

# OPI TrueSense Kit Windows Software Development Kit

---

## *User's Guide*

Version	Date	Author	Comments
1.10	20131122	mpeng	1. Added brief explanation of new commands/interface
1.00	20130503	mpeng	1. Original

## Introduction

This document describes the software development kit that will allow a user to write software that can access the OPI TrueSense Kit hardware to change settings and obtain data gathered by the kit.

## Abbreviations

Unified Controller (controller)	C or UC
TrueSense (sensor)	S or TS
Memory Module (memory)	M or MM

## System Overview

The TrueSense Kit is composed of a unified controller (C or UC), truesense sensor (S or TS), and memory module (M or MM). These abbreviations are used throughout the kit.

The UC connects to a computer through a USB port using a Communications Device Class (CDC) resulting in a virtual com port appearing on the computer. Even though CDC is a standard USB class, it still requires a driver on Windows platforms. This driver is provided by Atmel which has the benefits of being digitally signed for Windows. It should appear as “Communications Device Class ASF example” in the Device Manager. The UC is the slave and the computer is the master. The UC and computer exchange packets that are encased in the OPI Link protocol. The protocol describing the encased commands are defined in the OPI Wired Frame Definition. The UC is a passive device in that it only reacts to commands given by the master. For every packet received by the UC, it should pass one packet back.

The UC can connect through a micro SD mechanically compatible, SPI electrically compatible port to either the MM or TS or both in tandem configuration. The UC is always the master in this SPI configuration and talks to both TS and MM, which are both slaves. The communication content between the UC, MM, and TS is not described here since they are indirectly accessed through commands given to the UC. Most of the commands passed over SPI are for retrieving data or changing settings. When the TS is physically plugged in to the UC, either directly or through the MM, it will not collect sensor data (consequently it will not write data to MM when plugged in). The TS will only generate sensor data when it is not physically connected to the UC.

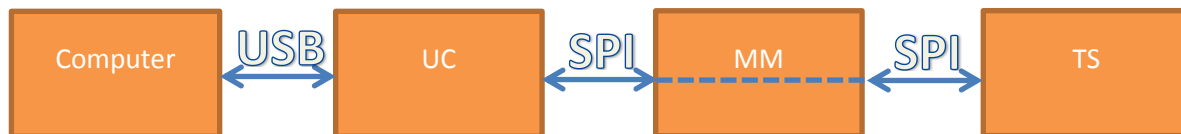


Figure 1: System Diagram with the different parts physically connected.

If the TS is not plugged into the UC, then it will send out wireless data if RF transmit is enabled. In this configuration, the UC can only receive TS data and cannot actively set any TS parameters. If a MM is plugged into the TS, the TS and MM communicate through an SPI interface where TS is the master and MM is the slave. If the memory module write setting is enabled on TS, it will write collected sensor data

to the MM. It can hold over 11 hours of data, which may be discontinuous. That is, if the MM is disconnected from TS and reconnected at a later time, then TS will still write data to MM, but it will be data that is collected at that point in time.

Note that both RF and MM write mode can be enabled simultaneously. They also can be both disabled which would result in no sensor data being recorded anywhere.

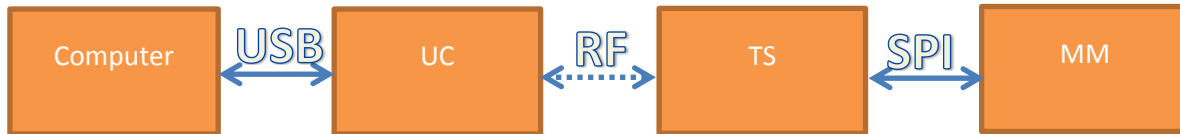


Figure 2: System Diagram with the different parts physically connected.

