PUSH SDK HTTP/
Protocol V3.2.0

Release by 2018-06-30

Revision History

Date	Version	Description	Author
3/26/2013	1.0	Revised version from original	Simon Zhang
3/18/2014	2.0	Revise the quite a number of PUSH protocols, and rewrite query protocols	Simon Zhang
10/10/2014	2.1	Refurbish the protocol specs and detail, Add shortkey API, multiple verifying modes and data sync part.	Simon Zhang
2/17/2014	2.2	Remove the TODO and other none support features from the document to avoid confuse	Simon Zhang
5/27/2014	2.3	Add the face template command	Simon Zhang
2/17/2015	2.4	Add the misc information missing from firmware 6.8.0 (build 640) + /push sdk 2.0.3.636+ update.	Simon Zhang
02/2017	3.1	Revised on existing APIs: 3. Clear and Reset Data 7.2. Add or Modify User Information 7.5. Add or Modify User's Face Template 7.15 User TimeZone/Lockout Schedules 7.16. Ring Bell Schedule 7.17.3 New Shortkey settings 12. Remote Enrollment 14. View Last Punch New Features: 2.4.2. Upload the Attendance Records(TFT series) 3. Clear and Reset Data	Ryan

		7.3. Add or Modify User Extended Information 7.11.3. Delete SMS 7.13. Jobcode (Enhanced Work Code) 7.14 Tipcode(Tip Entry) 11. Data Query 13. Self Service 15. Manually Add Punches **** is symbol of modification to TFT series specifically.**	
05/2017	3.1.1.	Add keyword “FACE” for configuring the uface to upload face enrollment in real time. **** is symbol of modification to TFT series specifically.**	Ryan
10/2017	3.1.2	2.2.4 Add verification combination code for Face reader **** is symbol of modification to TFT series specifically.**	Ryan
06/2018	3.2.0	7.3. Add user level fingerprint threshold configuration. 13.3 Self Service Dialogue **** is symbol of modification to TFT series specifically.**	Ryan

Part I: PUSH SDK Introduction

1. Document Audience

This document is serving as technical reference for Clock customers and third party developers who are going to integrate Clock into third party time attendance system or workforce management system.

2. Terminology

Upload：“Upload” within this document refers to transferring data from the device to the server.

Download：“Download” within this document refers to transferring data from the server to the device.

Part II: PUSH SDK Features

1. Supported System

Server：Any web server support HTTP protocol.

2. PUSH Features：

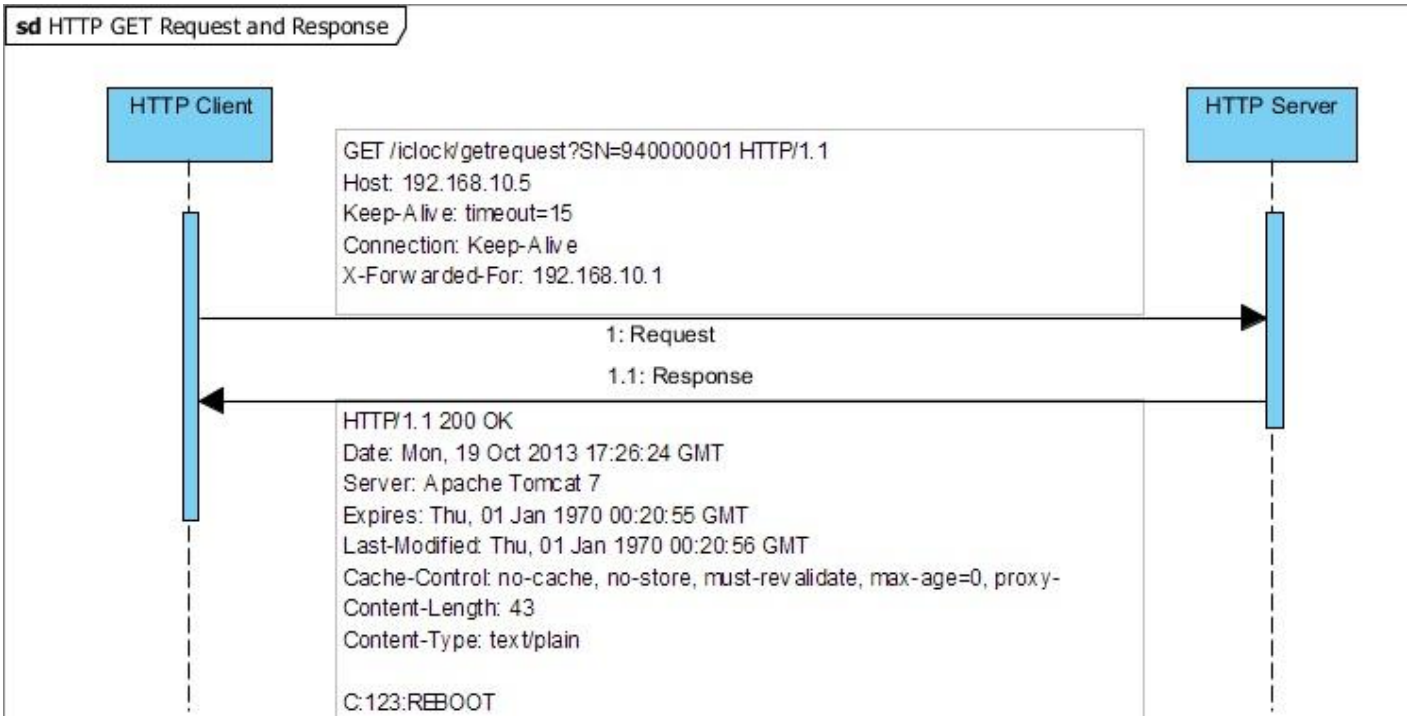
- All the main devices supported, e.g. the black and white screen and the / 3.5/3/8 inch color screen devices.
- Improved and standardized server access interface.
- Database access interface to different devices.
- Time synchronization of the device from Server, with time zone setup on the device (including half-hour increment).
- Automatically upload the attendance records
- Automatically upload the attendance photos
- Automatically upload the system log
- Automatically upload the newly added/updated users’ info, including users’ basic information, fingerprint and face template.
- Automatically delete the oldest data, when the attendance records overflow. (The deleting data amount can be customized.)
- Receive the command notice at the server (UDP command notice, which needs internet support.).
- System commands sent by the server, e.g. ls etc.
- Check the data update on the device from the server.
- Delete attendance records, attendances pictures and all the data on the device from the server, including attendance records, attendances pictures and user info etc.
- Gather the device information from the server.
- Set the device options from the server.
- Restart the device from the server.
- Reload the device options from the server.
- Control the device to open locks from the server.
- Deactivate the alarm of the device from the server.

