**IA473 ARM Board Communication Protocol**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Name** | **Function** | **Date** | **Signature** |
| **Written by** | Kenny Liu | System Engineering | 2015-05-15 |  |
| **Checked by** | Value Xie | System Engineering |  |  |
| **Validation By** |  | Project Manager |  |  |
| **Approved by** |  | Director |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **DOCUMENT UPDATE** | | | |
| Version | Date | Name |  |
| V0.1 | 2015/05/15 | Kenny Liu | Draft Version |
| V0.2 | 2015/05/20 | Kenny Liu | Add description for communication between Diags and MAC |
| V0.3 | 2015/05/21 | Kenny Liu | Add description for Datalogger and status machine |
| V0.4 | 2015/05/22 | Kenny Liu | Add detail description for Datalogger data |
| V0.5 | 2015/05/28 | Kenny Liu | Add debug dummy |
| V0.6 | 2015/10/17 | Libo Deng  &Rui Liu | Modify detail description for Datalogger data, and description of instruction ACK for error or simple result |

Index

[1 Introduction 3](#_Toc420101923)

[1.1 Objectives 3](#_Toc420101924)

[1.2 Reference 3](#_Toc420101925)

[1.3 Terminology and Alias 3](#_Toc420101926)

[2 System Diagram and Configuration 3](#_Toc420101927)

[2.1 Diagram 3](#_Toc420101928)

[2.2 Configuration 4](#_Toc420101929)

[2.2.1 Command Interface Configuration 4](#_Toc420101930)

[2.2.2 Diags Interface Configuration 5](#_Toc420101931)

[2.2.3 Datalogger Interface configuration 5](#_Toc420101932)

[3 Command Message Format 6](#_Toc420101933)

[3.1 DUT & Console Command Format 6](#_Toc420101934)

[3.2 TCP Message Format 6](#_Toc420101935)

[3.3 Escape Character '$' 7](#_Toc420101936)

[4 ACK Message Format 7](#_Toc420101937)

[4.1 Ack to DUT & Console command 7](#_Toc420101938)

[4.2 ACK to TCP command 7](#_Toc420101939)

[4.3 ACK for error or simple result 8](#_Toc420101940)

[5 Datalogger Message illustration 8](#_Toc420101941)

[5.1 Data format 8](#_Toc420101942)

[5.1.1 Frame format for Datalogger 8](#_Toc420101943)

[5.1.2 Timestamp format 8](#_Toc420101944)

[5.1.3 First Byte Flag 9](#_Toc420101945)

[5.1.4 Other Bytes for Data 9](#_Toc420101946)

[5.2 Example: 10](#_Toc420101947)

[5.2.1 ADC data example: 10](#_Toc420101948)

[5.2.2 Timestamp example 10](#_Toc420101949)

[6 Command Example 10](#_Toc420101950)

[7 Command List 11](#_Toc420101951)

[8 Status Machine for Command Execution 11](#_Toc420101952)

# Introduction

## Objectives

This document provides a detailed illustration for the communication protocol of IA473 FCT fixture. It can be a design reference for the software of MAC and MCU.

## Reference

|  |  |  |
| --- | --- | --- |
| **#** | **File name** | **No. of documents** |
| 1 | ARM Board Command List |  |
| 2 |  |  |

## Terminology and Alias

| **Alias** | **Description** |
| --- | --- |
| TBD | To Be Decided |
| UUT | Unit Under Test |
| DUT | Device Under Test |
| TCP | Transfer Control Protocol |
| CLI | Command Line Interface |

# System Diagram and Configuration

## Diagram

IA473 FCT automation Tester’s communication function is shown as below:



There are four kinds of message transported in the fixture.

The first one is the commands and ACK to control fixture. If the ARM Board get these commands, it should act as the commands and reply ACK to the command source. There are three command source: MAC, DUT, Console. All of them are marked with red in the block diagram. The DUT control fixture with a standalone UART. MAC should record these commands log.

The second kind of message is the data exchanged between DUT Diags and MAC. From the serial port of Diags, all of the data will be sent to MAC through Ethernet.

The third kind of message is the current data and time stamp for Datalogger. It will be sent to MAC directly.

If we open

## Configuration

### Command Interface Configuration

There are three ways to control Fixture:

* + UART/SPAM from DUT

-- 115200 8-N-1 : Baudrate 115200, Data 8 bits, Parity None, Stop 1 bit, Flow control Non

* + Fixture MCU UART Console

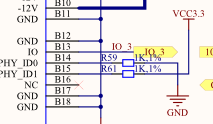
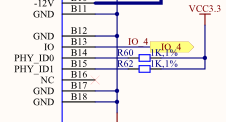
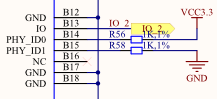
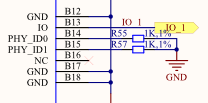
-- 115200 8-N-1 : Baudrate 115200, Data 8 bits, Parity None, Stop 1 bit, Flow control Non

* + Fixture RPC over TCP/IP:

--Port: 7600 (TCP)

--Default IP: 169.254.1.32

There are 4 ARM Boards in one fixture. If they are connected with bridge board. The resistors on bridge board will set the IP address for each channel.



**IP Address configuration**

|  |  |
| --- | --- |
| PHY\_ID0, PHY\_ID1(Setting Resistors) | IP Address |
| 00 | **169.254.1.32** |
| 10 | **169.254.1.33** |
| 01 | **169.254.1.34** |
| 11 | **169.254.1.35** |

### Diags Interface Configuration

* + Diags over TCP/IP:

--Port for DUT UART: 7601 (TCP)

--Port for SPAM: 7602 (TCP)

After the client in MAC binding socket, the MAC can exchange data with Diags.

