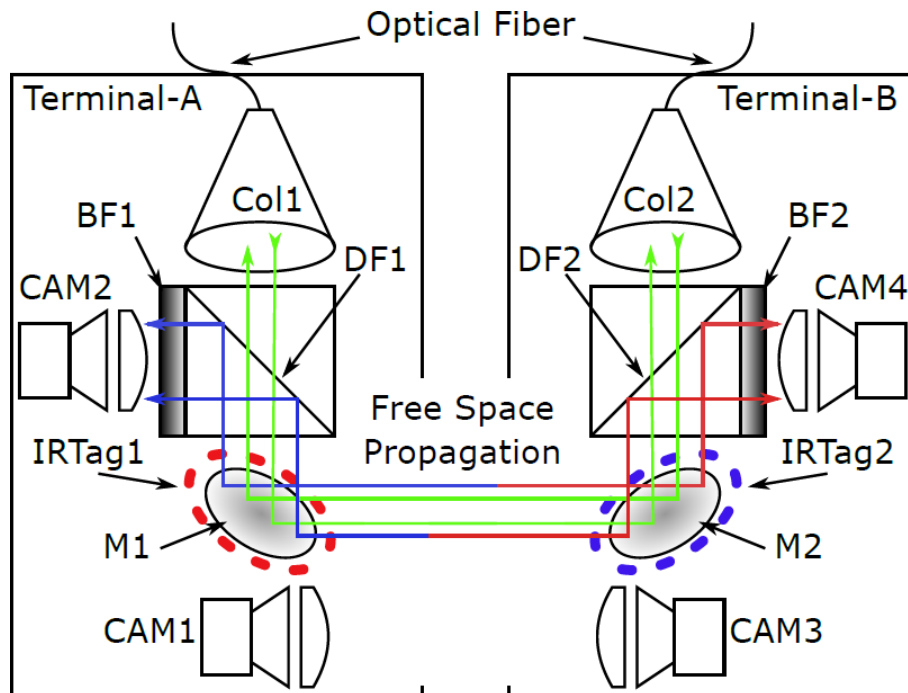
2.1 System description

Figure 6 – Block diagram of FWF system

Figure 6 shows the demonstrator architecture and Figure 7 shows the assembled units. Some parametric and notational details are shown in Table 4. Figure 6 shows that this is a co-design of a narrow-beam optical wireless communication system and an optical localisation and tracking system. More information on the state-of-the-art (SoTA) fiber wireless fiber (FWF) systems can be found in D3.1.

The green lines shown in Figure 6 represent a narrow 1550 nm beam of light, which carries the communication information. This is light from an optical fiber based transmitter, which is connected to terminal A via an optical fiber. This light is collimated at Terminal-A, and steered towards Terminal-B with the use of a dual-axis mirror from OptoTune. The OptoTune mirror provides ±25° mechanical tilt, resulting in ±50° optical deflection. At Terminal-B, another OptoTune mirror steers the beam towards the collimator, which focuses the beam and launches it back into the optical fiber. A optical fiber based receiver then decodes the optical signal.

Notation	Details
BF1	Bandpass Filter-1, Central Wavelength 800 nm
BF2	Bandpass Filter-2, Central Wavelength 900 nm
Col1, Col2	Collimator-1 and Collimator-2
CAM1, CAM3	PixyCams with 50° horizontal and 30° vertical half Field-of-View (FoV)
CAM2, CAM4	PixyCams with 4° horizontal and 2.2° vertical half Field-of-View (FoV)
DF1,DF2	Dichroic Filter 1 and 2, cut-off at 900 nm
IRTag1	890 nm Infrared beacon for tracking
IRTag2	800 nm Infrared beacon for tracking
M1, M2	OptoTune mirrors for beam-steering

Table 4 – FWF Demonstrator Notation and Parameters.