- Check and transfer the new data by the server.
- Read the device files from the server.
- Download the files from the server to the device, including device firmware upgrade.
- Download the messages from the server to the device.
- Support add/update/delete the user info and fingerprint on the device from the server.
- Support the server control of the device to register the users' fingerprint, not including the Face device series yet.
- Upload/download/delete the users' pictures from the server side.
- Support DNS.
- Support HTTPS communication.

Part III: PUSH SDK Specs

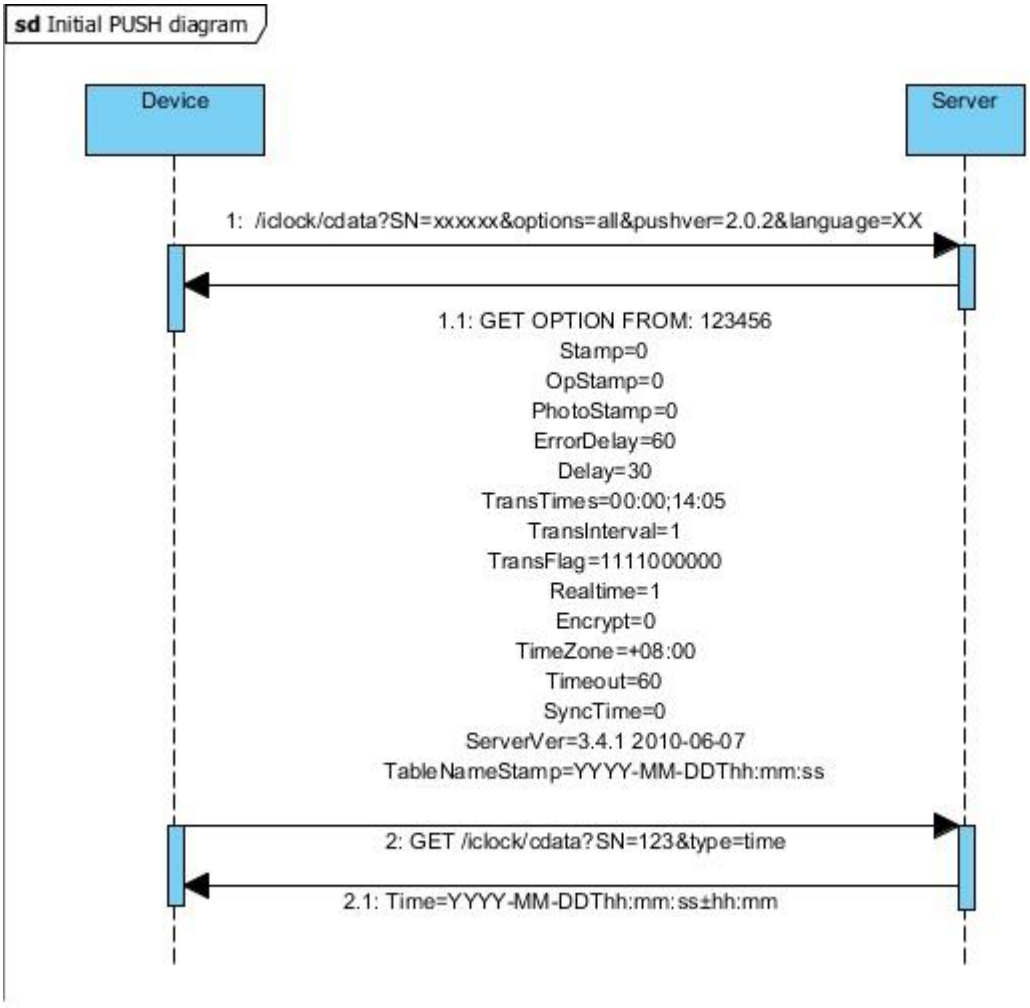
1. PUSH SDK Protocol Sequence Diagram

1.1. Genera HTTP GET request/response sequence diagram

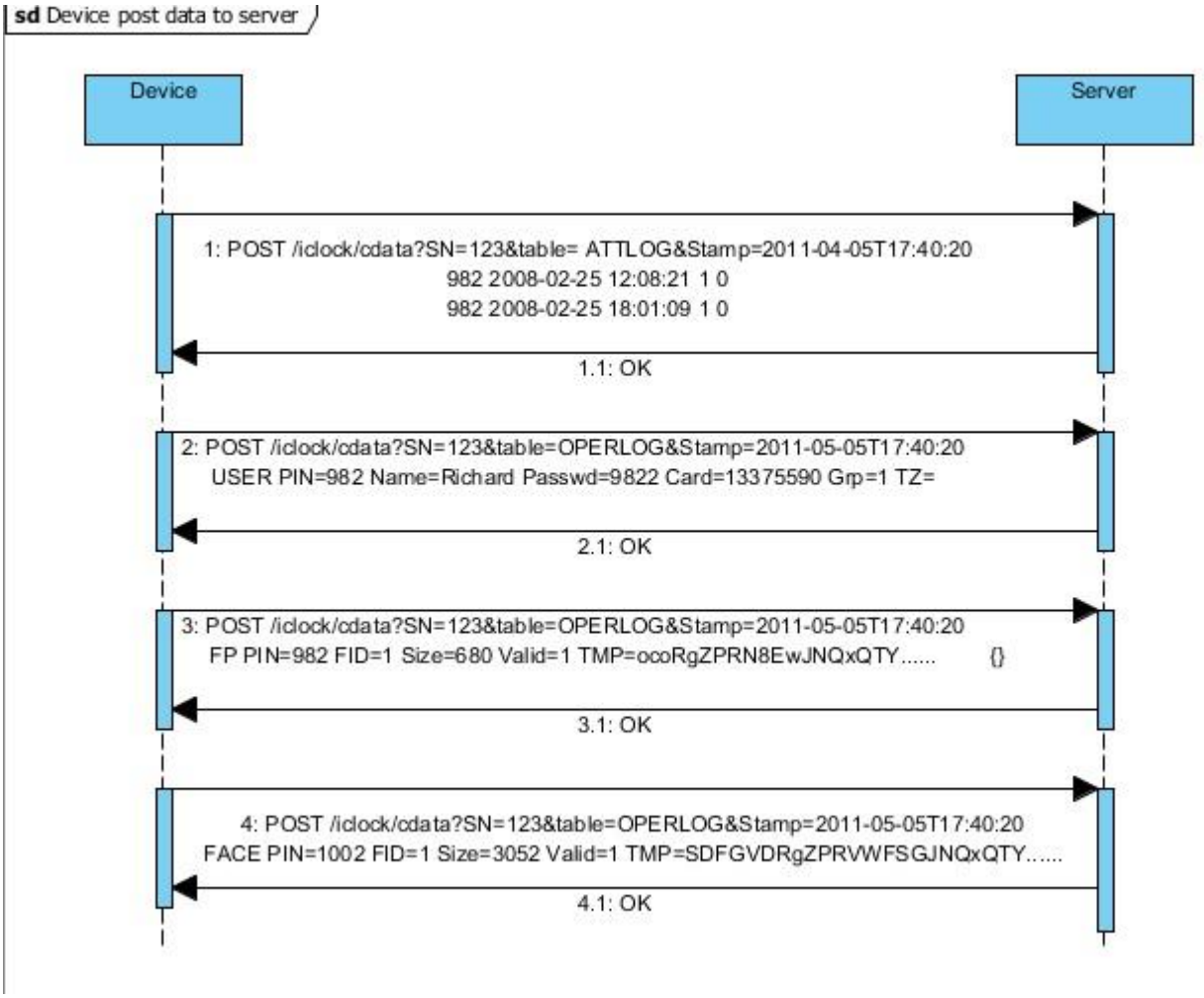


1.2. General HTTP POST request/response sequence diagram

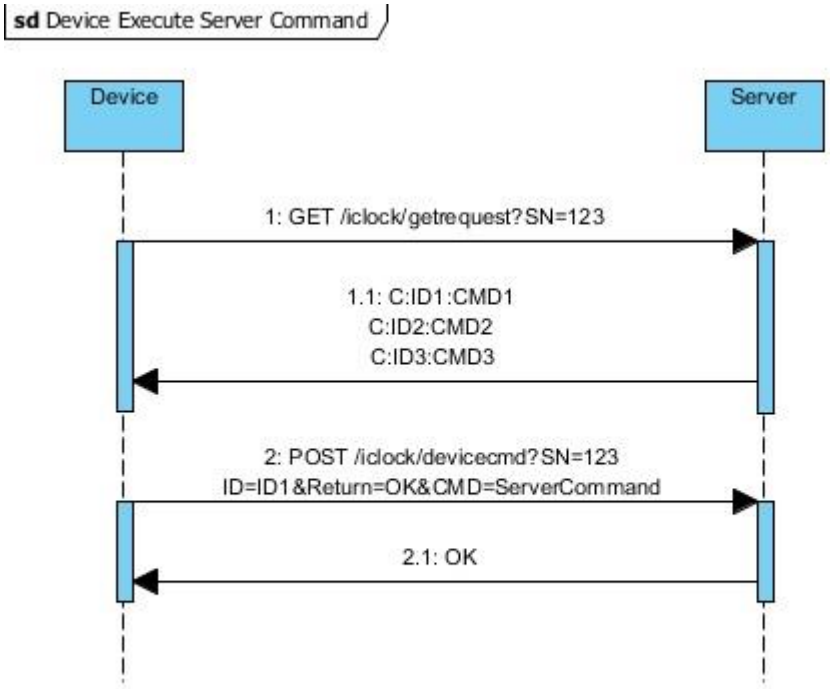
1.3. Device Initial Request to Server and Time Sync Request Diagram



1.4. Data Post From Device to Server Diagram



1.5. Device Execute Server Command General Process Diagram



1.6. Summary of General PUSH API Requests:

1. Initial PUSH API Request:

When the device starts up, the device sends the initial HTTP request to the server with clock serial number and other information which asks server to provide PUSH initial option settings in response to device request. Upon receiving the initial options, device caches the value in clock memory for future process.

2. Real-time Data Post to Server:

When new employee registration data or the attendance record created on clock, the data will be posted to the server in real-time if PUSH initial option set to real time mode.

3. Heartbeat Request:

PUSH SDK provides the near real time monitoring function whereby device sends consecutive heartbeat request to server in configurable intervals. In fixed time interval, device sends a heartbeat request to the server, reporting device online status and retrieves pending command message from server command queue if there is in response.

4. Execute Server Command

With heartbeat request, device is able to retrieve the command from server and execute locally inside clock, upon completing the command, the device updates the server the result by using command result request.

2. PUSH SDK DATA Transfer Protocols

PUSH SDK is a file transfer protocol we developed on HTTP protocol, which is implemented through the device direct access to the server. Main operating environment: reliable Internet connection with TCP/IP protocol, such as local network and the World Wide Web.

Advantages:

1. Automatically upload the new/updated data.
2. Take benefit of HTTP Protocols.
3. Rapid development for third party integration with clock.

Disadvantages: Only TCP/IP protocol is supported.

Note: The device must have PUSH SDK package installed and enabled.

Target Audients： WEB Development Engineer

2.1. Data Transfer between the Device and the Server

The data transfer between the device and the server follows HTTP protocol. The device GET/POST request to the server; and then the results are sent back to the device by the server. Except for some specific data, all the data is transferred in plain text, batch data records are separated by new line feed ("\n") in multiple lines. Each message is using FieldName=Value paired formatted string, the content of data fields varies from different device. For example, the user information on the fingerprint device does not include the Card information, whereas that on the RF device does, as illustrated below:

- User information on the fingerprint device(not include RF Card):

  USER PIN=982 Name=Richard Passwd=9822 Grp=1 TZ=

- User information on the RF fingerprint device with RF Card included:

  USER PIN=982 Name=Richard Passwd=9822 Card=13375590 Grp=1 TZ=

The device determine communication status base on the standard HTTP status code from HTTP response header. The successful communication is determined by HTTP status code equals 20, illustrated as below:

```
HTTP/1.0 200 OK
HTTP/1.1 200 OK
HTTP/1.2 200 OK
```

As for the number and string format, we use C language placeholder, starting with "%", as illustrated below:

- %d - Signed decimal integer
- %s - String
- %x, %X - Unsigned hexadecimal integer

You can simply insert a number between "%" and "d", "s", "x" or "X", to specify the field width. For example, "%3d" is printed as a 3 digit decimal integer, preceded by spaces if it has less than 3 digits; "%8s" is printed as an 8-chracter long string, preceded by spaces if it has less than 8 characters. If the specified field width is less than the actual string or number digital length, they are printed according to their actual length. In addition, if you want the preceded spaces to be filled with "0", a "0" should be inserted before the field width, e.g. "%04d" represents a 4-digit output preceded by "0"s, if its length is less than 4. The default fillings are always placed in the front; if you want to fill in the back, a negative field width should be applied.

