Matchstiq™ Z3u

RF Transceiver • Platform

HARDWARE USER MANUAL

V1.6 - OCTOBER 13, 2022

CHANGELOG

Revision	Date	Description	Author
0.0.1	2020-05-07	Pre-release draft, initial version	Meaghan Z
0.0.2	2020-09-03	Additional details added	Meaghan Z
0.0.3	2020-09-28	Additional details, format conversion	Barry L
1.0	2020-10-01	v1.0 release	Barry L
1.1	2021-03-16	v1.1 release	Barry L
1.2	2021-07-09	Added GPSDO and power consumption info, format cleanup.	Barry L
1.3	2022-01-10	Added USB host mode info.	Barry L
1.4	2022-03-03	Added info on how to change network settings	Barry L
1.5	2022-05-25	Updated microSD capacity (1TB), external reference clock to squarewave only, and component temperature rating -40C	Barry L
1.6	2022-10-13	Updated serial# info, Rev.B and C/D debug connector pin orientation, power consumption and thermal info, Z3u block diagram	Barry L

DISCLAIMER

Epiq Solutions is disclosing this document (“Documentation”) as a general guideline for development. Epiq Solutions expressly disclaims any liability arising out of your use of the Documentation. Epiq Solutions reserves the right, at its sole discretion, to change the Documentation without notice at any time. Epiq Solutions assumes no obligation to correct any errors contained in the Documentation, or to advise you of any corrections or updates. Epiq Solutions expressly disclaims any liability in connection with technical support or assistance that may be provided to you in connection with the Information. THE DOCUMENTATION IS DISCLOSED TO YOU “AS IS” WITH NO WARRANTY OF ANY KIND. EPIQ SOLUTIONS MAKES NO OTHER WARRANTIES, WHETHER EXPRESSED, IMPLIED, OR STATUTORY, REGARDING THE DOCUMENTATION, INCLUDING ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT WILL EPIQ SOLUTIONS BE LIABLE FOR ANY CONSEQUENTIAL, INDIRECT, EXEMPLARY, SPECIAL, OR INCIDENTAL DAMAGES, INCLUDING ANY LOSS OF DATA OR LOST PROFITS, ARISING FROM YOUR USE OF THE DOCUMENTATION.

TABLE OF CONTENTS

INTRODUCTION

This document provides an overview of Epiq Solutions' Matchstiq Z3u software defined radio (SDR) [1], a highly integrated wideband RF transceiver plus Linux computer. The following topics will be discussed:

- Overview of the Matchstiq Z3u hardware and available interfaces
- Matchstiq Z3u usage/integration options
- Matchstiq Z3u Platform Development Kit usage

LEGAL CONSIDERATIONS

Matchstiq Z3u is distributed all over the world. Each country has its own laws governing reception and transmission of radio frequencies. The user of Matchstiq Z3u and associated software is solely responsible for insuring that it is used in a manner consistent with the laws of the jurisdiction in which it is used. Many countries, including the United States, prohibit the transmission and reception of certain frequency bands, or receiving certain transmissions without proper authorization. Again, the user is solely responsible for the user's own actions.

PROPER CARE AND HANDLING

Each Matchstiq Z3u unit is fully tested by Epiq Solutions before shipment, and is guaranteed functional at the time it is received by the customer, and ONLY AT THAT TIME. Improper use can cause it to become non-functional. In particular, a list of actions that may cause damage to the hardware include the following:

- Handling the unit without proper static precautions (ESD protection) when the housing is removed or opened up
- Improper shutdown of the Matchstiq Z3u
- Connecting a transmitter to the RX port without proper attenuation
- Executing custom software and/or an FPGA bitstream that was not developed according to guidelines

The above list is not comprehensive, and experience with the appropriate measures for handling electronic devices is required.

REFERENCES

1. Matchstiq Z3u Product Page

   https://epiqsolutions.com/rf-transceiver/matchstiq-z/

2. Epiq Solutions Support Portal

   https://support.epiqsolutions.com

3. Sysfs Interface for Userspace

   https://www.kernel.org/doc/Documentation/gpio/sysfs.txt

4. Zynq UltraScale+ Device Technical Reference Manual

   https://www.xilinx.com/support/documentation/user_guides/ug1085-zynq-ultrascale-trm.pdf

5. Petalinux Tools for Embedded Systems

   https://www.xilinx.com/support/documentation-navigation/design-hubs/dh0016-petalinux-tools-hub.html

6. Xilinx's Wiki "How to Format SD card for SD boot"

   https://xilinx-wiki.atlassian.net/wiki/spaces/A/pages/18842385/How+to+format+SD+card+for+SD+boot

TERMS AND DEFINITIONS

Term	Definition
A/D	Analog to Digital converter
BSP	Board Support Package
D/A	Digital to Analog converter
dB	Decibel
dBm	Decibels referenced to one milliwatt (mW)
dpkg	Debian Package
eMMC	Embedded Multi-Media Controller
ESD	ElectroStatic Discharge
FPGA	Field Programmable Gate Array
GALILEO/BEIDOU	European Union's Galileo and China's BeiDou Navigation Satellite Systems
GHz	gigahertz
GLONASS	GLObal NAvigation Satellite System
GPIO	General Purpose Input / Output
GPS	Global Positioning System
GPSDO	Global Positioning System Disciplined Oscillator
IF	Intermediate Frequency
IMU	Inertial Measurement Unit
I/Q	In-Phase / Quadrature Phase
JTAG	Joint Test Action Group
ksps	kilo samples per second
LED	Light Emitting Diode
MHz	megahertz
MMCX	Micro-Miniature CoXial RF connector
Msps	Mega samples per second
OTG	On The Go, a variant of the USB protocol supporting both host and device operation
PDK	Platform Development Kit
PL	Programmable Logic
PPM	Parts Per Million

PPS	Pulse Per Second
PS	Processing System
QSPI	Quad serial peripheral interface
QZSS	Quasi-Zenith Satellite System
RAM	Random Access Memory
RF	Radio Frequency
Rx	Receive
SBAS	Satellite-based Augmentation System
SDK	Software Development Kit
SDR	Software Defined Radio
SMA	SubMinature version A RF connector
SSH	Secure SHell
TCVCXO	Temperature Compensated Voltage Controlled Crystal Oscillator
Tx	Transmit
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus

Table 1: *Terms and Definitions*

SYSTEM OVERVIEW

Matchstiq Z3u is a small form factor software defined radio system that provides a wideband RF transceiver plus a Linux computer. With the integrated Linux computer, Matchstiq Z3u is targeted for applications where the necessary signal/protocol processing tasks can execute standalone, without the necessity of a host computer platform. Alternately, Matchstiq Z3u can be interfaced with a host system (PC running Linux / Windows / MacOS or Android device), where supplemental processing and control of the Matchstiq Z3u could take place on the host system. Whether running in a standalone configuration or with a host device, all configuration and streaming of the Matchstiq Z3u is completed within the libsidekiq software API.

A high level summary of Matchstiq Z3u features is shown below:

- Leverages Analog Devices' AD9361 to provide a RF transceiver covering 70 MHz to 6 GHz, with independent Tx and Rx frequencies
- Dual Rx mode supported
- Integrated Rx pre-select filters
- Configurable A/D and D/A sample rates up to 61.44 Msamples/sec
- Integrated Linux computer provided via Xilinx Zynq UltraScale+ XCZU3EG System-on-Chip
- 2 GB of LPDDR4 RAM
- Integrated GPS receiver with PPS for high accuracy
- On-board 40 MHz TCVCXO with +/- 0.1PPM accuracy; support for optional external 10 MHz or 40 MHz reference clock input, software controlled
- 128 MB of QSPI flash storage for uboot bootloader and Linux kernel
- 128 GB eMMC for root filesystem storage
- microSD slot for additional storage or alternative boot options
- USB 3.0 network interface
- USB 2.0 OTG interface
- USB Micro-B serial console
- Zynq UltraScale+ PS/PL GPIO available
- Size: 2.64" x 3.63" x 0.81" (67.1mm x 92.3mm x 20.1mm) including SMA connectors
- Weight: 5.6 oz. (6.7 oz. with the optional mounting plate)
- Typical power consumption: under 6 W

A block diagram of Matchstiq Z3u is shown below:

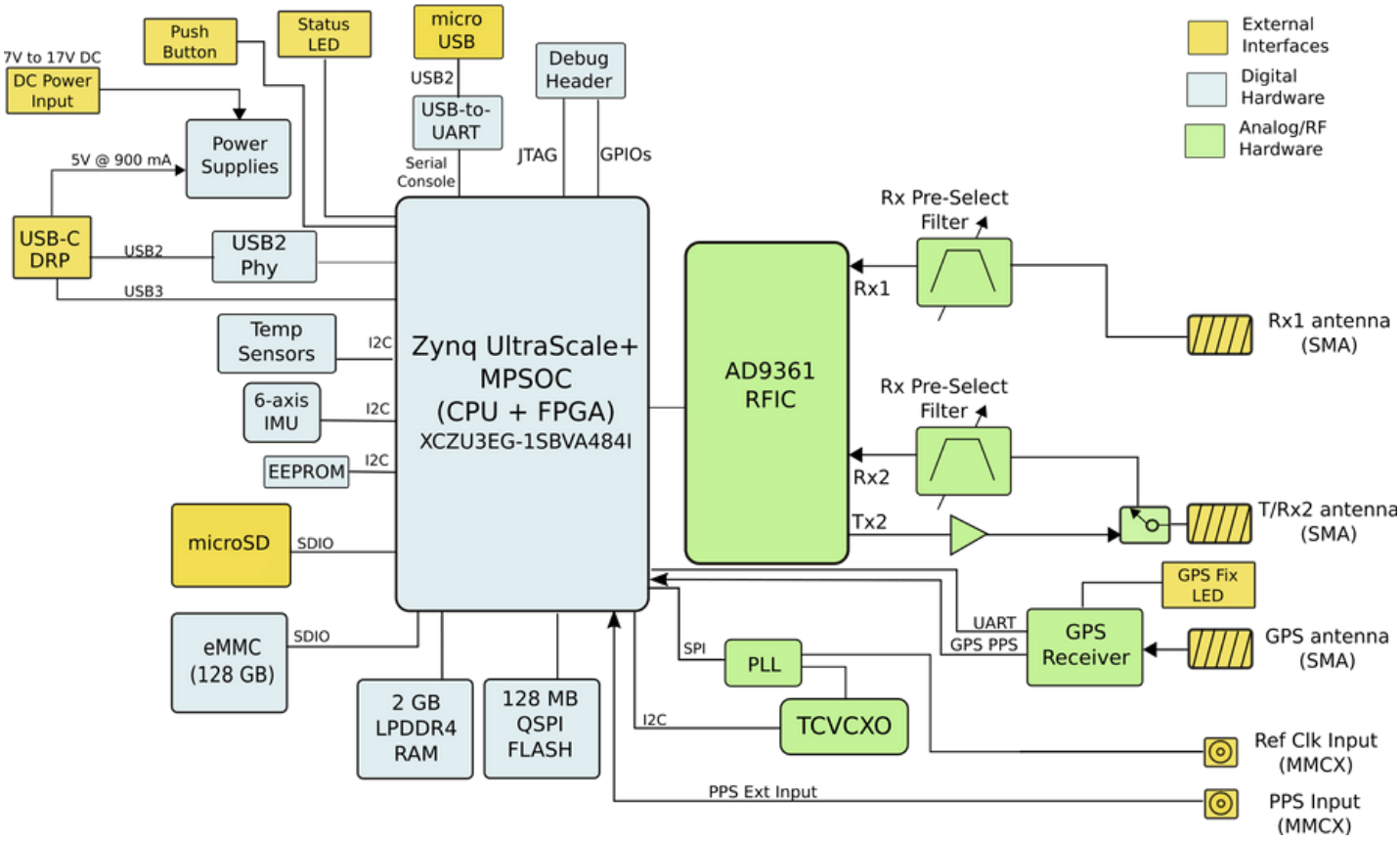

Figure 1: *Block diagram of the Matchstiq Z3u*

HARDWARE SPECIFICATION

RF RECEIVER SPECIFICATION

RF Inputs	SMA connector (50 ohms); total of two SMA connectors that can be used to switch between three external antennas (can be used as dual RX or dual-use Tx or Rx port);
Architecture	Zero-IF (direct conversion)
Tuning Range	70 MHz to 6 GHz
Rx Pre-Select Filter Bank	Sub-octave pre-select filtering for interference protection from ~50 MHz to 6 GHz; Automatically selected when tuning the RF receiver.
Tuning Step Size	~2.4 Hz
Tuning Time	~1 ms
Typical Noise Figure	8 dB
Typical IIP3	-18 dBm
Max RF Input Power Level (without damage)	+20 dBm
Gain Control Range	0 to 76 dB, 1 dB steps
Gain Control Mode	Manual or Automatic
A/D Converter Sample Rate	233 Ksps to 61.44 Msps
A/D Converter Sample Width	12 bits
Typical I/Q balance	> 50 dB

