Tucson Amateur Packet Radio

TangerineSDR

Architecture and System Requirements

Preliminary Rev 0.3

30 May 2019

1. Introduction

The TangerineSDR is a modular FPGA-based Software Defined Radio. It is built on several plug-together PCB modules: a Data Engine (DE) baseboard, one to three plug-in RF Modules (RFM or RFDM) and a plug-in Clock Module (CKM). In addition, external components will be defined for additional functions such as low-pass/high-pass filters, power amplifiers, front panel controllers, power supplies, enclosures, etc.

The Data Engine (DE) baseboard (**Error! Reference source not found.**) contains an FPGA, support logic and four types of interfaces: Clock, RF, communications and low-speed I/O.

The FPGA is an Altera/Intel MAX10 with 50K LEs. It has a micro-SDXC card slot for local buffering of data. Local SDRAM is provided for soft-core CPU program storage, and QSPI flash memory is provided for non-volatile storage of FPGA programming images as well as other parameters. An optional battery-backed up RTC is provided.

The Clock interface consists of an M.2 connected 4-port differential clock interface with support for VCXO, OCXO and GPSDO clock sources. Input(s) and output(s) are provided for synchronization of multiple DEs to one clock source, either from one DE or from an external clock source.

The RF interfaces consist of two M.2 connected 500MByte/s LVDS parallel RFM interfaces and an MEC connected 500MByte/s RFM interface. These interfaces are used to connect receive ADCs, transmit DACs and other RF subsystems (e.g., Lime Microsystems or Analog Devices transceiver chipsets). The MEC RFM is compatible with SDRstick™ HF1, HF2 and TX2 boards. The M.2 interfaces are electrically identical and are used to connect custom TangerineSDR RF Modules (RFMs).

There are five communications interfaces: GbE, Superspeed (full duplex 5Gbps) USB 3.0 Device, Highspeed USB 2.0 Host, 802.11b/g/n Wi-Fi and Bluetooth 4.2. The GbE is implemented as a three-port switch, with one local port connected to the FPGA and the other two externally connected to RJ45 jacks. One external port is for connection to a Single Board Computer (SBC) for SDR Command and Control and the other is for connection to a local LAN or Internet gateway for direct streaming of data. GbE is intended to be the main high-speed data connectivity, with USB 3.0 as an auxiliary port. The USB 2.0 Host port is intended to support RTL-SDR dongles. Wi-Fi and Bluetooth are build options.

The low-speed I/O interface consists of a single through-hole connector option. It contains I2C, SPI, UART, I2S and parallel GPIO interfaces. A single RS-485 port is provided for an optional remote sensor. Four low-speed analog inputs and four analog outputs are also provided.

With appropriate RF and clock modules and FPGA programs, the TangerineSDR can be used for many applications.

Power is supplied to the DE from an off-board 11V - 15V power supply or by a 12V battery. The power input is rated at 19V maximum and 11V minimum. Power and clocks are supplied to the RF Module connectors via the DE baseboard.

1.1. Example Configurations

The two M.2 RFM connectors are electrically equivalent, and any RFM can be plugged into either connector. The intent of this modular approach is to support multiple widely disparate RF functions with one FPGA-based Data Engine. A few examples are listed below.

1.1.1. Receiver

The TangerineSDR can be configured as receive-only by using one of the two RF Module (RFM) connectors for an RF receiver (RX) module. The RX module can be a single-channel or a multi-channel receive module occupying one M.2 RFM connector. Two RX modules can be used simultaneously, one in each M.2 connector, if multiple synchronous receive channels are desired.

1.1.2. Transceiver

The TangerineSDR can be configured as a transceiver by using one RF Module connector for an RX module and the other for an RF transmitter (TX) module. Alternatively, since the two RF module connectors are arranged along one axis, a double-wide RF Module (RFDM) can be used with the DE board. This double-wide module will support high pin-count functions such as RF subsystem chipsets and many-channel receivers and transmitters.

1.2. Additional Documents

For Clock Module information, see the document **TangerineSDR Clock Module Specification** and for RF Module information, see the document **TangerineSDR RF Module Specifications**.

The Clock Module (CKM) and RF Module (RFM) interfaces are described in **TangerineSDR RF Module (RFM) Interface Specification** and **TangerineSDR Clock Module (CKM) Interface Specification.**

Ethernet protocol information is contained in the documents **TangerineSDR Local Command and Control (LCC) Protocol**, and **TangerineSDR Remote Command and Control (RCC) Protocol.**

1.3. I/O, Shields and Compatibility

The TangerineSDR DE is compatible with existing expansion boards: Raspberry Pi Hat/Micro Hat, Arduino Shield, mikroBUS^{TM} (Click^{TM}) board, Ultra96 Mezzanine and Digilent Pmod^{TM} modules. It is also able to act as a shield itself to Raspberry Pi 3, Odroid N2 and compatible Single Board Computers (SBCs). In order to meet all of these requirements, a single low-speed expansion connector will be used along with different adapter boards for desired connectivity.

1.4. System Architecture

Due to the limited capabilities of the SBC in most systems, a somewhat different system architecture is employed from a conventional SDR where the SBC is in the middle of the data path. The SBC will perform authentication and present a command and control interface to the network via its TCP/IP stack using Remote Command and Control (RCC) protocol. The SDR hardware is then controlled by the SBC using Local Command and Control (LCC) protocol. The LCC protocol is not exposed to the network outside the SBC-SDR connection for security reasons. The SBC can use LCC to set up direct high-speed UDP paths between the SDR hardware and any IP address for radio transmit of receive data. While the endpoints of these UDP streams may be outside the local network, the streams themselves can only be set up by the SBC under the LCC protocol. Since the SBC will be running an operating system (Linux assumed), RCC security will be implemented within the framework of that operating system. The system configuration is shown in **Figure** 1.

