# List of Commands and other packets

#define SERIAL\_ID\_PKT 0

#define VERSION\_PKT 1

#define DATE\_TIME\_PKT 2

#define CURRENT\_PKT 3

#define BOARD\_ID\_PKT 5

#define SWITCH\_CONNECT\_CHK\_PKT 6

#define SWITCH\_RELEASE\_CHK\_PKT 7

#define SINGLE\_WIRE\_OFFSET\_PKT 8

#define SWITCH\_STATUS\_PKT 9

#define SET\_SERIAL\_ID\_PKT 0x10

#define SET\_DATE\_TIME\_PKT 0x12

#define SET\_BOARD\_ID\_PKT 0x15

#define SET\_SINGLE\_WIRE\_OFFSET\_PKT 0x18

#define VOLTAGE\_FAIL\_PKT 0x20

#define FINAL\_PKT 0x30

#define START\_TEST 0x31

#define GET\_VERSION\_PKT 0x41

#define GET\_DATE\_TIME\_PKT 0x42

#define GET\_BOARD\_ID\_PKT 0x45

#define GET\_INT\_FIRM\_VERSION 0x50

#define BOARD\_VOLTAGES\_DATA 0x99

#define RESET\_JIG 0xA0

## RTC Commands

typedef enum

{

NO\_COMMAND,

STOP\_PC\_OPERATION = 0x01,

START\_RTC\_CALIB = 0x32,

GET\_PPM\_OFFSET,

GET\_TEMP\_OFFSET,

}e\_jigOperationCommands;

**Vendor App (testing Module) sends below commands**

1. GET\_INT\_FIRM\_VERSION, 11 retries, 5 sec timeouts
2. BOARD\_ID\_PKT, 2 retries, 5 sec timeouts
3. START\_TEST, 0 retries, 260 sec timeouts
   1. Start test ask for GET\_DATE\_TIME\_PKT
   2. PC Returns SET\_DATE\_TIME\_PKT with date time
   3. Then returns below pkt
      1. CURRENT\_PKT
      2. SWITCH\_CONNECT\_CHK\_PKT
      3. VOLTAGE\_FAIL\_PKT
      4. FINAL\_PKT

The Start command takes around 3+ minutes to return all packets so we can't add any retries to this command, otherwise, cycle time will be doubled.

**Solution**: Add ACK in Testing board firmware so that without waiting 3 minutes, we can add more retries if firmware missed some retries.

Issues I observed (We have tested 20 PCBs and 6 PCBs failed with this reason):

1. Sends GET\_BOARD\_NUMBER command

2. Timeout (3 seconds) happens

3. Resend GET\_BOARD\_NUMBER command again and receive the response of the first command.

4. Send START\_TEST command and firmware returns the GET\_BOARD\_NUMBER command (2nd command) response and START\_TEST command got missed.

Current solution: timeout increases from 3 seconds to 5 seconds, but this is not the correct solution.

An ACK must be added if the command takes more time to return the response.

As soon as APP gets some **complete** command from JIG, It stops its operation of calibration and testing and display the results.

APP send some command to start the JIG Controller Operation

App keeps listening the UART Data Asynchronously and do action based on the response received from the JIG

User Goes to PC for APP

User Presses the JIG

After this, if APP needs any specific data from JIG then it can query the same individually.

All other operation happens as usual

APP sends ‘start test’ command

APP receives BOARD\_ID\_PKT packet and stores the results

APP receives GET\_INT\_FIRM\_VERSION packet and stores the results

APP receives RTC operation packet and stores the results

App keeps listening the UART Data Asynchronously and do action based on the response received from the JIG

**Final Flow**

APP will keep waiting to get the firmware version

Soon after receiving the firmware version, App will send the command to start the RTC Calibration process.

After this, normal software flow will follow…

Once APP gets the response of RTC Calibration, it will send the **Stop\_PC\_comm** command