Table 2: *RF Receiver Spec*

RF TRANSMITTER SPECIFICATION

RF Input	SMA connector (50 ohms); dual-use connector supports operation as Tx or Rx port
Architecture	Zero-IF (direct conversion)
Tuning Range	70 MHz to 6 GHz
Tuning Step Size	~2.4 Hz
Tuning Time	~1 ms
Gain Control Range	0 to 89.75 dB, 0.25 dB steps
RF Output Power (P1dB)	+13 dBm < 2 GHz, +10 dBm above 2 GHz
D/A Converter Sample Rate	233 ksps to 61.44 Msps
D/A Converter Sample Width	12 bits
Typical I/Q balance	> 60 dB

Table 3: *RF Transmitter Spec*

CLOCK / SYNCHRONIZATION SPECIFICATION

RF Input Port	MMCX
On Board Reference Clock	40 MHz, +/- 0.1 PPM accuracy P/N: SiTime SIT5356
External Reference Clock Input Frequency	10 MHz or 40 MHz
External Reference Clock Input Power Range	0.8 – 3.0 Vpp, +7 dBm max from 50 ohm source for 10 MHz 0.8 – 1.3 Vpp, 0 dBm max from 50 ohm source for 40 MHz
PPS Input Level	3.3V max
GPSDO	Refer to GPSDO Performance section

Table 4: *Clock Spec*

DIGITAL SPECIFICATION

System on Chip	Xilinx Zynq UltraScale+ XCZU3EG Programmable Logic (PL) Specification:-154K Logic Cells-7.6 Mbits BlockRAM-360 DSP slicesProcessor System (PS) Specification:-Quad-core ARM Cortex A9 CPU running up to 1.5 GHz-Linux kernel 4.14, Ubuntu 18.04
RAM	2GB LPDDR4 RAM
Flash	128 MB QSPI
eMMC	128 GB
User Accessible microSD Slot	up to 1 TB
User I/O	USB 3.0 superspeed via USB-C connection, Linux serial console via USB Micro-B connection, Status LED, GPS Fix LED, and Push Button Switch
Debug I/O	RevB GPIO: Two PL (FPGA controlled) general purpose input/output pins, One PS (CPU controlled) general purpose input pins (can also be used to control boot mode), JTAG interface RevC GPIO: Four PL (FPGA controlled) general purpose input/output pins, Three PS (CPU controlled) general purpose input pins (can also be used to control boot mode), JTAG interface
GPS / GPSDO	OriginGPS ORG4033-MK05 NMEA sentences, PPS, and frequency-disciplining *GPSDO feature available in libsidekiq v4.16.1 / FPGA v3.14.1 release or later*
Temperature Sensor	Texas Instruments TMP103AYFFR Accuracy: -40 ºC to +125 ºC (+/- 1 ºC typ) Resolution: 1 ºC
Inertial Measurement Unit (IMU) Sensor	TDK / InvenSense ICM-20602 6-axis MotionTracking Device (3-axis gyroscope, 3-axis accelerometer) - Gyroscope sensitivity error: ±1% - Gyroscope noise: ±4 mdps/√Hz - Accelerometer noise: 100 µg/√Hz
Component Temperature Rating	-40 ºC to + 85 ºC

Table 5: *Digital Spec*

HARDWARE INTERFACES

Matchstiq Z3u provides a variety of different hardware interfaces for use by an end user. Each of these hardware interfaces is shown and defined below.

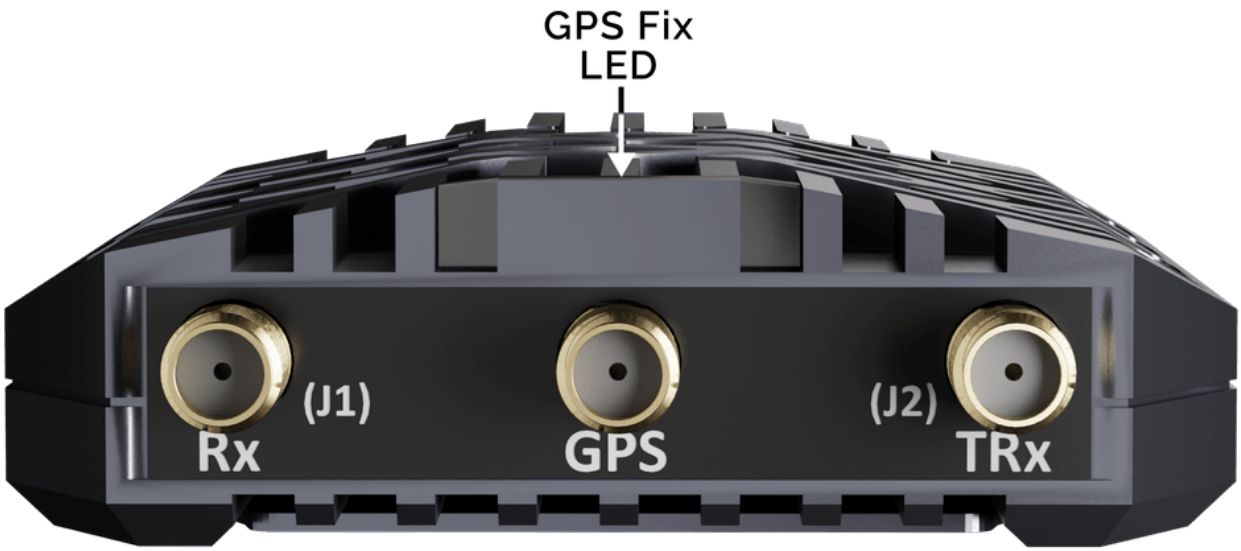

Figure 2: *Matchstiq Z3u front I/O ports*

RX (J1)

The Rx interface is a 50 ohm SMA jack connector that provides an RF input path that can be switched in to the RF receiver in the Matchstiq Z3u, allowing a user-specified antenna to interface to the RF receiver. This Rx port supports RF input frequencies between 50 MHz and 6 GHz. *The maximum safe RF input level to this port without damage is* ***+20 dBm***.

GPS

The GPS interface is a 50 ohm SMA jack connector that provides an antenna input path for the GPS receiver. Both passive and active GPS antennas can be used. *The maximum safe RF input level to this port without damage is* ***+10 dBm***.

TRX (J2)

The TRx interface is a SMA connector that provides an RF access path that can be used for either Rx or Tx. When operating in Dual Rx mode, this provides an RF access path to the second Rx path of the Matchstiq Z3u, such that Rx (J1) and TRx (J2) are both used as receivers tuned to the same LO. When operating in single channel TDD mode, TRx (J2) can be switched between receive and transmit. When operating in single channel FDD mode, TRx (J2) provides an RF access path to the transmitter. *The maximum safe RF input level to this port without damage is* ***+20 dBm***.

The selection of either transmit or receive mode for this RF port can be controlled via the libsidekiq software library.

Handle	RF Port [Single Channel Fixed-Mode]	RF Port [Dual Channel RX-Mode]	RF Port [Single Channel TRX-Mode]
skiq_rx_hdl_A1	skiq_rf_port_J1	skiq_rf_port_J1	skiq_rf_port_J2
skiq_rx_hdl_A2		skiq_rf_port_J2	
skiq_tx_hdl_A1	skiq_rf_port_J2		skiq_rf_port_J2

Table 6: *RF Port Mapping*

GPS FIX LED

The GPS Fix LED provides a visual status indicator that can be controlled via the FPGA. This LED is connected to a PL pin (IO_L3N_T0L_N5_AD15N_65) on the Zynq UltraScale+. By default, the LED is used to provide an indication if the GPS receiver has a fix.

Figure 3: *Matchstiq Z3u rear I/O ports*

PUSH BUTTON INPUT

The push button input is connected to the PS and is configured as a gpio-keys-polled device with the name of “ext_pb”. It is connected to a PS pin (PS_MIO70) on the Zynq UltraScale+. By default, the push button does not do anything, though monitoring `/dev/input/event0` for the key up event (103) will indicate when the push button event is detected. Example: `$ evtest /dev/input/event0`

STATUS LED

The Status LED #1 provides a visual status indicator that can be controlled through software. This LED is connected to a PS pin (PS_MIO_9) on the Zynq UltraScale+.

For Z3u Rev-B hardware, this LED is used to provide a heartbeat indicator after the system has booted Linux and is running, blinking at a rate of ~2 Hz. For Z3u Rev-C hardware, please refer to the table below.

Note, when the recovery image is booted, the Status LED blinks twice and then remains on for a longer period (inverted heartbeat).

LED Status	Description
Green	Power is applied
Green and blue	OS is booting up
Green and flashing blue (heartbeat)	OS boot complete, software is running

Table 7: *Z3u Rev-C Status LED*

EXTERNAL REFERENCE CLOCK INPUT (REF)

The External Reference Clock Input interface (software selectable) is a 50 Ohm MMCX jack connector that accepts an external 10 MHz or 40 MHz reference clock signal (software selectable) for the purpose of phase locking the on-board 40 MHz TCVCXO reference clock. This provides the facility to have multiple Matchstiqs share a common external 10 or 40 MHz reference clock. The use of this interface is optional and not required.

Coupling	AC
Frequency	10 or 40 MHz
Input level	0.8 – 1.3 Vpp, 0 dBm max. from 50 ohm source for 40 MHz 0.8 – 3.0 Vpp, +6 dBm max. from 50 ohm source for 10 MHz.

Waveform	squarewave

Table 8: *Electrical specification for external reference clock input*

PPS INPUT/OUTPUT (PPS)

The PPS interface is a 50 ohm MMCX jack connector that can be used to either provide a reference 1PPS input to the system or to output the 1PPS from the on board GPS receiver and is software selectable. The use of this interface is optional and not required. *A maximum recommended signal level of 3.3V can be applied to this input port.*

USB MICRO-B SERIAL CONSOLE

The USB Micro-B serial interface (CP2102 USB-UART) provides access to the Linux serial console. The default serial console runs at a rate of 115.2 kbps baud, with 8 data bits, 1 stop bit, no parity bits, and **no** flow control.

USB-C INTERFACE

The USB-C interface can be used to both power the Matchstiq Z3u and communicate via a network interface to the radio from a host device (such as a laptop or Android device). It supports a USB3.0 SuperSpeed interface.

MICROSD SLOT

The microSD slot can be used for removable storage or alternative boot device via a microSD card. Cards up to 1 TB are supported.

DC BARREL JACK INPUT (7-17VDC)

The DC power input barrel jack can be used as an alternative to the USB-C interface to power the Matchstiq Z3u.

The part number of the input barrel jack is **PJ-038-SMT** which is manufactured by CUI, Inc, with the center pin carrying DC voltage and the ring serving as ground. The mating connector is part number **PP-012**, also manufactured by CUI, Inc. *The acceptable DC voltage input range is between 7 and 17 VDC.*

TEMPERATURE & INERTIAL MEASUREMENT UNIT (IMU) SENSORS

The Matchstiq Z3u is equipped with a temperature sensor for monitoring on-board temperature and a IMU sensor for detecting orientation and tracking rotation or twist. The Z3u uses the following parts for monitoring:

- Texas Instruments Temperature Sensor TMP103CYFFR
- TDK / InvenSense High Performance 6-Axis MEMS MotionTracking™ Device ICM-20602

The libsidekiq software API provides access to these peripherals and test applications such as *read_temp* (included with the libsidekiq software bundle) demonstrate how to read the sensor's value. Additional information can be found in the Sidekiq Software Development manual.

IMU SENSOR (ICM-20602) AXIS ORIENTATION

+Z axis points up, +X axis direction is towards the rear (into page), and +Y direction is towards the left when the Matchstiq Z3u is oriented as pictured in Figure 2.

GPS

The Matchstiq Z3u includes an on-board GPS / GNSS receiver module to provide GPS timing as well as positioning and navigational data. The GPSDO requires an external GPS antenna to be connected to the SMA GPS antenna connector. The GPS can also be used to lock the internal oscillator to GPS timing/frequency.

The Matchstiq Z3u also provides a MMCX connector with the GPS receiver's pulse-per-second (PPS) signal for synchronizing additional radio modules.

The GPS receiver module is a miniature multi-channel GPS, GLONASS, GALILEO/BEIDOU, SBAS, and QZSS overlay systems receiver that continuously tracks all satellites in view, providing real-time positioning data in industry's standard NMEA format.

REFERENCE CLOCK OPTIONS

Matchstiq Z3u supports options to use either an internal (i.e. on-board) 40 MHz TCVCXO as a reference clock, or an external 10 or 40 MHz reference clock.

The internal reference clock is a SiTime SiT5356 high stability MEMS oscillator. This oscillator has 0.1ppm frequency stability over temperature with a +/- 6ppm tuning range. The GPS receiver module can also be used to lock the internal oscillator to GPS timing/frequency.

An external 10 MHz or 40 MHz reference clock (software selectable) can also be used for the purpose of phase locking the on-board 40 MHz TCVCXO reference clock.