Note that if more processing power is required in the system, the SBC may be replaced by a more powerful computer. All that is required on the local computer is a GbE port. In some systems, the SBC may do some local processing, but still require greater processing power or more storage. UDP streams may be directed to any needed network resource, either on the local network segment or out on the Internet.

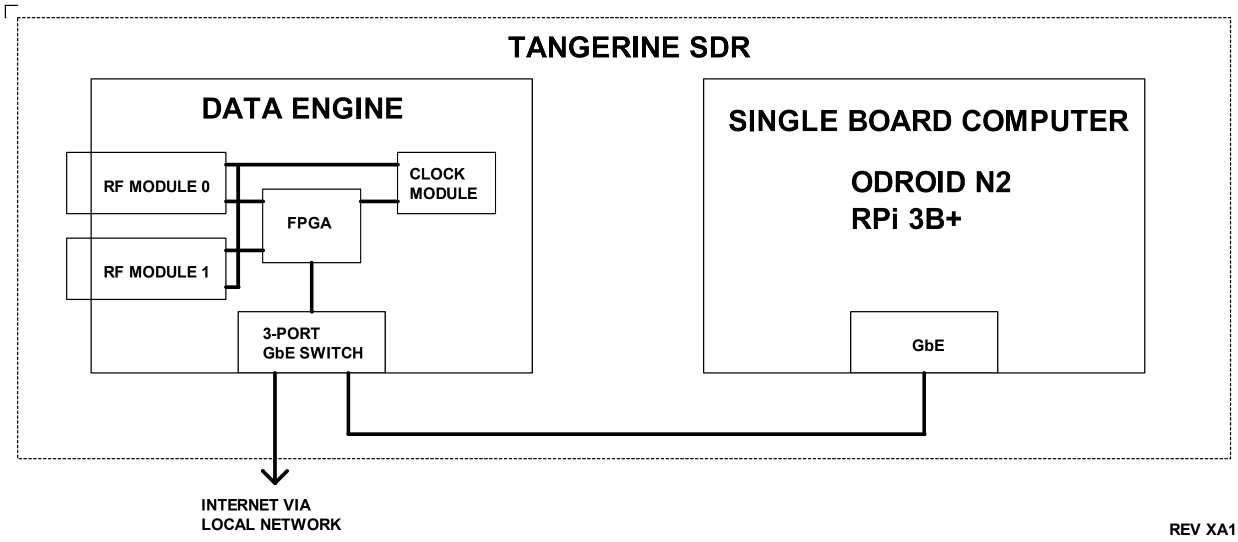

Figure 1 TangerineSDR Node Block Diagram

2. System Topologies

The programmable modular nature of the TangerineSDR system leads to many different network topologies. The LCC and RCC Ethernet protocols described in Section 3 must support all of these topologies. The basic building block for all Tangerine Systems is the TangerineSDR **Node**, shown in Figure 2.

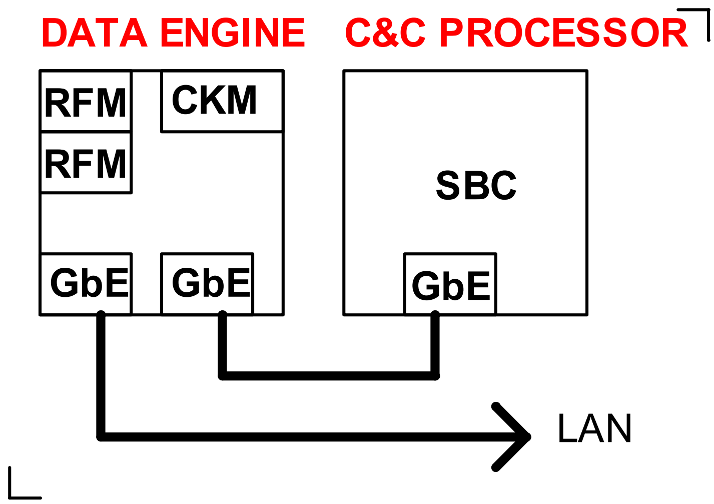

Figure 2 TangerineSDR Node

Even though the LAN port physically connects to the DE, all command and control of the node is handled by the SBC using TCP/IP. The two GbE ports on the DE, as well as the internal DE port to the FPGA (not shown) are connected as a 3-port GbE switch. The SBC controls the DE with UDP C&C protocol (LCC) over the same network segment. The SBC can command the DE to stream UDP data to either the SBC itself, or to IP addresses external to the node (i.e., out on the LAN). This UDP streaming is part of the LCC protocol.

2.1. Local Topologies

Each local configuration consists of one or more Nodes (DE/SBC pairs) acting as Server(s), with one or more computer(s) (SBCs, laptops or desktops) acting as Clients. In these scenarios, the Clients are in control of the system. Each Client can make requests of any Server in the system. Servers respond back directly to Client requests. Note that the protocol allows a Client to request a UDP stream to an IP address different than its own, but this is not necessarily supported in the local configurations.

Client-to-Client and Server-to-Server communications are not supported.

In all local configurations, communication is over the GbE port, and uses only the Local Command and Control (LCC) protocol. The local topology is used for small, self-contained, stand-alone systems, although in the Multi-Server Multi-Client configuration, the stand-alone system could become rather large.

2.1.1. Stand-alone SDR

The simplest local configuration is the stand-alone SDR (Figure 3). It consists of a TangerineSDR Node with an added display, keyboard/mouse and other radio specific I/O such as a speaker and microphone (not shown).