In order to steer the optical beam from Terminal-A to Terminal-B, and vice-versa, accurate location information of the opposite terminal is required. Therefore, a localisation and tracking system is required. As detailed in D3.1 and as shown in Figure 6, digital tracking cameras (pixycam) based localisation and tracking system has been developed. On each terminal there is a wide FoV camera (CAM1 at Terminal-A and CAM3 at Terminal-B) which is used to locate the IRTag on the opposite terminal with low resolution. The mirror is then steered to the appropriate direction. A fine-tracking camera with narrow FoV (CAM2 and CAM4 on A and B respectively) is placed on the terminal so the centre of its FoV tracks with the angular position of the steering mirror, and this is then used to locate the IRTag on the opposite terminal. The mirror is then steered once more, so that the Tag centre is centred on the fine-tracking camera. In practice calibration is required to account for alignment offsets of the tracking system with the communications system. The IRTags are circular rings formed by infrared LEDs, with a central wavelength of 800 nm and 890 nm for each terminal. The BF1 and BF2 optical filters prevent interference between the tracking systems at each terminal while DF1 and DF2 prevent interference between the communication and tracking links.
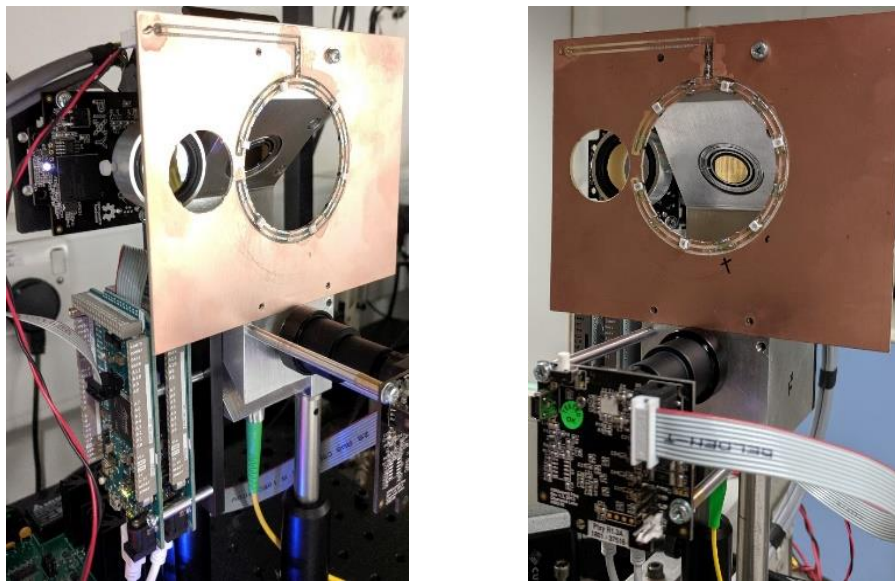

Figure 7 – Assembled transmitter (left) and receiver (right) units.

The characteristics of the system are shown in Table 5.

Operating Field of View	+/- 25 degrees horizontally and vertically
Tracking system resolution (theoretical)	0.014 degrees horizontally and Vertically (corresponding to 1 camera pixel)
Tracking system resolution (measured)	0.021 degrees horizontally and 0.014 degrees Vertically (corresponding to 1 camera pixel)
Communications wavelength	1300-1600 nm (bidirectional)
Fibre input/output	SMF 28- APC connectors
Free space beam diameter	1.12 mm (theoretical) at Collimator output
Transmission power	Limited to <7dBm (Class 1 operation) by control of laser power
Tracking beacon wavelength	800 nm terminal A and 890 nm terminal B

Table 5 – System features and specification.

2.2 User manual

MATLAB Codes:

HandHeldPM.m – This code acquires power from the hand-held power meter.
RotationStageESP300.m – This code provides remote control of the ESP300 universal motion driver which control the rotation stage.
ReadArduinoSerialAve.m – This code acquires the tracking data from the pixycam via Arduino board.
Fit_ellipse.m – This code provides elliptical fit to the detected LED ring for tracking.
WORTECS_D2V1_Tx_Align.m – This code is used to align the Tx unit with Rx unit.
WORTECS_D2V1_Rx_Align.m – This code is used to align the Rx unit with Tx unit.
OptoTuneMirrorAlignTxOnly.m – This code is used for the calibration of the Tx unit via manual alignment.
OptoTuneMirrorAlignRxOnly.m – This code is used for the calibration of the Rx unit via manual alignment.

Powering up:

1. Follow the OptoTune mirror driver manual and power up the mirror and drivers on both Tx and Rx Side.
2. Connect the LED rings to power supply and set the voltage to +5 V. At this voltage level the current drawn should be approximately 100mA.
3. The Arduino boards are powered by USB hub which requires +5V/+12V DC supply depending on which hub (7/10 port) is used.
4. Pixycams are powered by I^2C connection to the Arduino boards.