Regardless of which clock source is selected, this reference clock will serve as the reference for both the RF front end as well as the digital processing blocks in the FPGA. The selection of whether Matchstiq Z3u uses the internal reference clock or the external reference clock is stored as a configuration parameter in EEPROM. This parameter is read at power up, and the clock source determination is then made.

If the Matchstiq Z3u is configured to use an external 40 MHz reference clock, but no external reference clock is provided via the MMCX connector any application attempting to initialize the SDR will fail.

For cases where a customer would like to change the default reference clock setting (used at power-up), a software application is provided to update the EEPROM configuration settings.

Note that changing reference clock sources while running is currently not supported; the EEPROM configuration must be changed, and then the new reference clock source will be used at the next power up. Please contact Epiq Solutions for details [2].

GPS DISCIPLINED OSCILLATOR (GPSDO)

Matchstiq Z3u features hardware to support a GPS disciplined oscillator. The key components are the GPS receiver module with a 1 pulse per second (PPS) output and a voltage control oscillator. The GPS receiver module is a miniature multi-channel GPS, GLONASS, GALILEO/BEIDOU, SBAS, and QZSS overlay systems receiver that continuously tracks all satellites in view, providing real-time positioning data in industry's standard NMEA format. The GPSDO requires an external GPS antenna to be connected to the GPS antenna connector. As of *libsidekiq v4.16.1 / FPGA v3.14.1*, the GPS Disciplined Oscillator (GPSDO) functionality has been included on Matchstiq Z3u and can be enabled using libsidekiq API *skiq_gpsdo_enable()* function. When a GPS fix has been obtained by the on-board GPS, the FPGA uses the 1PPS signal to increase the accuracy of the radio's TCVCXO by automatically adjusting the DAC warp voltage. If no GPS fix can be obtained or is lost, the DAC warp voltage is kept at its current value; if no GPS fix is available on startup, the warp voltage is kept at its factory calibrated default value. As the FPGA is now in control of the warp voltage, this prevents its manual adjustment through the API.

GPSDO PERFORMANCE

The following measurements were collected with a Stationary Matchstiq Z3u / GPS receiver.

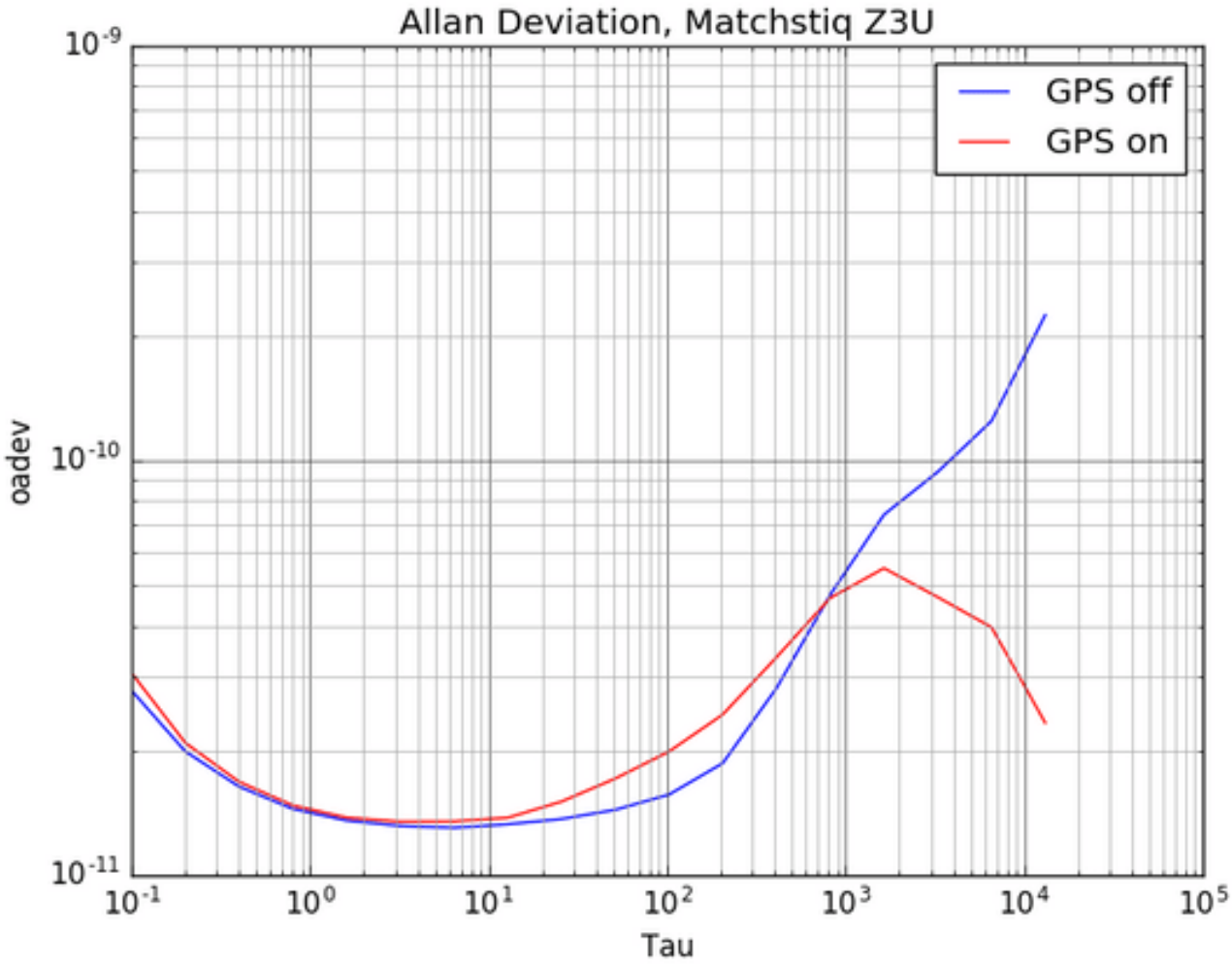

Figure 4: *Allan Deviation*

GPS / UART FUNCTIONALITY

As of package **z3u-core-epiq_1.0.0_arm64.deb** included in **install-z3u-complete_v1.0.0_20210302.sh**, a driver is available to control the GPIO signals connected to the GPS receiver.

GPS SYSFS

Control and status monitoring of the on-board GPS is provided through several sysfs entries. These entries are accessible from any application, the available entries are as follows:

Entry	Description	State
ant_bias_en	enables or disables antenna bias	**1** = enabled and **0** = disabled
has_fix	GPS Fix Status	**1** = GPS has fix and **0** = GPS does not have fix
power_en_n	Power to GPS module	**1**= disabled and **0** = enabled
reset	Control the RESET line to the GPS module	**1** = holds the device in reset and **0** = allows the device to run

Table 9: *Useful sysfs entries*

Linux sysfs GPS Control Examples

To read the GPS sysfs entries from the Z3u Linux command line:

```
sidekiq@z3u:~$ cd /sys/fs/skiq_gps/0

sidekiq@z3u:~$ ll
drwxr-xr-x 2 sidekiq root    0 Jun 12 11:02 ./
drwxr-xr-x 3 sidekiq root    0 Jun 12 11:01 ../
-rw-r--r-- 1 sidekiq root 4096 Jun 12 11:02 ant_bias_en
-r--r--r-- 1 sidekiq root 4096 Jun 12 11:02 has_fix
-rw-r--r-- 1 sidekiq root 4096 Jun 12 11:02 power_en_n
-rw-r--r-- 1 sidekiq root 4096 Jun 12 11:02 reset

sidekiq@z3u:~$ tail -n +1 *
==> ant_bias_en <==
0
==> has_fix <==
0
==> power_en_n <==
0
==> reset <==
0
```

When using an active GPS antenna which requires a DC bias voltage, you will need to enable the GPS antenna bias:

```
sidekiq@z3u:~$ cd /sys/fs/skiq_gps/0
sidekiq@z3u:~$ echo 1 | sudo tee ant_bias_en
```

To verify that the GPS antenna bias is enabled:

```
sidekiq@z3u:~$ ll
drwxr-xr-x 2 sidekiq root    0 Jun 12 11:02 ./
drwxr-xr-x 3 sidekiq root    0 Jun 12 11:01 ../
-rw-r--r-- 1 sidekiq root 4096 Jun 12 11:02 ant_bias_en
-r--r--r-- 1 sidekiq root 4096 Jun 12 11:02 has_fix
-rw-r--r-- 1 sidekiq root 4096 Jun 12 11:02 power_en_n
-rw-r--r-- 1 sidekiq root 4096 Jun 12 11:02 reset

sidekiq@z3u:~$ tail -n +1 *
==> ant_bias_en <==
1
==> has_fix <==
0
==> power_en_n <==
0
==> reset <==
0
```

After a several minutes, you should see that the GPS has a fix:

```
sidekiq@z3u:~$ ll
drwxr-xr-x 2 sidekiq root    0 Jun 12 11:02 ./
drwxr-xr-x 3 sidekiq root    0 Jun 12 11:01 ../
-rw-r--r-- 1 sidekiq root 4096 Jun 12 11:02 ant_bias_en
-r--r--r-- 1 sidekiq root 4096 Jun 12 11:02 has_fix
-rw-r--r-- 1 sidekiq root 4096 Jun 12 11:02 power_en_n
-rw-r--r-- 1 sidekiq root 4096 Jun 12 11:02 reset

sidekiq@z3u:~$ tail -n +1 *
==> ant_bias_en <==
1
==> has_fix <==
1
==> power_en_n <==
0
==> reset <==
0
```

GPS UART

The on-board GPS can provide NMEA-0183 messages through a UART device. A UART character device file is available as `/dev/ttyS0` and this device file may be used directly in any application (whether it uses libsidekiq or not) to receive NMEA-0183 messages.

This device file may also be used in conjunction with *gpsd* or *gpsmon*, for example.

`$ gpsmon /dev/ttyS0` or `$ gpsmon 127.0.0.1`

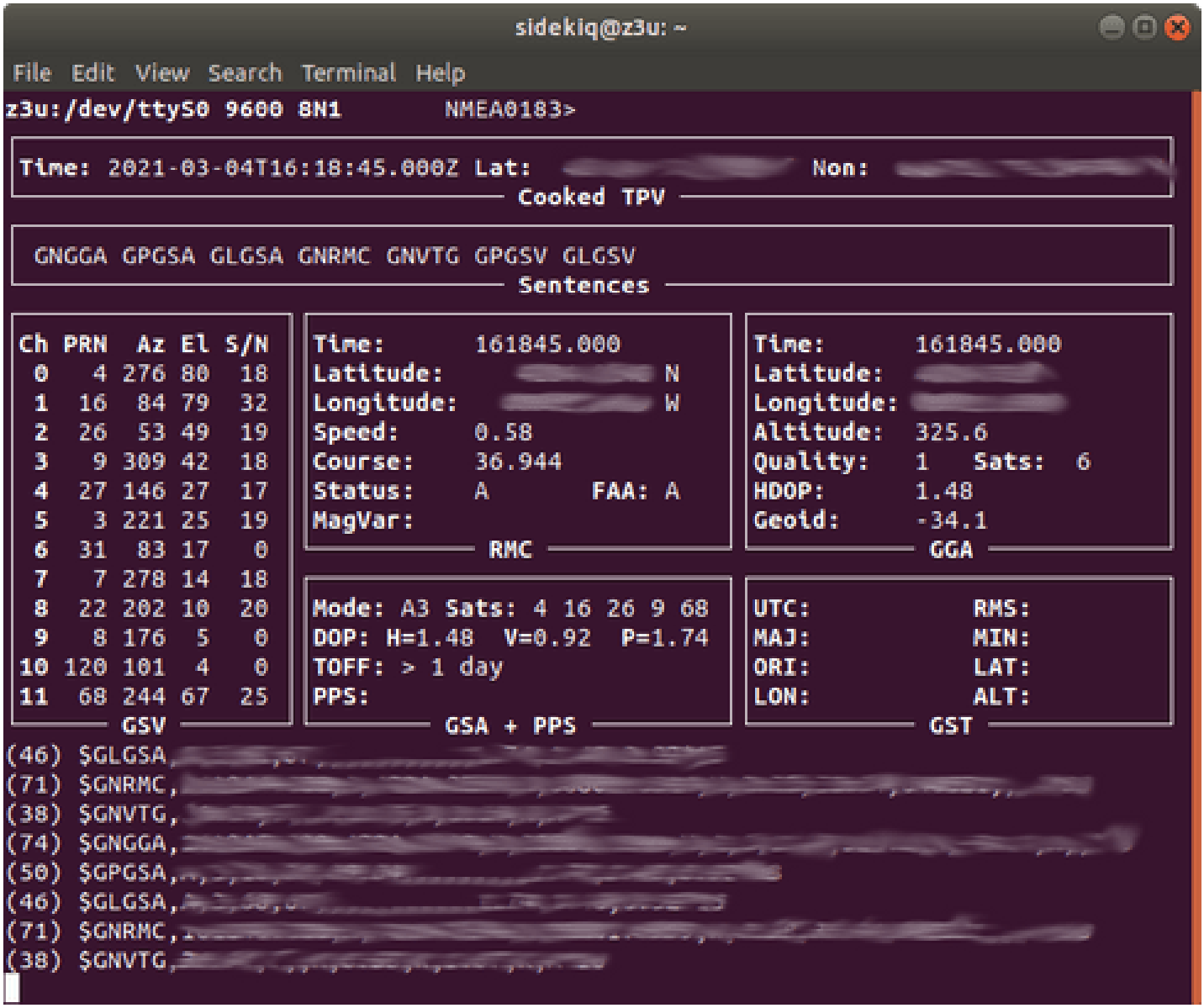

Figure 5: *GPSMON Example*

MATCHSTIQ Z3U SETUP

After you have removed and verified that all the package contents are present as outlined in the Matchstiq Z3u Package Contents, setup the system as follows:

The Matchstiq Z3u can be powered up via either the USB3 interface or the 7-17VDC input. The Matchstiq Z3u can be used by either running libsidekiq based application directly on the Z3u or on a host system running libsidekiq connected to the Z3u's network interface via the USB-C connection.