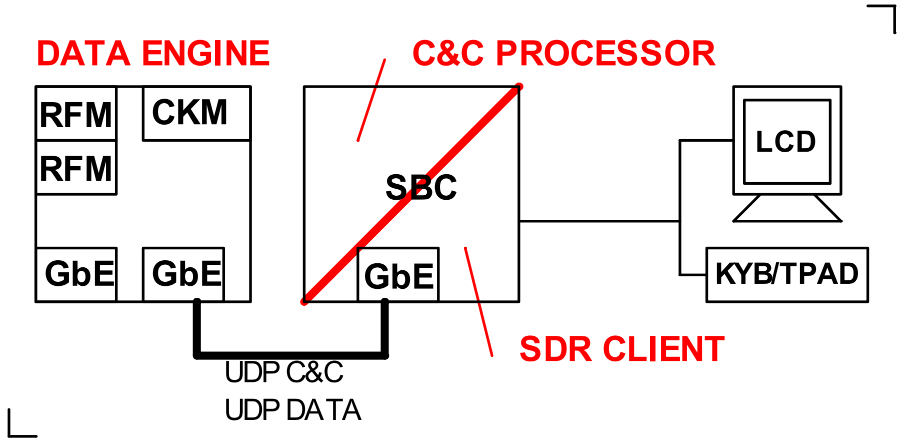

Figure 3 Stand-Alone TangerineSDR

In this case, the SBC runs both Command and Control software and the SDR client software, which may be a challenge for most low-end, inexpensive SBCs.

2.1.2. Single Server Local Systems

In Single Server local systems, one TangerineSDR node acts as a Server to one or more Clients, shown in Figure 4 and Figure 5. The number of Clients may be expanded up to the Gbe channel capacity or the DE's ability to process UDP streams, whichever is exceeded first. In order to service more than one Client, an external GbE switch must be used to provide connectivity.