2.2. Initialize Device PUSH Function from the Server

Before the communication between device and server starts, the device should be initialized with the configuration settings from the server, the device is to send the following initial request to the server which is the very first request submitted from device to server:

Initial GET Request from device to server to load PUSH initial options:

```
GET /iclock/cdata?SN=%s&options=all&pushver=%s&language=%d
```

SN=%s is the device serial number;

pushver=%s is the PUSH SDK protocol version number, which is not supported by the older versions;

language=%d is the language ID tag which is provided by device. Please refer to the language ID tag table below.

Language ID tag:

ID tag	Language
83	Chinese
69	English
…	…

The server must response the following PUSH options to device, **this is the most important step, without responded PUSH options data, clock won't work properly**:

```
GET OPTION FROM: [clock SN]
ErrorDelay=60
Delay=30
TransTimes=0
TransInterval=0
TransFlag=AttLog\tOpLog\tAttPhoto\tEnrollUser\tChgUser\tEnrollFP\tChgFP
Realtime=1
Encrypt=0
TimeZone=-05:00
Timeout=60
SyncTime=3600
ServerVer=[PUSH server version]
[TableName]Stamp=[YYYY-MM-DDThh:mm:ss]
```

- "GET OPTION FROM" is followed by the device serial number.
- "ErrorDelay" is the time interval (in seconds) between the server connection failure and the reconnection.
- "Delay" is the time interval (in seconds) between each repeated server access under normal network connection.
- "TransTimes" is the exact time spot (hh:mm, 24-hour format) when checking and sending the new data, with multiple time spots separated by semicolons, 10 time spots as the upper limit.
- "TransInterval" is the time interval (in minutes) between each checking and sending the new data.
- "TransFlag" is the ID tag for transferring data from the device to the server, with the maximum length of 90 characters.

The tags takes format like “AttLog\tOpLog\tAttPhoto\tEnrollUser\tChgUser\tEnrollFP\tChgFP” for supported data object auto upload to server. Please refer to Table 2-1 for the supported data object, where the “Operation Log” automatic upload has to be set if the “User” and the “Fingerprint” automatic uploading is set.

Allowed Transferring Data Object	
String Tag	**Description**
AttLog	Attendance Records
OpLog	Operation Log
AttPhoto	Attendance Pictures
EnrollUser	Enroll in New User
ChgUser	Change User Information
EnrollFP	Enroll in New Fingerprint
ChgFP	Change Fingerprint
FPImag	Fingerprint Images
EnrollFACE	Enroll in New Face
ChgFACE	Change Face
FACE	**Enroll/Change Face in FA1

Table 2-1

Note:

It can accept binary format option value such as:

TransFlag=1111111000

This format comes from very early version of PUSH, but it is well kept for backward compatible.

Each position stands for one data object, 1=enable, 0=disable

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
AttLog	OpLog	AttPhoto	EnrollUser	ChgUser	EnrollFP	ChgFP	FPImg	EnrollFACE	ChgFACE

- “Realtime” indicate whether the transfer is in real time or not.
  1 – Real time transfer mode. 0 – Transfer time specified by “TransTimes” and “TransInterval”.
  We take real-time mode as default.
- “Encrypt” represents whether the data is encrypted or not. If it is enabled, encryption algorithm is applied.
  Default to 0: non-encrypt mode.
- “TimeZone” is the time zone where clock is operated, following hh:mm format, for example:
  East time zone is in GMT(-05:00), so it is set as: TimeZone=-05:00.
- “Timeout” stands for HTTP connection timeout (in seconds), default to 60 seconds.
- “SyncTime” is the clock time sync interval (in seconds) with server. If set to 0, stop time sync process.
- “ServerVer” is the server version number and time.
- “[TableNameStamp]” is the time stamp from date/time field of last uploaded record. “TableName” is consistent with the Table Names supported by the device. All the time stamps of data uploading are returned to the device in the following format, with one Table Name on one line:
  [TableName]Stamp=[YYYY-MM-DDThh:mm:ss]

  For example:

ATTLOGStamp=2011-04-05T17:40:20

Is the time stamp of the last loaded attendance record from clock.

OPERLOGStamp=2011-04-05T17:40:20

Is the time stamp of the last loaded operation record from clock.

You can and the necessary time stamp of the last loaded data object as you need. Please refer to Table 2-2 for the Table Names supported by the standard devices. Table Names must be capitalized.