In the case of interfacing with the Matchstiq Z3u via a host system, a USB-C cable connecting to a USB3.0 capable host should be used to both power the Matchstiq Z3u as well as communicate to the host. Host systems supported are: Linux, Windows, MacOS, or an Android device.

In the case of applications running standalone directly on the Matchstiq Z3u, the unit can be powered with the power adapter provided and does not require a host device.

When power is applied to the Matchstiq Z3u, the unit will power on and the status LED will turn on a solid green. After several seconds, once the system is fully booted up, the Status LED will begin blinking.

INTERFACING TO MATCHSTIQ Z3U OVER USB

Once the Matchstiq Z3u is powered up and the Linux kernel has booted, the device appears as a USB Ethernet gadget network device with a default IP address of `192.168.0.15`. The host - the system that the Z3u is connected to - is offered the default IP address of `192.168.0.20` over DHCP (if enabled by the host).

DEFAULT LOGIN

USER IMAGE

The default username and password for the Ubuntu 18.04 user image is `sidekiq`. The `sidekiq` user is configured with sudo access. The user image is a full Ubuntu Linux distribution residing on the eMMC of the device. When the user image is booted, the Status LED turns on twice rapidly and then remains off for a longer period.

RECOVERY IMAGE

If there are any problems encountered when booting the Ubuntu 18.04 distribution, a reboot of the system will result in the recovery image being booted. The default root username for the recovery image is `root`, and the default password is `root`. Additionally, the Z3u default IP address is `192.168.0.15`. When the recovery image is booted, the Status LED blinks twice and then remains on for a longer period (this is inverted from the user image => inverted heatbeat status).

```
$ ping 192.168.0.15
PING 192.168.2.140(192.168.0.15) 56(84) bytes of data.
64 bytes from 192.168.0.15: icmp_req=1 ttl=64 time=0.323 ms
64 bytes from 192.168.0.15: icmp_req=2 ttl=64 time=0.390 ms
64 bytes from 192.168.0.15: icmp_req=3 ttl=64 time=0.225 ms
64 bytes from 192.168.0.15: icmp_req=4 ttl=64 time=0.405 ms
64 bytes from 192.168.0.15: icmp_req=5 ttl=64 time=0.343 ms
--- 192.168.2.140 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3999ms
rtt min/avg/max/mdev = 0.225/0.337/0.405/0.064 ms
```

If a network connection to the Z3u cannot be established, then the serial console will be required to recover the Z3u.

With a successful ping session complete, network connectivity between the Matchstiq Z3u and the host PC has been confirmed. From here, the user can proceed to establish a secure shell (SSH) connection to the Matchstiq Z3u unit from the host Linux PC.

```
$ ssh root@192.168.0.15
Last login: Wed Aug 26 14:16:37 2020 from 192.168.0.20
root@z3u:~#
```

Contact Epiq Solutions [2] for process on recovering system via network interface.

CHANGING THE DEFAULT IP ADDRESS

The IP address can be updated using the Linux console or the Micro-B USB serial console. There are two 2 uboot environment variables, `dev_ipaddr` is the Matchstiq Z3u usb network IP address and `host_ipaddr` is the host computer usb network IP address.

Linux console example of setting the Z3u IP address to `192.168.7.15` and the host computer IP address to `192.168.7.20`

```
$ sudo fw_setenv dev_ipaddr 192.168.7.15
$ sudo fw_setenv host_ipaddr 192.168.7.20
```

Serial console example of setting the Z3u IP address to `192.168.7.15` and the host computer IP address to `192.168.7.20` via U-Boot

Connect a USB-A (male) / USB Micro-B (male) cable from your host computer and the Z3u Micro-B USB serial console. From the host computer, open a serial console to the Z3u using screen, minicom, PuTTY, etc. `HostPC:~$ screen /dev/ttyUSB0 115200,cs`

Power up the Z3u and interrupt the boot process by hitting any key to stop autoboot

```
U-Boot 2018.01 (Mar 03 2021 - 15:10:50 +0000) Xilinx ZynqMP ZCU102 rev1.0

DRAM:  2 GiB
EL Level:       EL2
Chip ID:        zu3eg
MMC:   sdhci_transfer_data: Error detected in status(0x408000)!
mmc@ff160000: 0 (SD), mmc@ff170000: 1 (eMMC)
SF: Detected n25q1024a with page size 256 Bytes, erase size 64 KiB, total 128 MiB
In:    serial@ff010000
Out:   serial@ff010000
Err:   serial@ff010000
Board: Xilinx ZynqMP
Hit any key to stop autoboot:  0
ZynqMP>
ZynqMP> setenv dev_ipaddr 192.168.7.15
ZynqMP> setenv host_ipaddr 192.168.7.20
ZynqMP> saveenv
Saving Environment to SPI Flash...
SF: Detected n25q1024a with page size 256 Bytes, erase size 64 KiB, total 128 MiB
Erasing SPI flash...Writing to SPI flash...done
```

USB NETWORK CONFIGURATION

As of package **z3u-usb-net_1.0.0_arm64.deb** included in **install-z3u-complete_v1.0.0_20210302.sh**, the profile used to manage the USB network interface is configurable via a u-boot parameter. Valid profiles are:

- ECM (Ethernet Control Model)

- EEM (Ethernet Emulation Model)
- NCM (Network Control Model)

By default, the ECM profile is selected and used. This profile is supported by the majority of host systems. The EEM profile has been determined to provide an overall reduction in latency with the network connection and slight overall throughput improvement. However, it has been observed that not all host systems support this profile (such as some Android devices).

To update the profile used in the system, the u-boot environment `usb_network_profile` can be used or via the Z3u Linux command line:

To select EEM: `sidekiq@z3u:~$ sudo fw_setenv usb_network_profile eem`

To select ECM: `sidekiq@z3u:~$ sudo fw_setenv usb_network_profile ecm`

To view current selection: `sidekiq@z3u:~$ sudo fw_printenv usb_network_profile`

USB HOST MODE

Support for USB host mode is available on Matchstiq Z3u Rev-C hardware. Dynamic switching between host mode and device mode is not currently supported. The USB host mode setting can be non-persistent or persistent.

- **Non-persistent USB Host Mode**: this setting does not persist between power-cycles and allows users to boot into host mode for testing purposes without changing the default USB Device Mode setting.
- **Persistent USB Host Mode**: this setting persists between power-cycles

NON-PERSISTENT USB HOST MODE

Connect a USB-A (male) / USB Micro-B (male) cable from your host computer and the Z3u Micro-B USB serial console. From the host computer, open a serial console to the Z3u using screen, minicom, putty, etc. `HostPC:~$ screen /dev/ttyUSB0 115200,cs`

Power up the Z3u and interrupt the boot process by hitting any key to stop autoboot

At the `ZynqMP>` prompt, type `run $cp_img_sel` and then type `bootm $loadaddr#conf@2`

Example output of using host mode:

```
ZynqMP> run $cp_img_sel
SF: Detected n25q1024a with page size 256 Bytes, erase size 64 KiB, total 128 MiB
device 0 offset 0x3840000, size 0xbe5b29
SF: 12475177 bytes @ 0x3840000 Read: OK

ZynqMP> bootm $loadaddr#conf@2

## Loading kernel from FIT Image at 10000000 ...
   Using 'conf@2' configuration
   Trying 'kernel@0' kernel subimage
     Description:  Linux Kernel
     Type:         Kernel Image
     Compression:  gzip compressed
     Data Start:   0x100000c0
     Data Size:    6825848 Bytes = 6.5 MiB
     Architecture: AArch64
     OS:           Linux
     Load Address: 0x00080000
     Entry Point:  0x00080000
     Hash algo:    sha1
     Hash value:   639939cbaab6e9e994fe6cddeb6ef5a4b4567e21
   Verifying Hash Integrity ... sha1+ OK
## Loading fdt from FIT Image at 10000000 ...
   Using 'conf@2' configuration
   Trying 'fdt@1' fdt subimage
     Description:  Flattened Device Tree blob using USB host mode
     Type:         Flat Device Tree
     Compression:  uncompressed
     Data Start:   0x10bdbcb8
     Data Size:    39117 Bytes = 38.2 KiB
     Architecture: AArch64
     Hash algo:    sha1
     Hash value:   0a308ea8e0a84070452a6b28a46e100aac30a140
   Verifying Hash Integrity ... sha1+ OK
   Booting using the fdt blob at 0x10bdbcb8
## Loading fpga from FIT Image at 10000000 ...
   Trying 'fpga@0' fpga subimage
     Description:  FPGA
     Type:         FPGA Image
     Compression:  uncompressed
     Data Start:   0x1068c2cc
     Data Size:    5568778 Bytes = 5.3 MiB
     Load Address: 0x00f00000
     Hash algo:    md5
     Hash value:   55caeed40fc74e0bdba4b9f48fcb87c6
   Verifying Hash Integrity ... md5+ OK
   Loading fpga from 0x1068c2cc to 0x00f00000
  design filename = "z2_top;UserID=0XFFFFFFFF;Version=2018.3"
  part number = "xczu3eg-sbva484-1-i"
  date = "2020/12/11"
  time = "14:46:44"
  bytes in bitstream = 5568668
zynqmp_align_dma_buffer: Align buffer at 0000000000f0006e to 0000000000efff80(swap 0)
   Programming full bitstream... OK
   Uncompressing Kernel Image ... OK
   Loading Device Tree to 0000000007ff3000, end 0000000007fff8cc ... OK

Starting kernel ...

.....
```

Once it boots, you can view the on-board hubs running `lsusb` which will list the detected USB devices.

```
sidekiq@z3u:~$ lsusb
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

PERSISTENT USB HOST MODE

Control of the Z3u boot configuration can be achieved through use of u-boot arguments.

A new u-boot environment variable *usb_host_bootcmd* can be introduced. The user will need to set and configure this to perform boot using the *conf2* of the FIT image.

1. From the Z3u serial console or SSH session:

```
sidekiq@z3u:~$ sudo fw_setenv usb_host_bootcmd "run \${cp_img_sel} && bootm
\${loadaddr}#conf@2"
```

2. You'll need to change the boot command to use the new environment variable just introduced. This can be updated by running:

```
sidekiq@z3u:~$ sudo fw_setenv bootcmd "run usb_host_bootcmd"
```

3. Once the environment variables have been configured, the Matchstiq Z3u can be rebooted:

```
sidekiq@z3u:~$ sudo reboot
```

If you monitor the serial console output when when booting with host mode selected, you should see the following:

```
## Loading kernel from FIT Image at 10000000 ...
Using 'conf@2' configuration
```

Once it boots, you can view the on-board hubs running `lsusb` which will list the detected USB devices.

```
sidekiq@z3u:~$ lsusb
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

To switch back to the USB device mode configuration, the *bootcmd* u-boot environment variable can be updated to its default value by running:

```
sidekiq@z3u:~$ sudo fw_setenv bootcmd "run default_bootcmd"
```

MATCHSTIQ Z3U BASIC USAGE

Matchstiq Z3u includes a completely integrated Linux computer that controls the operation of the card. Linux userspace applications can be executed either standalone on the Matchstiq Z3u itself or on the host system, controlling the Matchstiq Z3u via the network interface presented via the USB Ethernet Gadget.

Please note that the Matchstiq Z3u serial number can be found on the Matchstiq Z3u housing (white label) under the magnetic mount, if installed, and is also stored on a read-only Linux partition located in `/calibration/matchstiq_z3u_sn.txt` *The Matchstiq Z3u PCB sub-assembly 9xxx serial number is stored in eeprom and can be read using the* `version_test` *application located in* `/home/sidekiq/sidekiq_image_current/test_apps`

INCLUDED APPLICATIONS

Several applications are included with the Matchstiq Z3u in order to help you test and verify your setup, such as our standard libsidekiq command-line test applications.