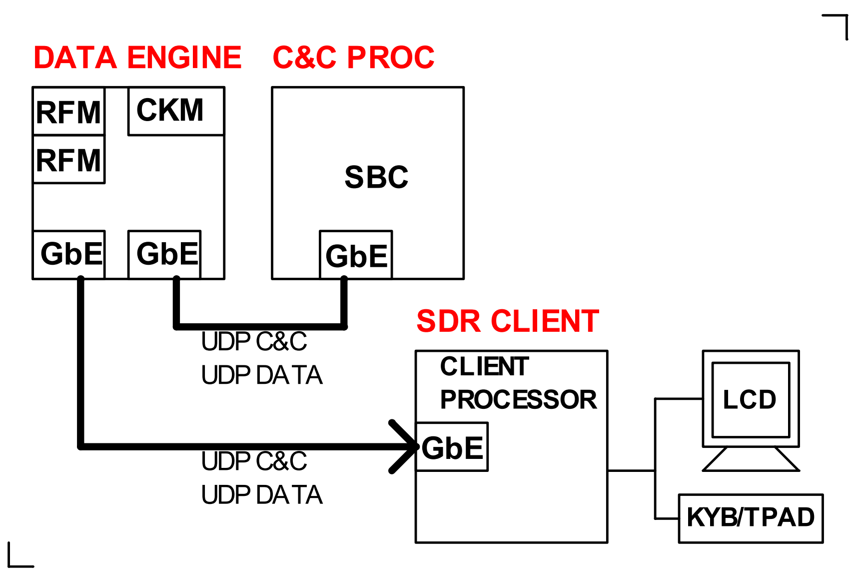

Figure 4 Single-Server Single-Client TangerineSDR

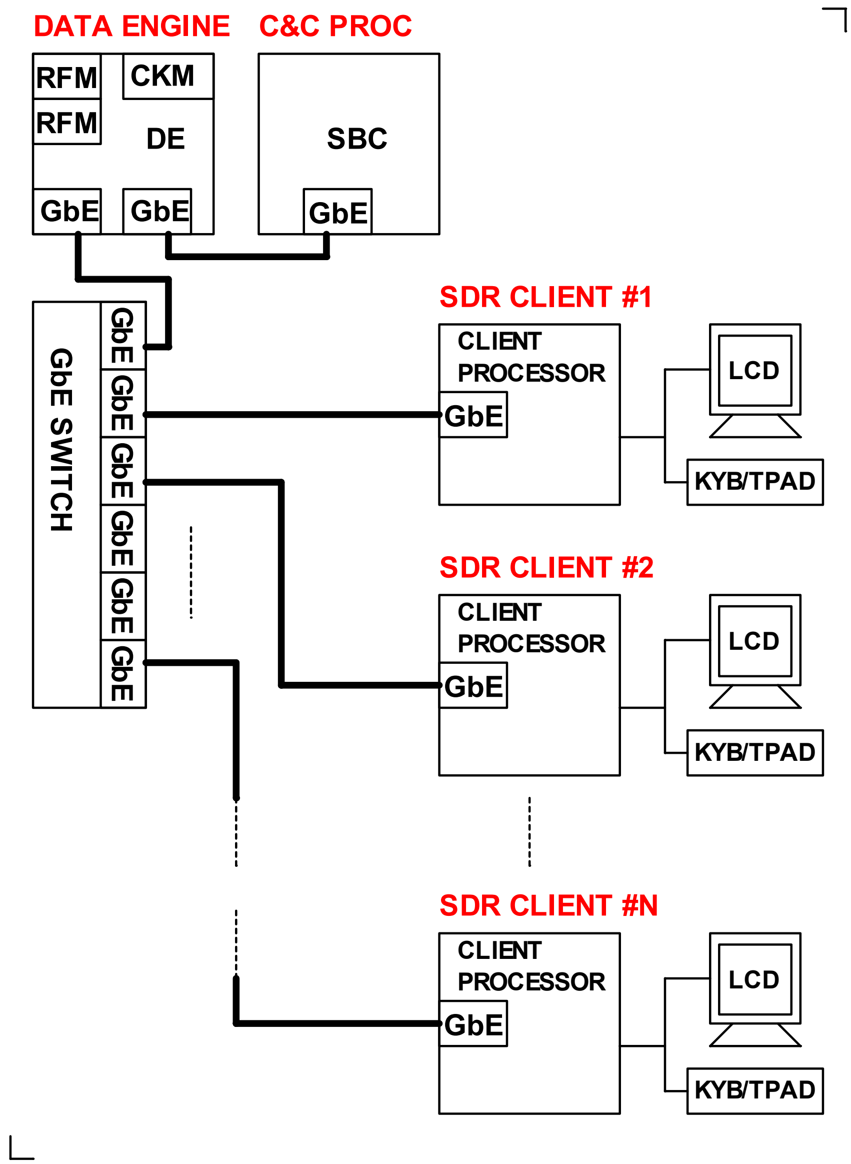

Figure 5 Single-Server Multi-Client TangerineSDR

2.1.3. Multi-Server Local Systems

In Multi-Server local systems, multiple TangerineSDR Nodes act as an array of Servers to one or more Clients, shown in Figure 6. The number of Servers and Clients may be expanded up to the Gbe channel capacity. An external GbE switch is used to provide connectivity. The individual DEs that make up the Server array may be synchronously clocked, since DE hardware provides an external synchronization input.

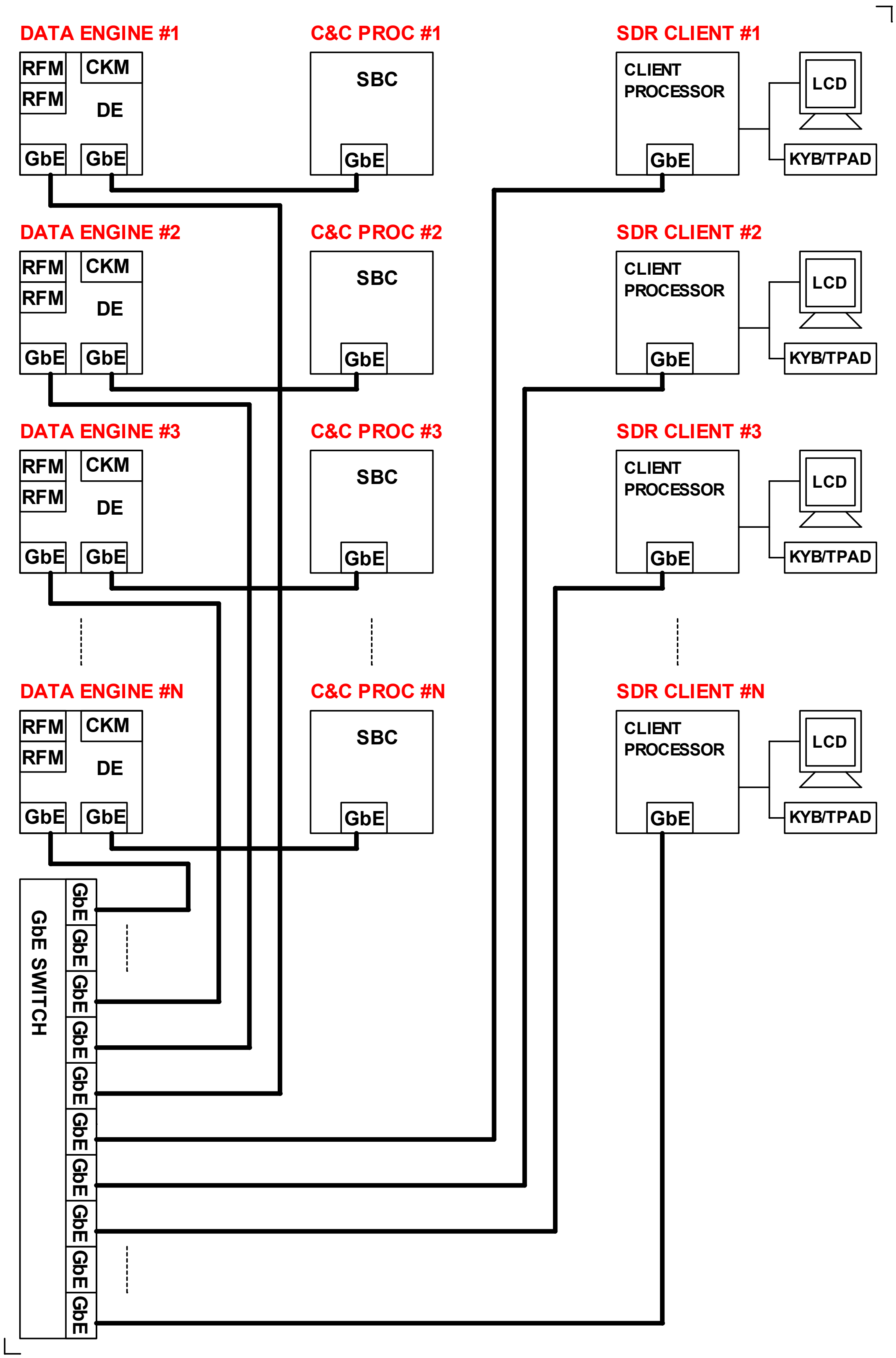

Figure 6 Multi-Server Multi-Client TangerineSDR

2.2. Remote Node Topologies

While technically not limited to a single Node, it is the only remote topology supported at this time. Note that a “Node” in this context can act either as a Server or Client to the Central Server, depending on how the networking software is designed. The Remote Node could be expanded into a more capable “Server” by adding additional processing resources to its local network. In effect we would be replacing the C&C Processor Single Board Computer with a more capable computer, such as a powerful desktop PC or even multiple PCs.