## Hardware Description

The UC is essentially a RF receiver with USB and SPI capability. The RF receiver receives zigbee like signals on software configurables channels. The UC has some limited data interpretation/correction functions, but mostly executes simple commands given by the computer. It has a real-time clock that is kept alive by the battery mounted on it. The counter frequency is 4096 Hz to allow it sub-millisecond resolution to determine RF packet reception. The battery is rechargeable and recharged when USB power is present. The UC is also an event/tag recorder with the two buttons located on it. These buttons, when pressed, create a time record of the button being pressed which can then be read out at a later time. This can be useful for marking events in the course of recording data. The UC also has 4 LEDs which show the state of things. The orange LED lights up when the UC is awake and has power. It will be off if the UC is disconnected from a power source or actively shutdown to preserve battery life. The green LED will toggle state whenever it receives some kind of RF data. Thus a constant state (always on or always off) indicates no RF signal is present. The UC has a mechanical microSD compatible socket. The pin mappings are different however than the SD standard. The user should only plug in an MM or TS. It does not work with uSD memory cards and in fact may damage them.

The MM is just an SPI Flash memory chip with 256 megabit capacity. It is used with TS to record high quality data when RF interference effects are not desirable. It can be read out and erased by the UC when it is plugged in. Based on the established data format, it can record over 11 hours of the data generated by TS.

The TS is an electrical signal, temperature, and 3-axis acceleration sensor. When unplugged from the UC it is powered by its own battery which can last a day of sending its measured data out over RF constantly. The battery is recharged when plugged into UC from UC's battery or from USB power. The electrical signal, designed for sensing small signals (input voltage range is ~400uV to 800uV, with resolution of better than 1 uV), is sampled at 512Hz. The temperature has about 1 degree C accuracy sampled at 8Hz, but resolution can be increased by oversampling. The 3-axis acceleration sensor has x and y directions sampled at 8Hz with a range of +/-2g and resolution of 8bits. The z direction is sampled at 32Hz with a range of +/-2g and resolution of 8bits as well. It weighs about 3g. The TS also has a real-time counter

which allows absolute time, if previously set, to be saved with data when writing the MM. The frequency of this real-time counter is 512Hz, the same as the ADC sample rate.

Both the TS and UC have shutdown modes in which battery drain is minimal and should be used when storing for extended periods of time.

## Hardware/Software Interface

Since the UC shows up as a virtual com port on the computer, one only has to access the com port in software. Functions to open the com port and pass commands are in the `opi_win.cpp` and `opi_win.h`. These source files are written in ANSI C, but use windows functions to access the com port. With these functions, included in ones C based project, one can access the plugged in TS and MM through the UC to retrieve data or change different settings. Please see the source code which has more detailed descriptions of the functions, what they require as arguments, and what they return.

For an example of interface, please see the `opiconsole` source code which is meant as a demonstration application. In the `opiconsole` source code, `ucRefresh` function, called every time some kind of action happens in the program, is illustrative of opening the correct com port automatically, then using it to set the time of the UC and TS, checking if there are any events/tags in the UC, and checking if a plugged in MM has any data or not. Other functions in `opiconsole` can be examined for examples on how to change TS settings, get data off the MM.

Additional commands exist to put the UC in ON and OFF mode which will dictate the TS mode when it is plugged in. In the ON mode, the UC has RF functionality on and will consume more power. In the OFF mode, the UC has RF functionality turned off and will consume less power. By implication, in the OFF mode, not RF data will be detected. For TS, in the ON mode, it will act as a sensor when pulled out of the UC socket. In the OFF mode, the TS will immediately shutdown when pulled out of the UC socket. These modes are introduced to maximize power savings and accidental battery drain.

Furthermore, SPI interface activation and deactivation commands are introduced. These are used so that the UC can work with future products. One product requires triggering which can be done with the microSD socket.

Additionally, a battery cycling command is introduced which will put the UC into a maximal current consumption mode without reacting to any commands so that the battery can be discharged and then later recharged to improve battery capacity issues that may have been introduced by lack of use of the UC.

## Documentation

Commands that sent between the computer and UC are defined by the OPI Wired Frame Protocol described in OPI Wired Frame Definition v1.00. These commands are encased by OPI Link as described in OPI Link Protocol v1.00. Data that is received by UC can be stored in a .opi format as described in OPI File Definition v1.00. This format stores only received data and can contain data from multiple sensors.

It is easily converted to industry standard formats such as .edf and .wav with the provided executable. The source code for the executable is provided as well so the user can examine how hardware interface is done and what kind of applications are possible. Qt was chosen as the platform to write the demonstration application due to its open source nature and cross-platform portability.