LIBSIDEKIQ TEST APPLICATIONS

Test applications are located in `/home/sidekiq/sidekiq_image_current/test_apps`

```
sidekiq@z3u:~$ cd /home/sidekiq/sidekiq_image_current/test_apps/
```

A user can display libsidekiq and fpga version information with the **version_test** application. The application should return results that look something like the following:

```
sidekiq@z3u:~$ ~/sidekiq_image_current/test_apps/version_test
1 card(s) found: 0 in use, 1 available!
Card IDs currently used     :
Card IDs currently available: 0
Info: initializing 1 card(s)...
SKIQ[3396]: <INFO> libsidekiq v4.13.255-z3u-dev-20200826 (g1ff8b75f9)
version_test[3396]: <INFO> Sidekiq card 0 is serial number=9X0J, Z3U (rev B) (part ES032201-B0-
00)
version_test[3396]: <WARNING> FPGA capabilities indicate no support for reading/writing flash
for card 0
version_test[3396]: <INFO> Sidekiq card 0 FPGA v3.14.0, (date 20081217, FIFO size unknown)
version_test[3396]: <INFO> Sidekiq card 0 is configured for an internal reference clock
version_test[3396]: <INFO> Loading calibration data for Sidekiq Z3U, card 0
**********************************************************
* libsidekiq v4.13.255-z3u-dev-20200826
**********************************************************
**********************************************************
* Sidekiq Card 0
  Card
    accelerometer present: true
    part type: Z3U
    part info: ES032201-B0-00
    serial: 9X0J
    xport: custom
  FPGA
    version: 3.14.0
    git hash: 0x1eefb308
    build date (yymmddhh): 20081217
    tx fifo size: unknown
  RF
    reference clock: internal
    reference clock frequency: 40000000 Hz
version_test[3396]: <INFO> Unlocking card 0
```

RAW I/Q CAPTURE

A user can perform an RF capture of I/Q samples using the default configuration by executing the *rx_samples* application as follows:

```
sidekiq@z3u:~$ ~/sidekiq_image_current/test_apps/rx_samples -c 0 --handle=A1 -r 7e6 -b 5.6e6 -f
1e9 -d /tmp/out
Info: using Rx handle A1
Info: initializing card 0...
SKIQ[3435]: <INFO> libsidekiq v4.13.255-z3u-dev-20200826 (g1ff8b75f9)
rx_samples[3435]: <INFO> Sidekiq card 0 is serial number=9X0J, Z3U (rev B) (part ES032201-B0-
00)
rx_samples[3435]: <INFO> libiio v0.19 (tag a74f874)
rx_samples[3435]: <WARNING> FPGA capabilities indicate no support for reading/writing flash for
card 0
rx_samples[3435]: <INFO> Sidekiq card 0 FPGA v3.14.0, (date 20081217, FIFO size unknown)
rx_samples[3435]: <INFO> Sidekiq card 0 is configured for an internal reference clock
rx_samples[3435]: <INFO> Loading calibration data for Sidekiq Z3U, card 0
rx_samples[3435]: <INFO> card 0: number of tx channels supported 1, number of rx channels
supported 2
rx_samples[3435]: <INFO> RF IC version 5.2.255
rx_samples[3435]: <INFO> Default warp voltage not currently stored (user) (card=0)
rx_samples[3435]: <INFO> Default warp voltage not currently stored (factory) (card=0)
rx_samples[3435]: <INFO> No stored default warp voltage is available for card 0, using the
library default
Info: initialized card 0
Info: opened file /tmp/out.a1 for output
Info: tcvcxo warp voltage left at factory setting
Info: packed mode disabled
Info: tunable RX LO frequency range = 47000000Hz to 6000000000Hz
Info: RX calibration mask configured as 0x0
Info: configured Rx LO freq to 1000000000 Hz
Info: configured auto gain mode
Info: actual sample rate is 7000000.000000, actual bandwidth is 5600000
Info: acquiring 100000 words at 1018 words per block
Info: num blocks to acquire is 99
Info: configured for I/Q data mode
Info: starting 1 Rx interface(s)
Info: stopping 1 Rx interface(s)
Info: done receiving, start write to file for hdl 0
Info: Done without errors!
rx_samples[3435]: <INFO> Unlocking card 0
sidekiq@z3u:~$
```

This command will save I/Q samples to a file named /tmp/out.a1 using values for 7 Msps sample rate, 5.6 MHz channel bandwidth, 1 GHz tune frequency. The data is stored in the file as 16-bit I/Q pairs with ‘I’ samples stored in the upper 16-bits of each word, and ‘Q’ samples stored in the lower 16-bits of each word. Additional available options are described by executing *./rx_samples -h*

INTERFACING TO MATCHSTIQ Z3U OVER SERIAL CONSOLE

Access to the Linux serial console via the USB Micro-B connector requires the user to connect a USB Micro-B cable between the Matchstiq Z3u and a host PC USB 2.0 port running a serial terminal emulator program such as minicom, PuTTY, or screen. The USB Micro-B interface of the Z3u will enumerate as a USB serial port (typically at `/dev/ttyUSB0` or similar on the host Linux system). The default Linux serial console on Matchstiq Z3u runs at a baud rate of 115.2 kbps, with 8 data bits, one stop bit, and no parity bits.

```
Ubuntu 18.04.4 LTS z3u ttyPS0
z3u login: sidekiq
Password:
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.14.0-xilinx-v2018.3 aarch64)
 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
This system has been minimized by removing packages and content that are
not required on a system that users do not log into.
To restore this content, you can run the 'unminimize' command.
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
sidekiq@z3u:~$
```

LINUX BOARD SUPPORT PACKAGE OPTIONS FOR MATCHSTIQ Z3U

OVERVIEW

The Matchstiq Z3u board support package (BSP) consists of a collection of software & FPGA components needed to allow the Matchstiq Z3u to boot Linux and execute a user's radio applications. These components include the uboot bootloader, Linux kernel, FPGA reference design, and other ancillary components such as the root filesystem and Linux device tree.

BSP BUILD PROCESS

The Matchstiq Z3u BSP utilizes the PetaLinux 2018.3 utility (refer to [5]) to build the kernel and u-boot. The reference BSP and the build process is outlined in [2].

The user root filesystem is Ubuntu 18.04. Standard Debian packages for the aarch64 CPU architecture can be installed as desired.

FPGA RESOURCE AVAILABILITY

Epiq Solutions' BSP includes an FPGA reference design that is optimized for resource efficiency to maximize the FPGA resources available to support the inclusion of custom processing blocks. The resource utilization of the FPGA is shown below.

Resource	Used	Total	Util%
Logic Cells	52.36K	154K	34%
CLB LUTs	10353	70560	14.67%
Block RAM Tile	39.5	216	18.29%*
DSPs	0	360	0.00%

* Optional buffering included to support sustained high-rate Rx and Tx streaming between CPU and FPGA fabric in Zynq; can be reduced depending on CPU ↔ FPGA streaming requirements

Table 10: *Z3u FPGA Reference Design Resources*

FPGA FEATURE SET AND CUSTOMIZATION

Epiq Solutions' BSP includes the Sidekiq FPGA reference design as its basis, which allows customers to migrate their designs from existing Sidekiq cards to Matchstiq Z3u. The Sidekiq FPGA reference design implements key features outlined below:

- **Timestamp management to support time-tagged receive and on-time transmit:** This allows a customer's application to accurately time tag each block of baseband I/Q data that is received so that all upstream processing blocks have a precise sense of time. This also provides a means to keep track of overflow / underflow conditions if the CPU can't keep up. Additionally, on-time transmit is supported to ensure that baseband I/Q data is upconverted and transmitted out at RF at precisely the requested time. This capability is key to enabling any sort of slotted time division multiple access (TDMA) communications system such as those found in cellular communications systems.
- **Pulse Per Second (PPS) integration:** This allows an externally generated PPS signal to be utilized by Matchstiq Z3u for the purpose of keeping track of time. This also provides a means to synchronize the time between physically separated Matchstiq Z3u units, where each Z3u unit receives a PPS signal.
- **Flexible buffering between the CPU (PS) and FPGA (PL):** In order to ensure sustained high transport streaming between the PL and the PS, it is necessary to include the appropriate amount of buffering to ensure no samples are dropped. Epiq Solutions' FPGA reference design allows the buffering scheme to be optimized to support the customer's requirements.
- **Well-defined “user app” space in the FPGA reference design to simplify custom IP block integration:** The standard Sidekiq FPGA reference design is architected to easily allow customers to integrate their own IP processing blocks into the signal processing chain with minimal effort. This is the same architecture used on all existing Sidekiq cards.

MATCHSTIQ Z3U PLATFORM DEVELOPMENT KIT

OVERVIEW

The Matchstiq Z3u Platform Development Kit (PDK) provides additional access to the debug connector of the Matchstiq Z3u as well as additional I/O debug access.

The following sections outlines the basic features available with the Matchstiq Z3u Platform Development Kit.