We will assume that the Remote Node will act as a Server and the “Central Server” will be acting as a Client for the purposes of data collection, since it does not affect the hardware requirements of the Remote Node.

Consider the Central Server a data storage Client to an array of remote “Server” nodes that provide data to the Central Server on request. During data distribution, the Central Server will likely be a true data “Server” to many scientific user “Clients”.

While the Remote Node physical connections between the DE and SBC are identical, there are two different methods of feeding data to the Remote Client. The most flexible method, called “SBC Feed”, is to have the DE stream UDP data to the local SBC under control of that SBC. The SBC processes and formats this data and sends it on to the Central Server. This works well until the data rate exceeds the capacity of the SBC to process it. Since the FPGA is much faster than the SBC at certain tasks (like processing and formatting), we can use “DE Feed” to stream the data directly from the DE to the Central Server. DE Feed must use UDP transport, while SBC Feed can use either TCP/IP or UDP transport to send the data to the Central Server.

2.2.1. SBC Data Feed

SBC Data Feed is diagrammed in Figure 7. I/Q data flows from the antennas to the FPGA where it is processed and sent on to the SBC via UDP. It is further processed and formatted by the SBC and is then sent on to the Central Server by either UDP or TCP/IP. This places the SBC in the middle of the data path, limiting throughput to the SBCs capability, but allowing very flexible processing by an SBC software application.

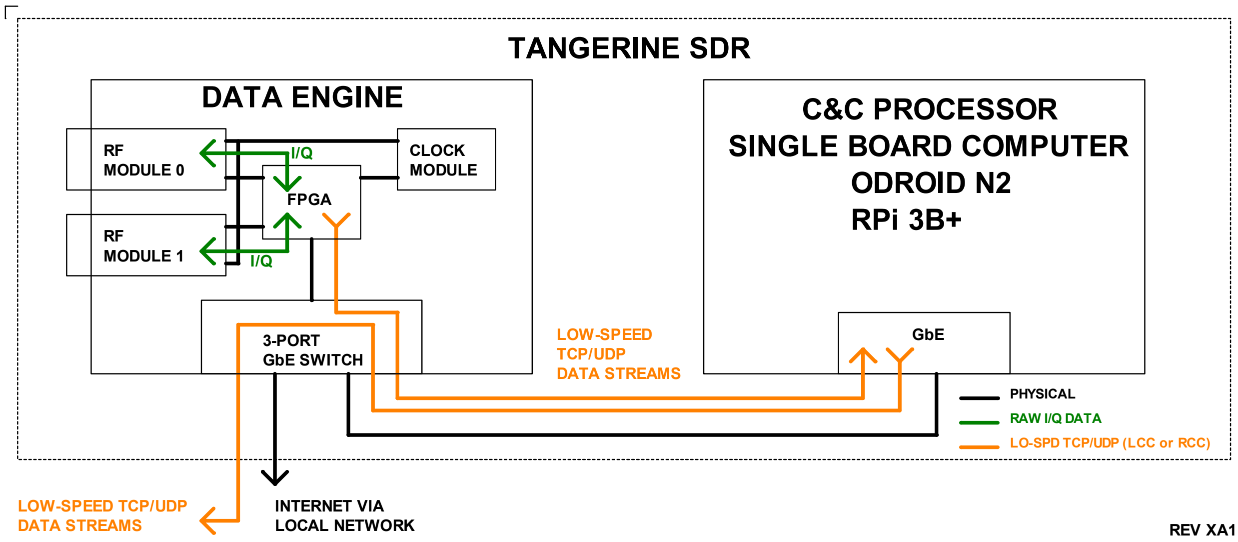

Figure 7 SBC Data Feed

2.2.2. DE Data Feed

DE Data Feed is diagrammed in

Figure 8. I/Q data flows from the antennas to the FPGA where it is processed and sent directly to the Central Server via UDP. This removes the SBC from the middle of the data path, providing greater throughput, but requiring less flexible processing by the FPGA configuration.