If the client amount is more than one? Should all of the clients will communicate with DUT?

### Datalogger Interface configuration

TCP Port: 7603

Command Interface:

**datalogger open -samplerate**

**datalogger close**

# Command Message Format

## DUT & Console Command Format



* **Command Body & Parameters: ASCII code**
* **Optional CRC: 2 Bytes, Not forced. '#' is the flag of this field. Another 8 bits is the CRC value of protocol field.**
* **End Flag: \r\n**

## TCP Message Format



* **Command Body : ASCII code**
* **[ID]: 16 bytes, command serial number, this body is only used in Ethernet**
* **End Flag: \r\n**
* **For the Ethernet, several frames can be put together into one TCP packet.**

## Escape Character '$'

* Escape Character for end flag

If the content of parameter have an end flag "\r\n". We should translate it to be "$\r\n". The "$" indicate these two bytes is not real ending flag.

* Detecting Frame

Detecting frame just use start flag and end flag. If we meet a end flag after start flag, we can consider it is one frame for the command. And we can start to find another frame.

# ACK Message Format

## Ack to DUT & Console command

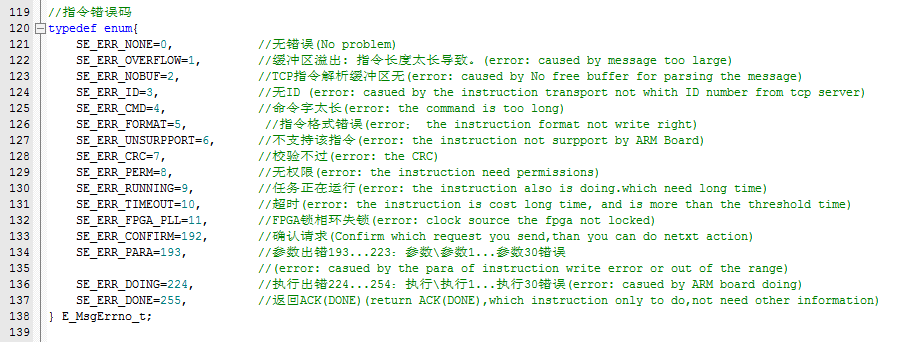


## ACK to TCP command



## ACK for error or simple result

Error Number: If any error, the result should be filled with error number.



If the command is executed rightly, the result can be filled with 'DONE'.

# Datalogger Message illustration

**MAC record the PP\_BATT\_VCC current profile in the test process. The ADC is 20-bits, the raw data will be sent to MAC, MAC calculate the result with calibration. ARM Board can record the time stamp and upload it to MAC. DUT can use trigger IO to let FPGA record a special time stamp. Calibration data for Data logger is stored in the EEPROM of Data Logger PCBA**

## Data format

* **ADC data frame length: 6 bytes**
* **Timestamp frame length: 7 bytes**

### Frame format for Datalogger



### Timestamp format

**UTC format, 32bits for second, 14bits for millisecond.**



UTC format : Time base is 1970-01-01T00:00:00Z.

|  |  |
| --- | --- |
| Time | Second |
| 1 min | 60 |
| 1 hour | 3600 |
| 1 day | 86400 |
| 1 week | 604800 |
| 1 month (30.44 day) | 2629743 |
| 1year (365.24 day) | 31556926 |

### First Byte Flag

**First Byte:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
| A | B | X | X | X | X | X | X |

**BIT7: 1 -- This is first bytes for one frame.**

**0-- This is not the first byte for ADC data frame or timestamp.**

**BIT6: If BIT7 = 1, BIT6 =0 : This is the first byte for ADC.**

**If BIT7 = 1, BIT6 =1 : This is the first byte for timestamp. Timestamp is 7 Bytes.**

### Other Bytes for Data

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| BIT7 | BIT6 | BIT5 | BIT4 | BIT3 | BIT2 | BIT1 | BIT0 |
| A | Data | Data | Data | Data | Data | Data | Data |

**The first bit is used for differentiating first byte of one frame.**

## Example:

### ADC data example:

**Data frame: 8A 2B 5A 73 20 00。 The Sample data is A576B990h。**



### Timestamp example

**Timestamp frame:**

**C5, 2A, 7B, 45, 58, 01, 48,**



**Valid data: UTC field(binary format): 0101,0110,0010,0001,1011,1110,0110,0101**

**Millisecond field(binary format): 00,0001,0000,0001**

**Timestamp: 2015/10/17 11:20:05 257ms**

# Command Example

**Tcxo Frequency measurement example**

* **Console**

**Send: frequency measure (LSM,100,0U) \r\n**

**Answer: ACK(32768.051 Hz) \r\n**

* **DUT**

**Send: frequency measure (tcxo) #0x34\r\n**

**Answer: ACK(32768.051 Hz) #0x52 \r\n**

**0x34, 0x52—CRC value**

* **Ethernet**

**Send: [0x0102030405060708090A0B0C0D0E0F10] frequency measure (tcxo) \r\n**

**Answer: [0x0102030405060708090A0B0C0D0E0F10] ACK (32768.051 Hz) \r\n**

**[0x0102030405060708090A0B0C0D0E0F10] –frame ID**

# Command List

Please check the document:< ARM Board Command List>

# Status Machine for Command Execution