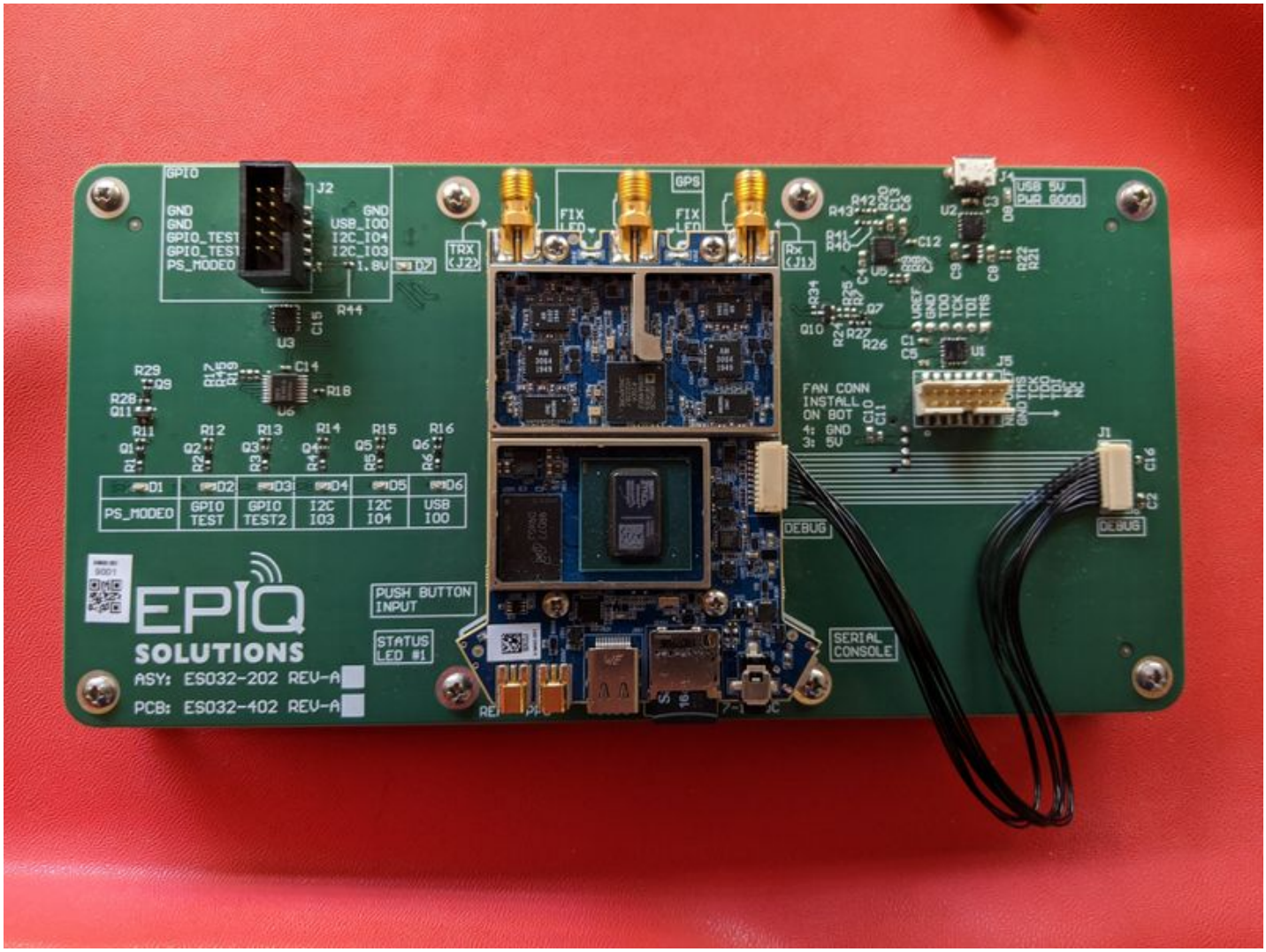

Figure 6: *Matchstiq Z3u PDK Rev-A*

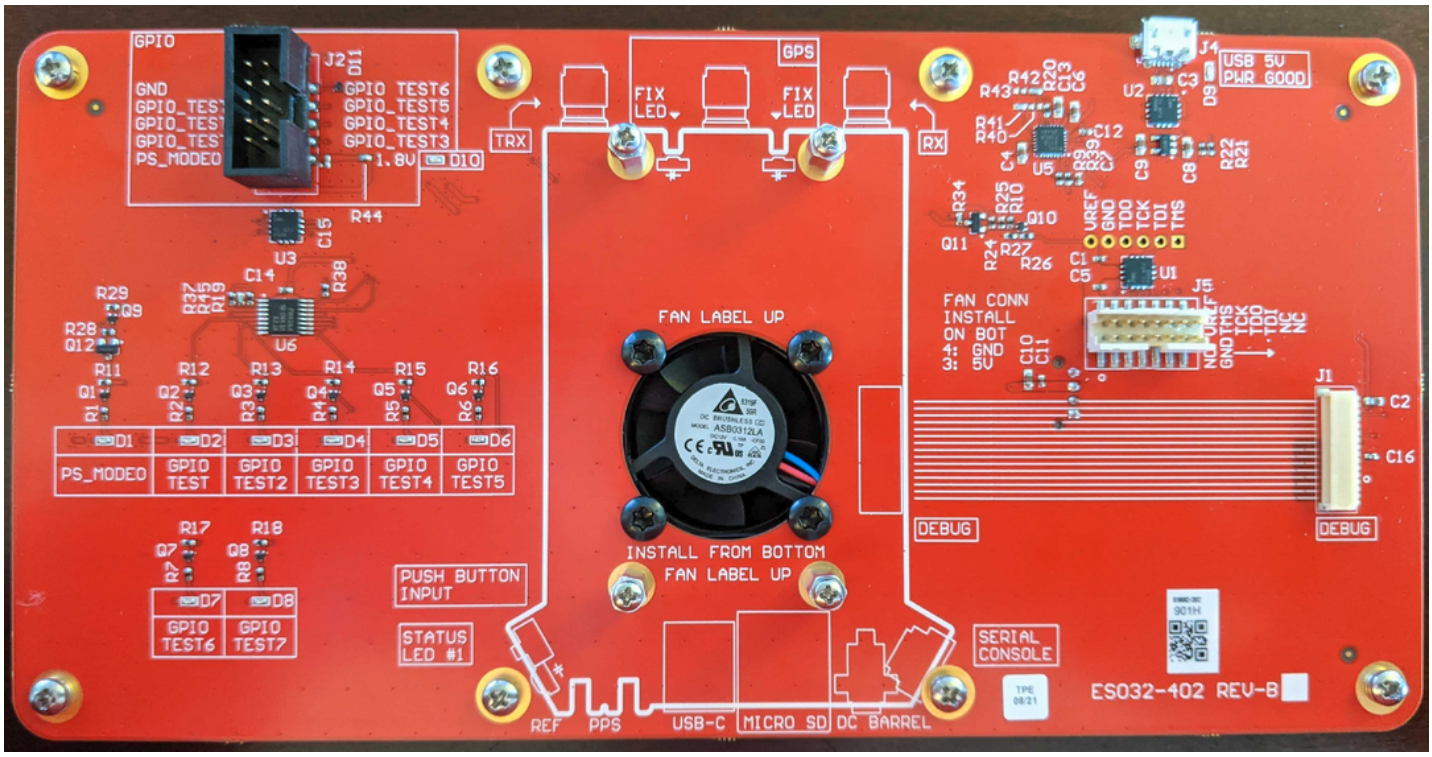

Figure 7: *Matchstiq Z3u PDK Rev-B*

DEBUG HEADER (J1)

The Debug Header (Rev-A P/N: SM10B-SRSS-TB(LF)(SN) and Rev-B P/N: SM15B-SRSS-TB(LF)(SN)) of the Z3u PDK provides the ability to interface with JTAG, PL GPIOs, and PS GPIO signals.

Pin	Rev-A	Rev-B
1	V1P8_VREF	V1P8_VREF
2	GND	GND
3	JTAG_TCK	JTAG_TCK
4	GND	GND
5	JTAG_TMS	JTAG_TMS
6	JTAG_TDO	JTAG_TDO
7	JTAG_TDI	JTAG_TDI
8	GPIO_TEST	GPIO_TEST
9	PS_MODE0	PS_MODE0
10	GPIO_TEST2	GPIO_TEST2
11	N/A	GPIO_TEST3
12	N/A	GPIO_TEST4
13	N/A	GPIO_TEST5
14	N/A	GPIO_TEST6
15	N/A	GPIO_TEST7

Table 11: *Z3u PDK Debug Header (J1)*

Please note that the Matchstiq Z3u PCB debug connector pin orientation changed, pin 1 on Rev. D is on the opposite side from where it was on Rev. B.

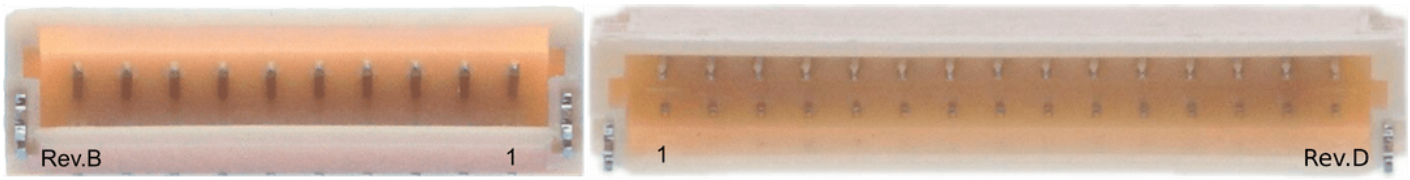

Figure 8: *Matchstiq Z3u PCB Debug Connector*

GPIO HEADER (J2)

The GPIO header (P/N: 72454-010LF) of the Matchstiq Z3u PDK provides the ability to interface with PL GPIOs, and a PS GPIO.

Pin	Rev-A	Rev-B
1	V1P8_IO	V1P8_IO
2	PS_MODE0_IO_EXP	PS_MODE0_IO_EXP
3	I2C_IO3	GPIO_TEST3
4	GPIO_TEST	GPIO_TEST
5	I2C_IO4	GPIO_TEST4
6	GPIO_TEST2	GPIO_TEST2
7	USB_IO0	GPIO_TEST5
8	GND	GND
9	GND	GND
10	GND	GND

Table 12: *Z3u PDK GPIO Header (J2)*

FAN HEADER (J3)

The Z3u PDK Fan Header (P/N: S04B-PASK-2(LF)(SN), mates with PAP-04V-S) provides the power to the Z3u PDK cooling fan.

Pin	Description
1	NC
2	NC
3	5VDC (VBUS)
4	GND

Table 13: *Z3u PDK Fan Header (J3)*

Name	Zynq Name	Linux Sysfs	Rev-A PDK Debug Pin
PL GPIO_TEST	EMIO0	418	8
PL GPIO_TEST2	EMIO1	419	10

Table 14: *Z3u Rev-B GPIO*

Name	Zynq Name	Zynq Pin	Linux Sysfs	Rev-B PDK Debug Pin
PL GPIO_TEST	EMIO2	E8	418	8
PL GPIO_TEST2	EMIO3	F4	419	10
PL GPIO_TEST3	EMIO0	D6	416	11
PL GPIO_TEST4	EMIO1	A8	417	12
PS GPIO_TEST5	MIO74	D18	412	13
PS GPIO_TEST6	MIO75	B17	413	14
PS GPIO_TEST7	MIO45	A11	383	15

Table 15: *Z3u Rev-C GPIO*

PL GPIO_TEST

The PL GPIO_TEST provides a GPIO interface accessible by the PL, connected to pin IO_L6P_HDGC_AD6P_26. This pin is configured as EMIO2 in the FPGA and is controlled via sysfs entries located at `/sys/class/gpio/gpio418`.

PL GPIO_TEST2

The PL GPIO_TEST2 provides a GPIO interface accessible by the PL, connected to pin IO_T3U_N12_65. This pin is configured as EMIO3 in the FPGA and is controlled via sysfs entries located at `/sys/class/gpio/gpio419`.

PS_MODE0

The PS_MODE0 provides a GPIO interface accessible by the PS, connected to pin PS_MODE0. Additionally, when powering on the Matchstiq Z3u, it can be used to control the boot mode to either boot from the onboard QSPI flash or from an image on the microSD card. When driven low, the QSPI flash is selected as the boot device (this is the default), alternatively, the microSD is selected.

USB MICRO-B CONNECTOR (J4)

The USB Micro-B connector (J4) of the Matchstiq Z3u PDK provides the ability to interface with PL GPIOs, and a PS GPIO via an IO expander. The GPIO connector signals are described in the table below.

Pin	Signal
1	VBUS
2	D-
3	D+
4	ID
5	GND
6	GND
7	GND
8	NC (not connected)
9	NC (not connected)
10	GND
11	GND

Table 16: *Z3u PDK USB Micro-B Connector (J4)*

JTAG HEADER (J5)

The JTAG header is Amphenol P/N 98424-G52-14ALF. It will allow the standard Xilinx 14-pin JTAG cable to be attached.

Pin	Description	Pin	Description
1	NC	**2**	VREF (V1P8_VREF)
3	GND	**4**	TMS
5	GND	**6**	TCK
7	GND	**8**	TDO
9	GND	**10**	TDI
11	GND	**12**	NC
13	PGND	**14**	HALT (NC)

Table 17: *Z3u PDK JTAG Xilinx Header (J5)*

MATCHSTIQ Z3U PDK STATUS LEDS

The status LEDs provide a visual status indicator and are described below.

LED	Z3u Rev-B / PDK Rev-A	Z3u Rev-C / PDK Rev-B
D1	PS_MODE0	PS_MODE0
D2	GPIO_TEST	GPIO_TEST
D3	GPIO_TEST2	GPIO_TEST2
D4	I2C_IO3	GPIO_TEST3
D5	I2C_IO4	GPIO_TEST4
D6	USB_IO0	GPIO_TEST5
D7	Z3u (V1P8_VREF) PWR GOOD	GPIO_TEST6
D8	USB 5V (VBUS) PWR GOOD	GPIO_TEST7

Table 18: *Z3u PDK Status LEDs*

POWERING UP PLATFORM DEVELOPMENT KIT

When using the Matchstiq Z3u installed in the PDK, both the PDK and Matchstiq Z3u are powered independently. The Matchstiq Z3u is powered by the the normal means (either through the USB-C connector or the DC barrel jack).

To power the PDK, the PDK's USB Micro B connector can be used with either a USB Micro-B to USB-A connection to a host system (such as a PC) or with the provided wallwart.

In general, since the PDK powers the thermal management for the unhoused Matchstiq Z3u, the PDK should be powered on *prior* to the Matchstiq Z3u.

LINUX CONSOLE SERIAL PORT

The Linux Console is available via the USB Micro-B connector of the Matchstiq Z3u. The default Linux serial console on Matchstiq Z3u runs at a baud rate of 115.2 kbps, with 8 data bits, one stop bit, and no parity bits. The default username/password of sidekiq/sidekiq can be used to log in to the system.

CONTROLLING GPIO

Access to the GPIO is supported through the **sysfs** entries in Linux, located at **/sys/class/gpio/** or via **gpiod**. Details on interfacing with the GPIO via **sysfs** is described in detail in [3]. All GPIO definitions for the Matchstiq Z3u begin at an offset of **338**. The MIO GPIO are under direct control of the PS, whose mapping is outlined in the table below. PL_GPIO are controllable via the FPGA and control via the sysfs can be enabled. In order to allow control of the PL GPIO via the Linux userspace, the pin must be mapped to an EMIO offset, as outlined in [4]. From that point, the Linux sysfs offset begins at 338+80=418. For example on Z3u Rev-C, for accessing a GPIO mapped to EMIO2, the Linux sysfs value used should be 338 + 80 + 0 = 418.

Name	ZynqUS+ GPIO Name	Linux Sysfs
PL GPIO_TEST	EMIO0	418
PL GPIO_TEST2	EMIO1	419

Table 19: *Z3u Rev-B GPIO Mapping*

Name	ZynqUS+ GPIO Name	Zynq Pin	Linux Sysfs
PL GPIO_TEST	EMIO2	E8	418
PL GPIO_TEST2	EMIO3	F4	419
PL GPIO_TEST3	EMIO0	D6	416
PL GPIO_TEST4	EMIO1	A8	417
PS GPIO_TEST5	MIO74	D18	412
PS GPIO_TEST6	MIO75	B17	413
PS GPIO_TEST7	MIO45	A11	383

Table 20: *Z3u Rev-C GPIO Mapping*

LINUX SYSFS GPIO CONTROL EXAMPLE

The following steps outline how to configure PL GPIO_TEST for Z3u as an output and configure the value to a logical high. *Please note that by default, the PL GPIO signals are exported.*

sidekiq@z3u:~$ echo 1 | sudo tee ant*bias*en

1. Configure the GPIO as an output

```
sidekiq@z3u:~$ echo out | sudo tee /sys/class/gpio/gpio418/direction
```

2. Verify the GPIO is configured as an output

```
sidekiq@z3u:~$ cat /sys/class/gpio/gpio418/direction
out
```

3. Configure the GPIO as a logical 1

```
sidekiq@z3u:~$ echo 1 | sudo tee /sys/class/gpio/gpio418/value
```

4. Verify the GPIO value is set to 1

```
sidekiq@z3u:~$ cat /sys/class/gpio/gpio418/value
1
```

GPIOD EXAMPLE

The GPIOs cannot be controlled via GPIOD if they are already under control of the sysfs entries. If it is desired to control the GPIO via GPIOD, then the GPIO must be “unexported” as shown in the **Controlling GPIO** example below.