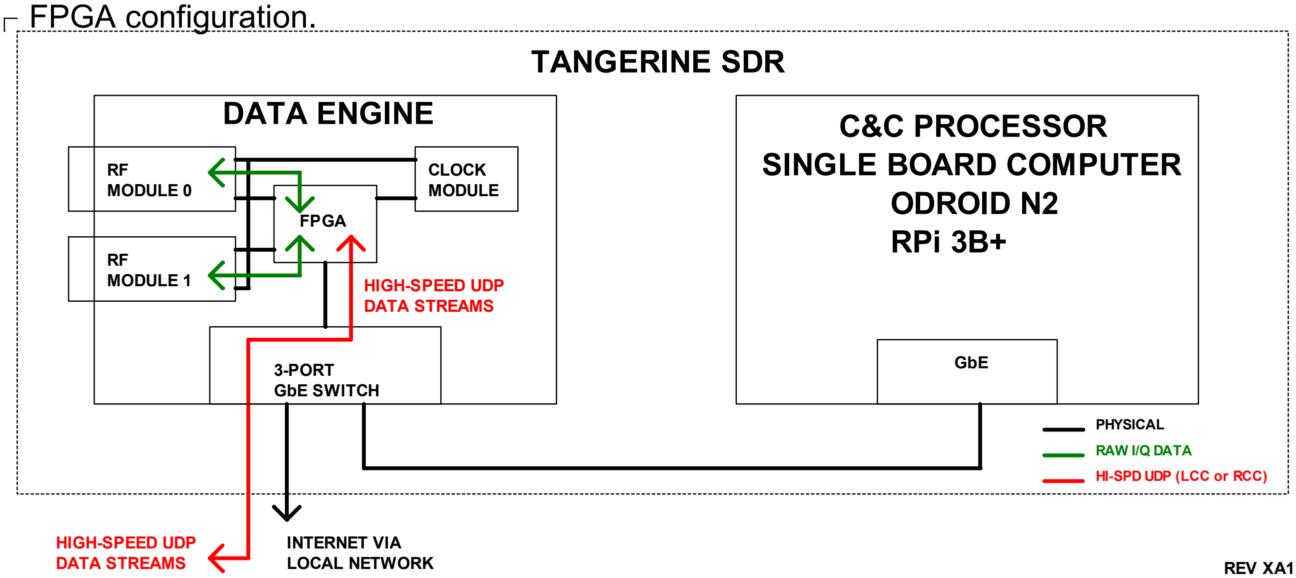

Figure 8 DE Data Feed

2.2.3. Command & Control (C&C)

Command & Control is diagrammed in Figure 9. TCP/IP commands flow from the Remote network through the GbE switch to the SBC where they are interpreted and sent to the DE via UDP. The DE sets up the data path either directly to a remote IP address and port (DE data Feed) or to the SBC for further processing (SBC data Feed).

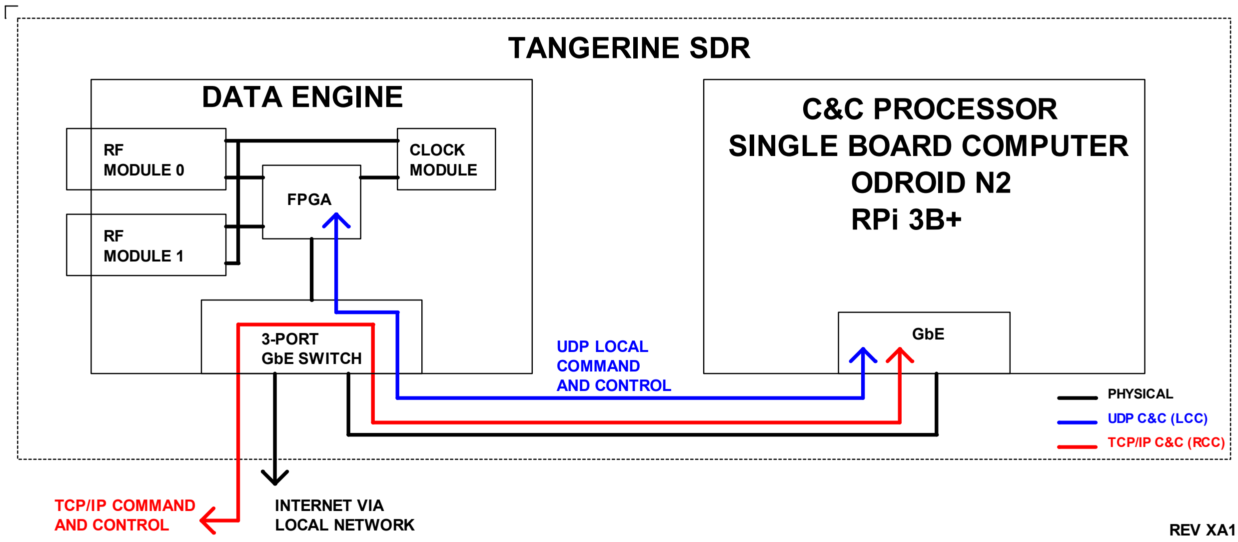

Figure 9 Command & Control

3. Ethernet Protocols

This section is not a protocol specification, but a place to list requirements that the protocol must satisfy as a minimum. The Local Command and Control protocol, or LCC, is UDP based and is used for communications between the DE and the SBC. These two pieces of hardware form a TangerineSDR “Node”. The DE does not implement TCP/IP or any kind of encryption or security features. It can be used for communication between multiple Servers and Clients, and packets are not intended to be routed out onto the Internet.

The Remote Command and Control protocol, or RCC, on the other hand, is implemented mostly using TCP/IP and is intended for communications between the SBC and the outside world, such as a Central Server. Authentication is required and encryption is possible. Data can be streamed via UDP using RCC, but all commands require TCP/IP connections.

All local topologies (see section 2.1) use LCC exclusively. Remote topologies use a mix of LCC and RCC.

3.1. Local Command and Control (LCC)

The Local Command and Control (LCC) is a UDP protocol that is used between the SBC and the DE within a TangerineSDR Node. The DE is not required to implement any TCP/IP communications. The LCC must be able to support multiple simultaneous Clients and Servers, and perform the following:

- Discovery broadcast (to Servers from Clients)
- Discovery Reply
- FPGA configuration flash erase
- FPGA configuration flash program
- Establish Client-to-Server command and status channels
- Establish various types of Client-to-Server data streams
- Establish various types of Server-to-Client data streams
- Channel management (time-outs, keep-alive, watchdog, etc)

3.1.1. Discovery

Discovery is a process by which each Client can identify every Server on its subnet. Clients broadcast a Discovery Request, and every Server within the subnet responds directly to each Client with a Discovery Reply.

3.1.2. Configuration Flash Erase and Program

Each Client may erase and re-program the FPGA image of any Server located on its subnet.