Table Name	Description	Automatic Upload
ATTLOG	Attendance Records	Yes
OPERLOG	Operation Log	Yes, also includes some operation data. When the operation data logs the update of user info, fingerprint/face template, the data object will be auto uploaded into server along with operation log.
ATTPHOTO	Attendance Pictures	Yes
SMS	Short Message Service	No
USER_SMS	User's Short Message	No
USERINFO	User Information	Yes
FINGERTMP	Fingerprint Template	Yes
FACETMP	Face Template	Yes
USERPIC	Users' Pictures	No

Table 2-2

The relationship of "TransTimes", "TransInterval" and "Realtime":

- If "Realtime" is set to 1, data is uploaded immediately, no matter what the other two settings are.
- If "TransInterval" is set to be greater than 0, data is automatically uploaded at each time interval, no matter what the other two settings are.
- If "TransTimes" is set at specific time, this option can be overridden by one of the other two settings.

Note:

Stamp=0 OpStamp=0 PhotoStamp=0

The above options are only for PUSH 1.0 and now removed from the PUSH INIT options in 2.0.3.

2.3. Sync Device Time with Server

The device sends the following command to the server to obtain the current time.

GET /iclock/cdata?SN=%s&type=time

The server returns the following information:

Time=YYYY-MM-DDThh:mm:ss±hh:mm

For example, if the server time is at EAST time 17:40:20 May 5th, 2011, the response is "Time=2011-05-05T17:40:20-05:00".

2.4. General Data Transactions

2.4.1. Upload the Attendance Records(TFT series)

POST Request of attendance data submitted to server:

POST /iclock/cdata?SN=%s&table= ATTLOG&Stamp=%s

[PIN]\t[Punch date/time]\t[Attendance State]\t[Verify Mode]\t[Workcode]\t[Reserved 1]\t[Reserved 2]

Example:

POST /iclock/cdata?SN=12300001&table= ATTLOG&Stamp=2008-02-25T12:08:21

982 2008-02-25 12:08:21 1 0 9001

982 2008-02-25 18:01:09 1 0 9001

**2.4.2. Upload the Attendance Records(TFT series) POST

Request of attendance data submitted to server:

POST /iclock/cdata?SN=%s&table= ATTLOG&Stamp=%s

[PIN]\t[Punch date/time]\t[Attendance State]\t[Verify Mode]\t[WorkCode1]\t[Reserved 1]\t[Reserved 2]\t

[JobCode1]\t[JobCode2]\t[JobCode3]\t[JobCode4]\t[JobCode5]\t[JobCode6]\t[TipCode1]\t[TipCode2]\t[TipCode3]

Example:

POST /iclock/cdata?SN=12300001&table= ATTLOG &Stamp=2008-02-25T12:08:21

982 2016-02-25 12:08:21 1 0 9001 0 0 1 2 3 4 5 6 1.0 2.0 3.0

982 2016-02-25 13:18:22 1 0 9001 0 0 1 2 3 4 5 6 1.0 2.0 3.0

Where:

- SN=%s: Device serial number;
- Table=ATTLOG: Reserved Keyword for Attendance Record (see Table 2-2) from the device;

- Stamp=%s: The timestamp when the request is submitted in format: yyyy-mm-ddThh24:mm:ss is, which will be saved in ATTLOGStamp (see Table 2-2) by the server, as a return value for PUSH initial option value.

According to the device configuration, a record is composed of many fields, separated by "\t" (horizon TAB). These fields are listed below:

- PIN – Employee/User ID which is the employee/user ID from 3rd party system
- Punch date/time - Attendance time, format: yyyy-mm-dd hh24:mm:ss.
- Attendance State – Attendance state, such as check in, check out… (see Table 2-3)
- Verify Mode – User identification method (see Table 2-3)
- WORKCODE – Used for work code, Job code or department transfer.
- RESERVED1 – Reserved field 1, can be left empty;
- RESERVED2 – Reserved field 2, can be left empty;

Some devices do not have the last three fields.

Verify Mode	Configured Value	Default Mapped Value in AttLog
Finger | Badge | Password	128	n/a
Finger	129	1
PIN	130	3
Password	131	0
Badge	132	2
Finger | Password	133	n/a
Finger | Badge	134	n/a
Badge | Password	135	n/a
PIN & Fingerprint	136	4
Finger & Password	137	5
Badge & Finger	138	6
Badge & Password	139	7
Badge & Password & Finger	140	8
PIN & Password & Finger	141	9
PIN Finger | Badge Finger	142	n/a
PIN | Badge	143	n/a
Non-validation PIN	*	16**
Non-validation Badge	*	17**
Non-val PIN | Non-val Badge	*	n/a
Face	148	n/a
Face & Card	149	n/a
Face & Pin	150	n/a

Table 2.3. Verifying mode table

Notes:

- **The multiple verifying mode is available from firmware ver 6.8.0(build 545) and above.
- To enable the multiple verifying mode, device option should be set as:

  ~UserExtFmt=1

~LockFunOn=0

- N/A - no mapped value for multiple verifying modes option;
- * - None validation mode is set in device options file, not set in user data;

  The option name in device is:

  PINOrCardNonVal, value = 0:Disable, 1-PIN mode, 2-Badge mode, 3-both
- ** - It is fixed value in attendance log, should not be mapped to other value.
- The mapping value is set in device option:

  VerifyValueMap=129:1,130:3,131:0,132:2,136:4,137:5,138:6,139:7,140:8,141:9

Attendance State Name	Default Value	Default Mapped value For DCS
Check In	0	0
Check Out	1	1
Meal In/Meal End	2	2
Meal Out/Meal Start	3	3
Break In/Break End	4	4
Break Out/Break Start	5	5
Overtime In		
Overtime Out		

Table 2.4. Attendance State

Notes:

- Clock supports up to 6 Attendance State which can be defined to any Attendance State and set to any. Attendance name with mapped values from either clock admin menu or PUSH API command.