Viewing all GPIO

All GPIO and their default assignments can be viewed with `gpioinfo`.

```
sidekiq@z3u:~$ sudo gpioinfo
[sudo] password for sidekiq:

gpiochip0 - 174 lines:
line  0:     unnamed      unused  input active-high
line  1:     unnamed      unused  input active-high
line  2:     unnamed      unused  input active-high
line  3:     unnamed      unused  input active-high
line  4:     unnamed      unused  input active-high
line  5:     unnamed      unused  input active-high
line  6:     unnamed      unused  input active-high
line  7:     unnamed      unused  input active-high
line  8:     unnamed      unused  input active-high
line  9: "heartbeat" "led-heartbeat" output active-high [used]
line 10:     unnamed      unused  input active-high
line 11:     unnamed      unused  input active-high
line 12:     unnamed      unused  input active-high
line 13:     unnamed      unused  input active-high
line 14:     unnamed      unused  input active-high
line 15:     unnamed      unused  input active-high
line 16:     unnamed      unused  input active-high
line 17:     unnamed      unused  input active-high
line 18:     unnamed      unused  input active-high
line 19:     unnamed      unused  input active-high
line 20:     unnamed      unused  input active-high
line 21:     unnamed      unused  input active-high
line 22:     unnamed      unused  input active-high
line 23:     unnamed      unused  input active-high
line 24:     unnamed      unused  input active-high
line 25:     unnamed      unused  input active-high
line 26:     unnamed      unused  input active-high
line 27:     unnamed      unused  input active-high
line 28: "gps_reset_n" "gps_reset_n" output active-low [used]
line 29: "gps_power_en" "gps_power_en" output active-high [used]
line 30:     unnamed      unused  input active-high
line 31:     unnamed      unused  input active-high
line 32: "gps_ant_bias" "gps_ant_bias" output active-high [used]
line 33:     unnamed      unused  input active-high
line 34:     unnamed      unused  input active-high
line 35:     unnamed      unused  input active-high
line 36:     unnamed      unused  input active-high
line 37:     unnamed      unused  input active-high
line 38:     unnamed      unused  input active-high
line 39:     unnamed      unused  input active-high
line 40:     unnamed      unused  input active-high
line 41:     unnamed      unused  input active-high
line 42:     unnamed      unused  input active-high
line 43:     unnamed      unused  input active-high
line 44:     unnamed      unused  input active-high
line 45:     unnamed      unused  input active-high
line 46:     unnamed      unused  input active-high
line 47:     unnamed      unused  input active-high
line 48:     unnamed      unused  input active-high
line 49:     unnamed      unused  input active-high
line 50:     unnamed      unused  input active-high
line 51:     unnamed      unused  input active-high
line 52:     unnamed      unused  input active-high
line 53:     unnamed      unused  input active-high
line 54:     unnamed      unused  input active-high
line 55:     unnamed      unused  input active-high
line 56:     unnamed      unused  input active-high
line 57:     unnamed      unused  input active-high
line 58:     unnamed      unused  input active-high
line 59:     unnamed      unused  input active-high
line 60:     unnamed      unused  input active-high
line 61:     unnamed      unused  input active-high
line 62:     unnamed      unused  input active-high
line 63:     unnamed      unused  input active-high
line 64:     unnamed      unused  input active-high
line 65:     unnamed      unused  input active-high
line 66:     unnamed      unused  input active-high
line 67:     unnamed      unused  input active-high
```

```
line 68:     unnamed      unused  input active-high
line 69:     unnamed      unused  input active-high
line 70:    "ext_pb"    "ext_pb"  input active-high [used]
line 71:     unnamed      unused  input active-high
line 72:     unnamed      unused  input active-high
line 73:     unnamed      unused  input active-high
line 74:     unnamed      unused  input active-high
line 75:     unnamed      unused  input active-high
line 76:     unnamed      unused  input active-high
line 77:     unnamed      unused  input active-high
line 78:     unnamed      unused  input active-high
line 79:     unnamed      unused  input active-high
line 80:     unnamed     "sysfs"  input active-high [used]
line 81:     unnamed     "sysfs"  input active-high [used]
line 82:     unnamed      unused  input active-high
line 83:     unnamed      unused  input active-high
line 84:     unnamed      unused  input active-high
line 85:     unnamed      unused  input active-high
line 86:     unnamed      unused  input active-high
line 87:     unnamed      unused  input active-high
line 88:     unnamed      unused  input active-high
line 89:     unnamed      unused  input active-high
line 90:     unnamed      unused  input active-high
line 91:     unnamed      unused  input active-high
line 92:     unnamed      unused  input active-high
line 93:     unnamed      unused  input active-high
line 94:     unnamed      unused  input active-high
line 95:     unnamed      unused  input active-high
line 96:     unnamed      unused  input active-high
line 97:     unnamed      unused  input active-high
line 98:     unnamed      unused  input active-high
line 99:     unnamed      unused  input active-high
line 100:     unnamed      unused  input active-high
line 101:     unnamed      unused  input active-high
line 102:     unnamed      unused  input active-high
line 103:     unnamed      unused  input active-high
line 104:     unnamed      unused  input active-high
line 105:     unnamed      unused  input active-high
line 106:     unnamed      unused  input active-high
line 107:     unnamed      unused  input active-high
line 108:     unnamed      unused  input active-high
line 109:     unnamed      unused  input active-high
line 110:     unnamed      unused  input active-high
line 111:     unnamed      unused  input active-high
line 112:     unnamed      unused  input active-high
line 113:     unnamed      unused  input active-high
line 114:     unnamed      unused  input active-high
line 115:     unnamed      unused  input active-high
line 116:     unnamed      unused  input active-high
line 117:     unnamed      unused  input active-high
line 118:     unnamed      unused  input active-high
line 119:     unnamed      unused  input active-high
line 120:     unnamed      unused  input active-high
line 121:     unnamed      unused  input active-high
line 122:     unnamed      unused  input active-high
line 123:     unnamed      unused  input active-high
line 124:     unnamed      unused  input active-high
line 125:     unnamed      unused  input active-high
line 126:     unnamed      unused  input active-high
line 127:     unnamed      unused  input active-high
line 128:     unnamed      unused  input active-high
line 129:     unnamed      unused  input active-high
line 130:     unnamed      unused  input active-high
line 131:     unnamed      unused  input active-high
line 132:     unnamed      unused  input active-high
line 133:     unnamed      unused  input active-high
line 134:     unnamed      unused  input active-high
line 135:     unnamed      unused  input active-high
line 136:     unnamed      unused  input active-high
line 137:     unnamed      unused  input active-high
line 138:     unnamed      unused  input active-high
line 139:     unnamed      unused  input active-high
line 140:     unnamed      unused  input active-high
line 141:     unnamed      unused  input active-high
```

```
line 142:     unnamed      unused  input active-high
line 143:     unnamed      unused  input active-high
line 144:     unnamed      unused  input active-high
line 145:     unnamed      unused  input active-high
line 146:     unnamed      unused  input active-high
line 147:     unnamed      unused  input active-high
line 148:     unnamed      unused  input active-high
line 149:     unnamed      unused  input active-high
line 150:     unnamed      unused  input active-high
line 151:     unnamed      unused  input active-high
line 152:     unnamed      unused  input active-high
line 153:     unnamed      unused  input active-high
line 154:     unnamed      unused  input active-high
line 155:     unnamed      unused  input active-high
line 156:     unnamed      unused  input active-high
line 157:     unnamed      unused  input active-high
line 158:     unnamed      unused  input active-high
line 159:     unnamed      unused  input active-high
line 160:     unnamed      unused  input active-high
line 161:     unnamed      unused  input active-high
line 162:     unnamed      unused  input active-high
line 163:     unnamed      unused  input active-high
line 164:     unnamed      unused  input active-high
line 165:     unnamed      unused  input active-high
line 166:     unnamed      unused  input active-high
line 167:     unnamed      unused  input active-high
line 168:     unnamed      unused  input active-high
line 169:     unnamed      unused  input active-high
line 170:     unnamed      unused  input active-high
line 171:     unnamed      unused  input active-high
line 172:     unnamed      unused  input active-high
line 173:     unnamed      unused  input active-high
```

Controlling GPIO

For example, from the Z3u Linux command line, to gain control of PL_GPIO_TEST via GPIOD, you must first run:

```
sidekiq@z3u:~$ sudo echo 418 > /sys/class/gpio/unexport
```

The value of the GPIO can then be configured with **gpioset**

```
sidekiq@z3u:~$ sudo gpioset 0 80=1
```

To query the value, if configured as an input, **gpioget** can be used.

```
sidekiq@z3u:~$ sudo gpioget 0 80
0
```

DEFAULT MEMORY PARTITION AND FILESYSTEM CONFIGURATION

Address Range	Size	Name	Additional Details
0x00000000-0x00800000	8M	boot	Bootloader(FSBL and u-boot)
0x00800000-0x00840000	262k	boot-env	u-boot environment
0x00840000-0x03840000	50M	kernel	Recovery Image(kernel and root filesystem)
0x03840000-0x06840000	50M	user	User image (kernel and FPGA)
0x06840000-0x06880000	262k	cal	Calibration Data

Table 21: *QSPI flash memory device default memory partition table*

Additional non-volatile storage space can be added in to the system through the internal eMMC 128 GB device which also contains the root filesystem. Non-violatile storage can be further expanded via a removable microSD card.

USING PETALINUX TO UPDATE UBOOT AND LINUX KERNEL

BUILDING

Contact Epiq Solutions [2] for details.

UPDATING

Contact Epiq Solutions [2] for details.

ADDITIONAL PACKAGES

Since Ubuntu is a Debian based system, standard Debian package installation using dpkg is supported.

MATCHSTIQ Z3U RECOVERY

If the bootloader (u-boot) is accidentally overwritten, the Matchstiq Z3u will no longer have the ability to boot. Depending the specific failure, there are a few approaches recovering a unit. Contact Epiq Solutions support for details.

USING MICROSD FOR DEFAULT KERNEL / DEVICE TREE / FPGA

If you’d like to boot from the microSD card, the 'force_sdboot' u-boot environment variable can be set on the Z3u.

Specifically, when in Linux, to set the *force_sdboot* variable, you can run the following on the Z3u:

```
sudo fw_setenv force_sdboot 1
```

Then, either issuing a reboot command or power cycling the Z3u will result in the kernel, device tree, and FPGA to be loaded from the contents of the microSD card.

For details on properly formatting and partitioning the microSD card, refer to [6].

At a minimum, the following files must be present on the boot partition: BOOT.BIN, image.ub, and system.dtb.

POWER CONSUMPTION

The power consumption of Matchstiq Z3u varies depending on the configuration and application of the radio. For these measurements, all tests were performed with root filesystem version 1.8.0 `/opt/VERSIONS`, libsidekiq v4.17.3 and FPGA v3.16.2.

Ambient Temp varied between 23 ºC - 25 ºC DUT power from 9VDC supplied to barrel jack. This utilizes the onboard SMPS (~90% efficient). SSH connection via USB3.0 interface, temperature sensor data being logged to SDcard.

Test Application	Power Consumption in Watts (9VDC supplied to barrel jack)
rx_benchmark -r 15.36e6 --handle=A1	4.94 W
rx_benchmark -r 30.72e6 --handle=A1	5.33 W
rx_benchmark -r 15.36e6 --handle=A1,A2	5.46 W
rx_benchmark -r 30.72e6 --handle=A1,A2	5.98 W
xcv_benchmark -r 30.72e6	5.46 W

Table 22: *Example power consumption measurements for Matchstiq Z3u*

MATCHSTIQ Z3U MOUNTING AND CABLING TO TABLET

Figure 9: *Matchstiq Z3u Bottom View*

1. Determine the desired location & orientation to mount the Matchstiq Z3u on the phone/tablet.
2. Clean the area with an isopropyl alcohol wipe.
3. Remove the adhesive liner from the metallic pad / Matchstiq Z3u.
4. Align the metallic pad / Matchstiq Z3u and attach to phone/tablet.
5. Attach antenna to Rx or TRx SMA connector.
6. Attach USB-C cable from Matchstiq Z3u USB port to USB-C port on phone/tablet.