3.1.3. Establishing UDP Streams

Each Client may request the establishment of a UDP stream between itself and any Server on its subnet. In addition, each Client may request the establishment of a UDP stream from and Server to any other address on its subnet. UDP streams are defined when they are established by defining parameters such as type (I/Q data, raw RF data, audio data, etc) sample rate (48ksps, 96ksps, etc), data width (8b, 16b, 32b) etc. UDP stream can also be established by Clients for command and status exchange between Clients and Servers.

3.2. Remote Command and Control (RCC)

Even though there is only one physical configuration of Remote Node, it can be implemented in two different ways, as explained in sections 2.2.1 and 2.2.2. The data and command paths for the DE Feed implementation are shown in

Figure 10.

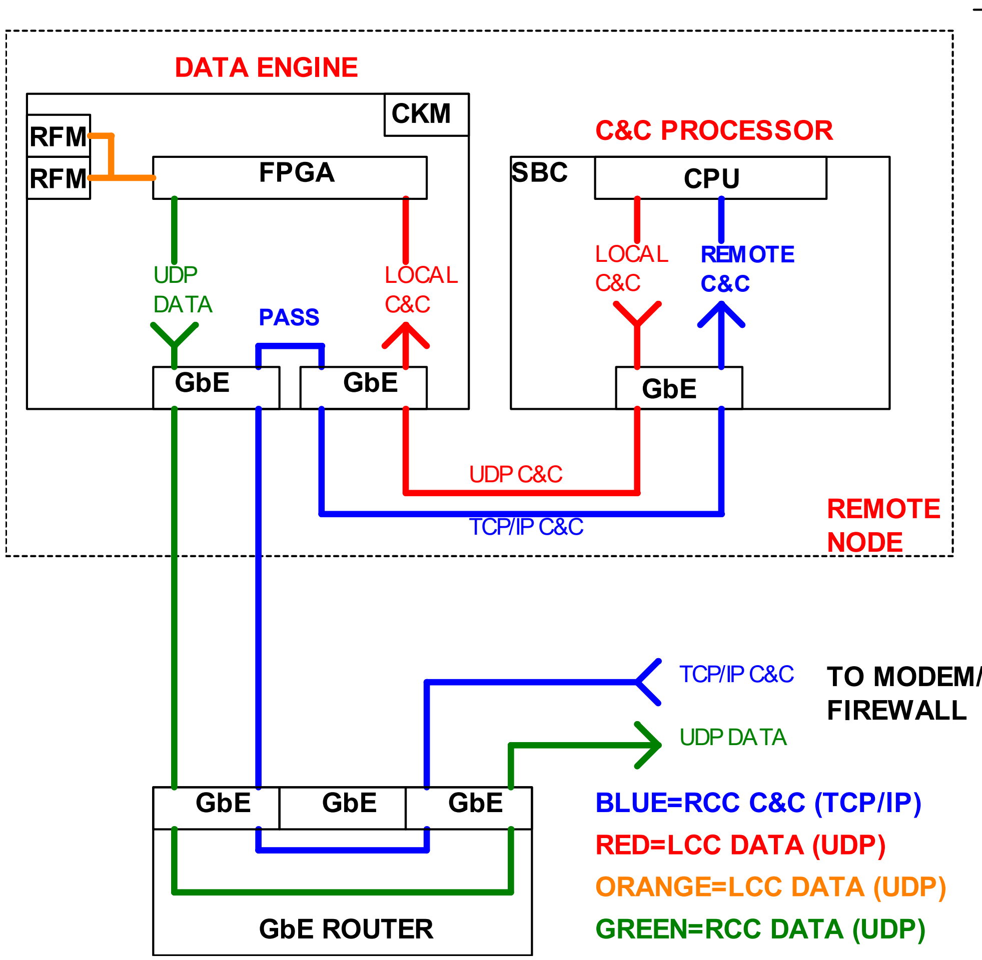

Figure 10 DE Data Feed Remote Node Command and Data Flow

The data and command paths for the SBC Feed implementation are shown in Figure 11.