Response from Server:

OK

If the server responds error information, e.g. HTTP 404/500, or there is no return value for a while (timeout), the device will treat it as failure and will resend the data.

2.4.3. Upload the User Information and the System Log

POST Request of OPERLOG data submitted from device to server:

Upload enrolled user info including badge, password to server: *POST /iclock/cdata?SN=%s&table=OPERLOG&Stamp=%s* *USER PIN=%s\tName=%s\tPasswd=%d\tCard=%d\tGrp=%d\tTZ=%s* Example:

POST /iclock/cdata?SN=%s&table=OPERLOG&Stamp=%s

USER PIN=982 Name=Richard Passwd=9822 Card=13375590 Grp=1 TZ=

Upload enrolled fingerprint to server

POST /iclock/cdata?SN=%s&table=OPERLOG&Stamp=%s

FP PIN=%s\tFID=%d\tSize=%d\tValid=%d\tTMP=%s

Example:

POST /iclock/cdata?SN=%s&table=OPERLOG&Stamp=%s

FP PIN=982 FID=1 Size=680 Valid=1 TMP=ocoRgZPRN8EwJNQxQTY.....

Upload enrolled face template to server

POST /iclock/cdata?SN=%s&table=OPERLOG&Stamp=%s

FACE PIN=%s\tSize=%d\tValid=%d\tTMP=%s

Example:

POST /iclock/cdata?SN=%s&table=OPERLOG&Stamp=%s

FACE PIN=1002 Size=3052 Valid=1 TMP=SDFGVDRgZPRVWFSGJNQxQTY.....

Where:

- SN=%s is the device serial number;
- *table*=OPERLOG is the keyword for Operation Log Table Name (see Table 2-2) from the device;
- Stamp=%s with format yyyy-mm-ddThh24:mm:ss is the timestamp for the data uploading, which will be saved in OPERLOGStamp (see Table 2-2) by the server, as a return value when requested by the GET command.

Each record begins with a starting tag indicating the content of the record, followed by a space and all the field values of the record, and finally closed by the newline character "\n". Fields are all in “FieldName=Value” form, separated by TAB char “\t”. Please refer to Table 2-4 and Table 2-5 for the detailed description.

Note: the data field name is case sensitive, please follow the exact case.

Starting Tag	Record Content	Field Description
USER	User information	PIN=982: User’s attendance number Name=Richard: User name Passwd=9822: Password Card=13375590: Badge ID card number * Grp=1: user group, default to 1; TZ= : user timezone for access control, default to null;
FP	User’s fingerprint template	PIN=982: User’s attendance number FID=1: User’s fingerprint template ID number Size=680: BASE64 encoded string length of the fingerprint template

		Valid=1: Validation of the fingerprint template TMP=... ...: BASE64 encoded fingerprint template
FACE	User's face template	PIN=1002: User's attendance number FID=1: User's face template ID number Size=3052: BASE64 encoded string length of the face template Valid=1: Validation of the face template TMP=... ...: BASE64 encoded face template
OPLOG	Administrator operation records	Operation code Administrator ID Time Operation Object 1 ** Operation Object 2 Operation Object 3 Operation Object 4

Table 2.5 Operlog table

Description:

* ID card number format: If enclosed in "[ ]", it represents a complete HEX card number; Otherwise, it is the same as the number shown during scan.

** The 4 "Operation Objects" within the operation records represent different parameters, or different objects. Please refer to Table 2-5 for operation parameter description.

Operation Code	Operation Description	Operation Parameter Description
0	Power on	
1	Power off	
2	Authentication failed	If the user does a 1：1 authentication check, "Operation Object 1" represents the user's PIN number.
3	Alarm	"Operation Object 1" represents the specific reasons, the possible values are: 50: Door Close Detected 51: Door Open Detected 55: Machine Been Broken 53: Out Door Button 54: Door Broken Accidentally 58: Try Invalid Verification 65535: Alarm Canceled
4	Enter the menu	
5	Change settings	"Operation Object 1" represents the modified item number. "Operation Object 2" represents the new value.
6	Register fingerprint	"Operation Object 1" represents the user's ID number. "Operation Object 2" represents the fingerprint template number. "Operation Object 3" represents the string length of the fingerprint template.