Figure 10: *Matchstiq Z3u Mag Mount*

MATCHSTIQ Z3U MECHANICAL OUTLINE

A dimensioned mechanical drawing of Matchstiq Z3u is shown below. In addition, a 3D model (in STP format) is also available. Please contact Epiq Solutions [2] for this model.

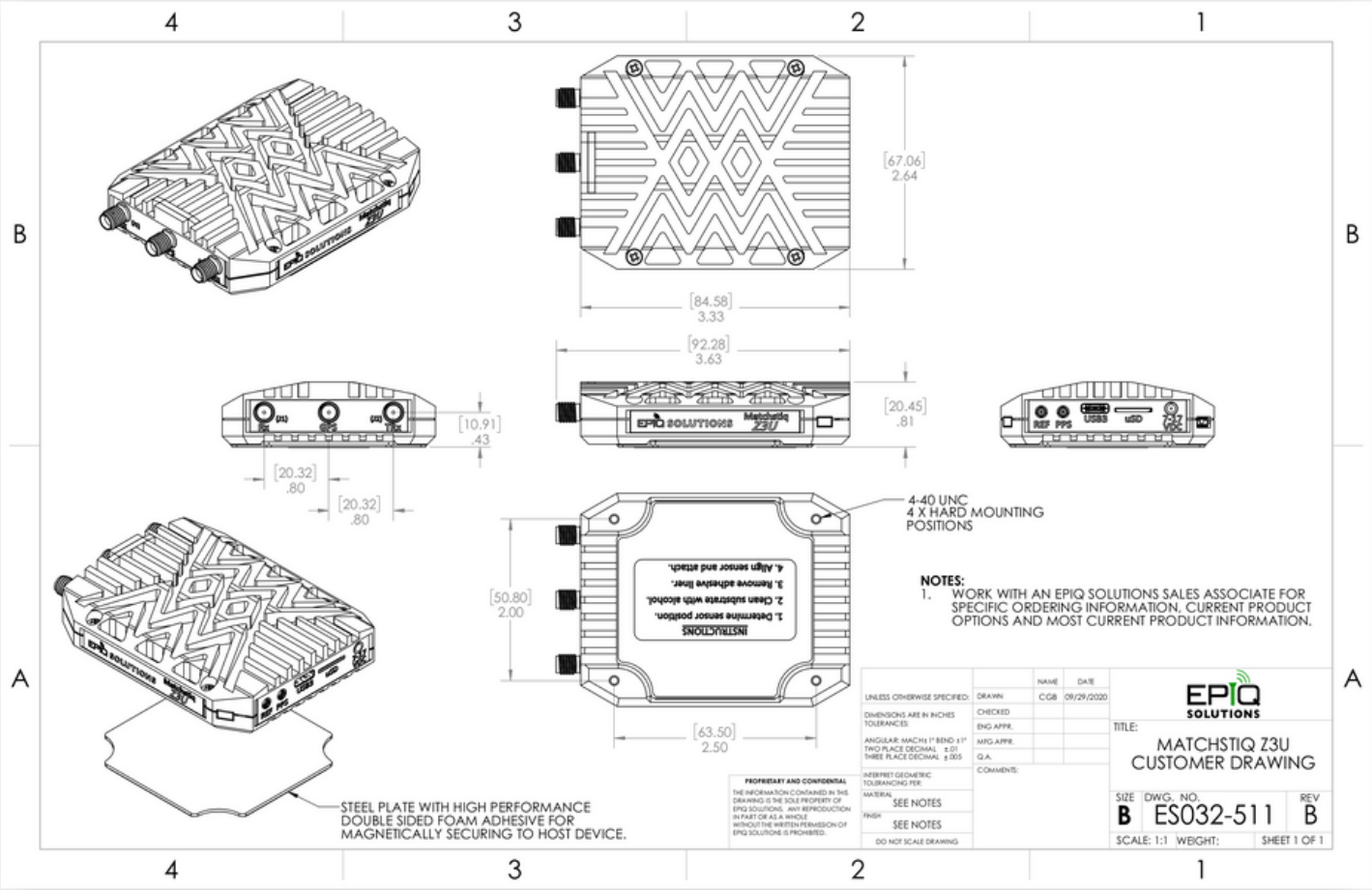

Figure 11: *Matchstiq Z3u Mechanical Outline*

MATCHSTIQ Z3U THERMAL

The Matchstiq Z3u was isolated from the bench top by being placed on two 3mm shims (to eliminate conduction cooling into bench) with the TOP facing up towards ceiling and bottom facing down toward bench top. The Z3u and was allowed to reach steady-state over a time period of about 2 hours.

- Ambient Temp varied between **23-25 ºC**
- Rise was calculated from the delta of the Temp Sensor and recorded ambient temp at time of test completion.
- DUT power from 9 VDC supplied to barrel jack. This utilizes the onboard SMPS (~90% efficient).
- SSH connection via USB 3.0 interface, temperature sensor data being logged to SDcard.

		Temperature Rise above Ambient (ºC_rise)				
Test Application	**Power Draw (W)**	**PCB Temp Sensor**	**SoC Temp Sensor**	**Top Housing Temp**	**Bottom Housing Temp**	**Avg (PCB, SoC) Thermal Resistance to Ambient (Sensor Avg)**
./rx_benchmark --rate=15.36e6	4.94 W	36 ºC	45 ºC	36 ºC	34 ºC	8.5 ºC per Watt
./rx_benchmark --rate=30.72e6	5.33 W	42 ºC	49 ºC	37 ºC	35 ºC	8.5 ºC per Watt
./rx_benchmark --handle=A1,A2 --rate=15.36e6	5.46 W	44 ºC	50 ºC	38 ºC	36 ºC	8.6 ºC per Watt
./xcv_benchmark --rate=30.72e6	5.46 W	43 ºC	50 ºC	39 ºC	36 ºC	8.5 ºC per Watt
./rx_benchmark --handle=A1,A2 --rate=30.72e6	5.98 W	46 ºC	53 ºC	41 ºC	38 ºC	8.3 ºC per Watt

Table 23: *Matchstiq Z3u Thermal*

APPENDIX A – MATCHSTIQ Z3U RF FRONT END

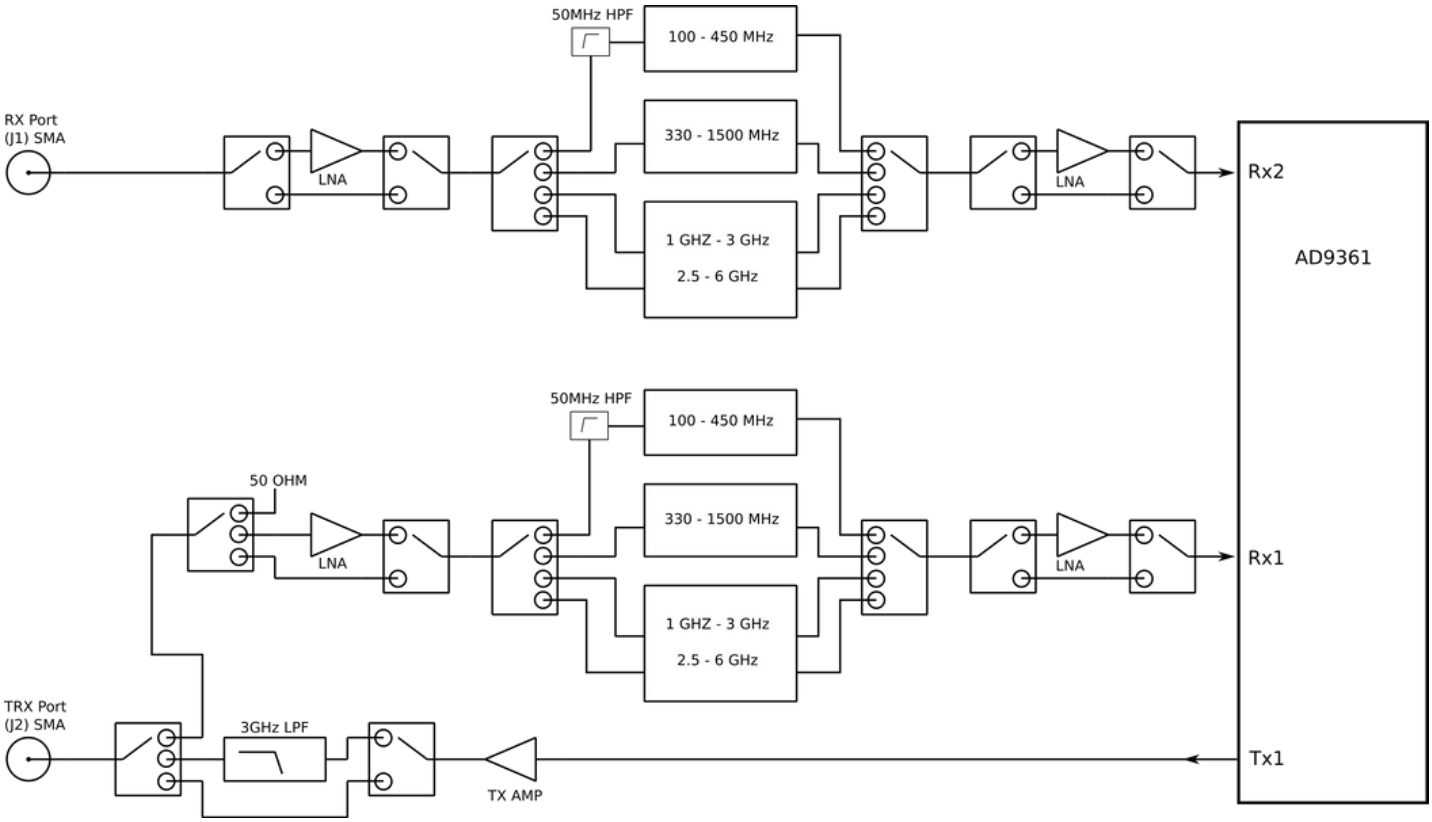

Figure 12: *Matchstiq Z3u RF Front End*

- The appropriate path through the filter banks is selected automatically based on the tune frequency.
- For frequencies < 100 MHz, the HPF bypass filter internal to the AM3090 is selected. the discrete 50 MHz HPF is then in series with the AM3090 150 MHz LPF, creating a 50 - 150 MHz passband filter.

Band	LO Tune Range	Band	LO Tune Range
1	47 – 135 MHz	11	366 – 475 MHz
2	135 – 145 MHz	12	475 – 625 MHz
3	145 – 150 MHz	13	625 – 800 MHz
4	150 – 162 MHz	14	800 – 1175 MHz
5	162 – 175 MHz	15	1175 – 1500 MHz
6	175 – 190 MHz	16	1500 – 2100 MHz
7	190 – 212 MHz	17	2100 – 2775 MHz
8	212 – 230 MHz	18	2775 – 3360 MHz
9	230 – 280 MHz	19	3360 – 4600 MHz

10	280 – 366 MHz	20	4600 – 6000 MHz

Table 24: *Matchstiq Z3u Rx pre-select filter bands*

APPENDIX B - STATEMENT OF VOLATILITY

Model	Matchstiq Z3u
Part Number	ES032-211
Manufacturer	Epiq Solutions
Address	3740 Industrial Avenue Rolling Meadows, IL 60008

Table 25: *Model, Part Number, and Manufacturer Info*

Memory Type	Memory Size	User Modifiable	Purpose	Process to Clear
External DDR3L RAM	2 GB	Yes	Application usage	Power-off
On-Chip FPGA BRAM	7.6 Mb	Yes	Application usage	Power-off
On-chip FPGA DRAM	1.8 Mb	Yes	Application usage	Power-off
On-Chip Memory	256 KB	Yes	Application usage	Power-off

Table 26: *Matchstiq Z3u Volatile Memory*

Memory Type	Memory Size	User Modifiable	Removable	Purpose	Process to Clear
QSPI Flash	128 MB	Yes	No	Holds bootloader and Linux kernel	Cleared with Linux utilities
eMMC NAND Flash	128 GB	Yes	No	Used for the root filesystem (i.e. Linux distribution) as well as persistent storage and application installation.	Cleared with Linux utilities
EEPROM	16 KB	No	No	Contains part number, revision, and serial number information	Must be returned to factory to clear
microSD card *optional, user supplied	up to 1 TB	Yes	Yes	Used for additional storage or alternative boot options	Cleared with Linux utilities

Table 27: *Matchstiq Z3u Non-Volatile Memory*

APPENDIX C - PREDICTED FAILURE RATE AND MTBF

Listed below is the Failure Rate and MTBF for the ES032-201-C Sidekiq Z3U Assembly.

The Calculations are derived from Relyence Reliability Software and based off a fixed/ground/controlled operating environment with an ambient temperature of 25°C.

Part Number	ES032-201-C
Description	ASY SKIQ-Z3U PCB Assy
Failure Rate (fpmh)	3.468940
MTBF (hours)	288,272.49
Calculation Model	Telcordia Issue 4
Operating Environment	Fixed/Ground/Controlled
Ambient Temperature	25°C

Table 28: *Sidekiq Z3U Failure Rate & MTBF*