Calibration Steps:

Note: The aberration correction requires bore-sight calibration. Ensure that the both the units are powered-up.

1. Place the units facing each other, roughly at bore-sight and same height, at 3-4m distance.
2. Open "WORTECS_D2V1_Tx_Align.m" code and set HCaltx and VCaltx to 0. Run "WORTECS_D2V1_Tx_Align.m" code to roughly point the laser beam at the Rx unit. Obtain the mirror angles (horizontal and vertical) from the command window. You will use these angles in "OptoTuneMirrorAlignRxOnly.m" code for 2D scanning the Rx mirror to obtain the optimal alignment angle.
3. Open "WORTECS_D2V1_Rx_Align.m" code and set HCalRx and VCalRx to 0. Run "WORTECS_D2V1_Rx_Align.m" code to roughly align the Rx unit with the Tx unit. Obtain the mirror angles (horizontal and vertical) from the command window. You will use these angles in "OptoTuneMirrorAlignRxOnly.m" code for 2D scanning the Rx mirror to obtain the optimal alignment angle.
4. Using infrared detector card, check the beam manually at the the Rx side. The beam should be pointing close to the centre of the Rx mirror and hence the centre of Rx collimator. We have to next manually correct the Tx mirror to point the beam to the centre of the Rx mirror.
5. Open "OptoTuneMirrorAlignRxOnly.m".
6. Set line 76-77 in "OptoTuneMirrorAlignRxOnly.m" by adding the Rx mirror angles noted in step 3:
    a. ScanVectorRxH = (ScanSolidAngle:-ScanStep:-ScanSolidAngle)+(Use Rx Horizontal angle here);
    b. ScanVectorRxV = (ScanSolidAngle:-ScanStep:-ScanSolidAngle)+(Use Rx Vertical angle here);
7. Set the Tx mirror angles at line 78-79 in "OptoTuneMirrorAlignRxOnly.m" using the values noted in step 4 as:
    a. fprintf(otmTx, 'angleA = x.xxdeg');pause(3e-3)
    b. fprintf(otmTx, 'angleB = x.xxdeg');pause(3e-3)
8. Adjust the angleA and angleB in step 7 and partly run the "OptoTuneMirrorAlignRxOnly.m" code to line 91 (the start of the FOR loop). Manually check the beam position on the Rx mirror and collimator.
9. Repeat step 8 until the beam is centred on the Rx mirror and hence the Rx collimator.
10. Next adjust NPP in code "OptoTuneMirrorAlignRxOnly.m". This is number of scanning points in one plane. Therefore, the total points to be scanned will be NPP^2. For the first 2D scan usually NPP=2^5 is sufficient.

11. Next adjust ScanStep in "OptoTuneMirrorAlignRxOnly.m". This is the scan step in degrees. Usually step of 0.1 is sufficient in the first scan.

12. Now run the code and obtain the new Rx pointing angles from the command window. This process could take 15-20 minutes.

13. After the completion of step 12, adjust the Rx angles to the new values following step 6. Set ScanStep to 0.01 and rerun the "OptoTuneMirrorAlignRxOnly.m" code. After this second 2D scan, we should have fairly optimal angles for the Rx mirror.

14. Next carry out a 2D scan at the Tx side. To do this open "OptoTuneMirrorAlignTxOnly.m" and adjust the angles in line 76-77 and 78-79 similar to the step 6 & 7. Set ScanStep = 0.01 and NPP = 2^3. This is usually sufficient. Run the code "OptoTuneMirrorAlignTxOnly.m" and obtain the optimal Tx pointing angles.

15. Next repeat step 13 to get the optimal pointing angle for Rx mirror.

16. Now we have the optimal pointing angles for the Tx and Rx for alignment at ~ bore sight. We should now adjust the parameters HCaltx, VCaltx in code "" and HCalRx, VCalRx in code "" to automatically align the communication system using the tracking system.

17. Once step 16 is complete, you can use code "RunAling.m" for automatic alignment. This completes the calibration process. You should repeat steps 1-17, if you start to receive low powers after you make physical changes to the FWF terminals.

18. Next you can move one of the terminal around and re-run the "RunAlign.m" code the re-align the terminal.