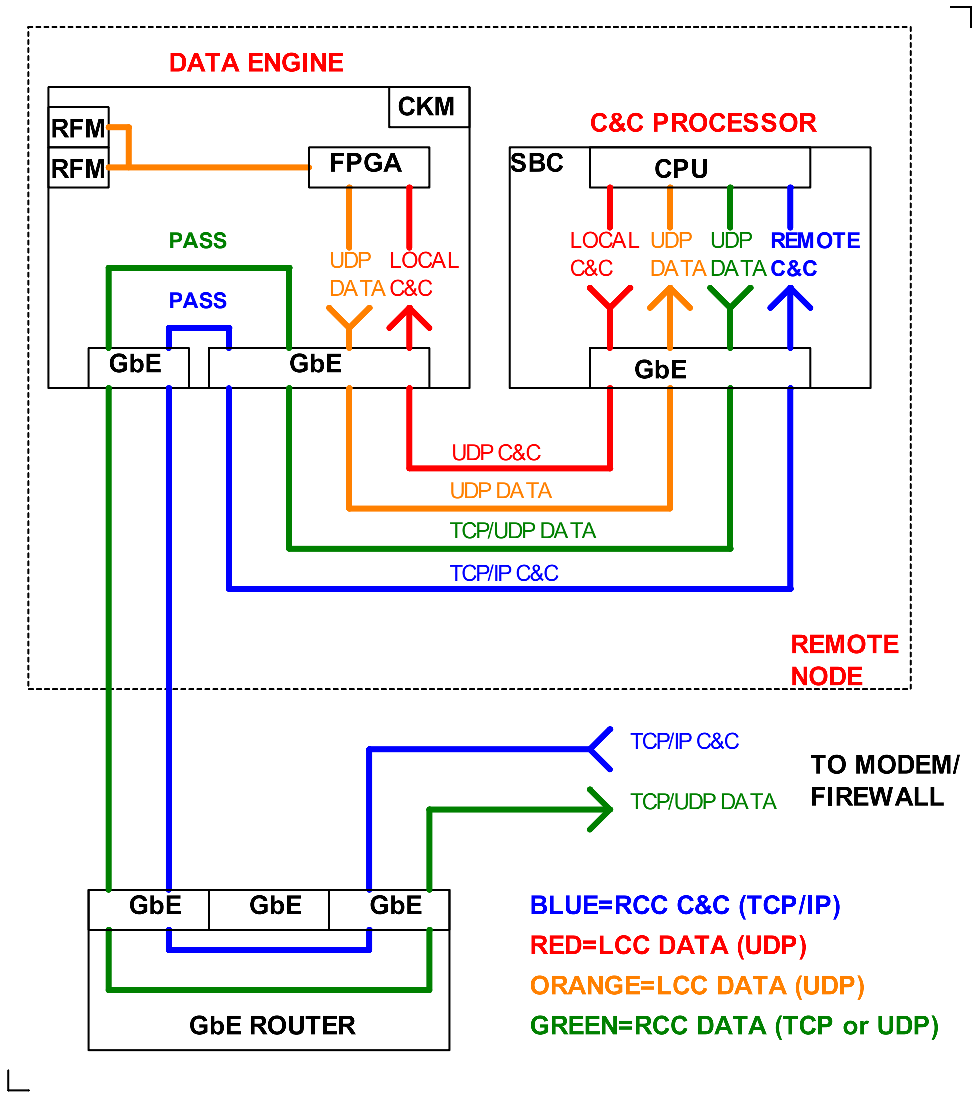

Figure 11 SBC Data Feed Remote Node Command and Data Flow

4. TangerineSDR Target Applications

4.1. HamSCI Personal Space Weather Station (PSWS)

The Personal Space Weather Station (PSWS) receiver RF Module has a receive range of 100kHz through 60MHz. Along with external sensors connected to either the DE or its attached SBC, it is useful for collecting ionospheric sounding and other solar data. Two channels are synchronously sampled at 122.88Msps by a dual 14-bit ADC. The PSWS RX Module has an on-board noise source that provides a known signal level used to calibrate the receiver. A 31-step, 1dB resolution attenuator controlled by the DE board provides AGC hardware to help prevent RF overload.

4.1.1. PSWS Clock Module

4.1.2. PSWS RF Module

4.1.3. PSWS DE

4.1.4. PSWS Additional Requirements

4.2. Phase IV Satellite Ground Station (P4G)

The RF Modules for Phase 4 Ground station support are ****TBD****. The P4G station consists of a 10GHz receiver and a 5GHz transmitter. The receiver will likely be a 900MHz LNB at the antenna, followed by a 900MHz SAW filter ahead of an under-sampled receiver operating in the 15^{th} (860 – 920MHz) or 16^{th} (920 – 980MHz) Nyquist band.

The transmitter will likely be a baseband design, with up-conversion and power amplification at the feed horn of the antenna.

4.2.1. P4G Clock Module

4.2.2. P4G RF Module

4.2.3. P4G DE

4.2.4. P4G Additional Requirements

4.3. Society of Amateur Radio Astronomers Receiver (SARARX)

4.3.1. SARARX Clock Module

4.3.2. SARARX RF Module

4.3.3. SARARX DE

4.3.4. SARARX Additional Requirements

4.4. EMI Software Defined Sniffer (SDSniffer)

4.4.1. SDSniffer Clock Module

4.4.2. SDSniffer RF Module

4.4.3. SDSniffer DE

4.4.4. SDSniffer Additional Requirements

4.5. STEM Research Software Defined Radio (STEM SDR)

4.5.1. STEM SDR Clock Module

4.5.2. STEM SDR RF Module

The RF Modules for experimenters' use are ****TBD****, although the PSWS receiver RF Module may be used for this purpose.

4.5.3. STEM SDR DE

4.5.4. STEM SDR Additional Requirements

4.6. Basic Amateur Radio HF Transceiver (BTRX)

4.6.1. BTRX Clock Module

4.6.2. BTRX RF Module

The RF Modules for general purpose use are ****TBD****, although the PSWS receiver RF Module may be used for this purpose.

4.6.3. BTRX DE

4.6.4. BTRX Additional Requirements

4.7. Advanced Amateur Radio Transceiver (ATRX)

4.7.1. ATRX Clock Module

4.7.2. ATRX RF Module

4.7.3. ATRX DE

4.7.4. ATRX Additional Requirements

5. TangerineSDR Components

The following cost goals are based on a build lot size of 1000 units and are actual build costs. Sale prices can be 30% to 50% higher or more, depending on retail sales arrangements.

5.1. Clock Modules (CKM)

5.1.1. CKM Descriptions

5.1.2. CKM Cost Goals

5.2. RF Modules (RFM)

5.2.1. RF Module Descriptions

5.2.2. RF Module Cost Goals

Cost target for the PSWS RX Module is $100, including BOM, PCB, assembly and test costs.

Costing for the P4G RX and TX Modules is ****TBD****.

Costing for future RF modules is obviously ****TBD****. The TangerineSDR DE will support very high data rates, especially if the RF Module is configured for supported LVDS data communication. Higher performance modules will cost more but will be fully supported by the same DE within the constraints of FPGA resources.

5.3. Data Engine

5.3.1. Data Engine Description

5.3.2. Data Engine Cost Goal

Cost goal for the TangerineSDR DE is $150, including BOM, PCB, assembly and test costs.