7	Register password	
8	Register HID card	
9	Delete user	“Operation Object 1” represents the user’s ID number.
10	Delete fingerprint	“Operation Object 1” represents the user’s ID number.
11	Delete password	“Operation Object 1” represents the user’s ID number.
12	Delete RF card	“Operation Object 1” represents the user’s ID number.
13	Clear data	
14	Create MF card	
15	Register MF card	
16	Sign up MF card	
17	Delete MF card registration	
18	Clear MF card	
19	Move the registration data to the card	
20	Copy the card data to the device	
21	Set time	
22	Factory settings	
23	Delete access records	
24	Cancel the administrator’s privilege	
25	Change group access settings	
26	Change user access settings	
27	Change access time	
28	Change the door-open settings	
29	Door open	
30	Enroll in new users	
31	Change the fingerprint properties	
32	Forced alarm	
37	Change existing user info	

Table 2.6 operation parameter table

Response from Server:

OK

2.4.4. Upload the Attendance Captured Pictures

If the device is equipped with a camera and has the on-site picture capturing and uploading function, the device will capture the on-site pictures and upload them to the server during the user authentication process.

POST Request of ATTPHOTO data from device to server:

POST /iclock/cdata?SN=%s&table=ATTPHOTO&Stamp=%s

PIN=%d *SN=%s* *size=%d* *CMD=type\0[BINARY IMAGE DATA]*

Where:

- SN=%s is for the device serial number.
- table=ATTPHOTO is the keyword for the Attendance Picture Table Name (see Table 2-2) of the device.
- Stamp=%s with format yyyy-mm-ddThh24:mm:ss is the time stamp of the picture, which will be saved in ATTPHOTOStamp (see Table 2-2) by the server, as a return value when requested by the GET command.
- PIN=%d is the attendance picture ID, and it is defined as following accordingly:
  - "PIN=DATETIME-U"，where DATETIME is the authentication time with a YYYYMMDDHHNNSS format; U is the user's attendance number. This format shows the user has passed the authentication check.
  - "PIN=DATETIME" shows the user did not pass the authentication check.
- size=%d defines the picture size.
- CMD=type represents the image type:
  - CMD=uploadphoto stands for the picture uploaded from the device;
  - CMD=realupload stands for the picture is uploaded in real-time.
- "\0" is the escape sequence used in C language.
- [BINARY IMAGE DATA] is the binary image data of the on-site captured JPG formatted picture.

Response from server:

OK

2.4.5. Execute the Commands from Server

For HTTP communication protocol, as HTTP client, device is always the side to initiate the talk, server is never able to initiate the talk with client. So if there is no direct way to have device perform the command from server side, we provides heartbeat request solution to achieve this: the device keeps sending the special request at regular intervals (usually 10~60 seconds), then server is able to send the command to device with the response message as below:

Heartbeat request from Device:

GET /iclock/getrequest?SN=%s

Where SN=%s is the device serial number.

Response message with commands returned from server:

C:[ID]:[Command] *C:[ID]:[Command]*

…

- The server can return multiple commands to the device in the response, each command is put in one single line ended by carriage return “\n”;
- Each command begins with “C:” which is hardcoded.
- [ID] type is %d, the command ID to identify each command.
- [Command] type is %s, the actual command body executed by clock.

Please refer to the following chapters for all the available commands and command format.

Note, we suggest no more than 200 commands or less 40KB in one message due to device limitation; if using GPRS mode, the size should be less than 10Kbyte.

When the device completes executing all the commands, it will send out separated request to update the server the execution results.

POST /iclock/devicecmd?SN=%s
ID=%d&Return=%d&CMD=%s

The request uses POST method to submit the execution results and where:

- SN=%s is the device serial number;
- ID=%d represents the corresponding command ID number;
- Return=%d represents the command result code;
  0: the command executed without error;
  Other: Failure. Please refer to the command description for more details.
- CMD=%s is the return data.

- 0 – Success
- -1 – Parameter error, Database Not initialized;
- -2 – Wrong operation;
- -3 – Access error, Flash I/O Error;
- -4 – No matched data found;
- -5 – No more space;
- -6 – Data is not valid;
- -9 – The size of the fingerprint template does not match the specified “Size”.
- -10 – The user specified by “PIN” does not exist.
- -11 – Invalid fingerprint template format
- -12 – Invalid fingerprint template
- -22 – Unknown command

Table 2.7 Result Code

Upon receiving the above requests from the device, the server will treat the corresponding commands as having been executed.

Note:

- After the clock for the initial GET request to the server, it will send the following request next the to the initial request :

```
GET http://host/iclock/getrequest?SN=%s&INFO= Ver%206.8.0(build640), 2, 0, 0, 192.168.1.201, 10, -1, -12, 101
```

The INFO parameter:

The following value is separated by ‘,’, and each field stands for specific information of the device:

INFO=[firmware version],[number of enrolled users], [number of fingerprints], [number of attendance records], [device IP],[Fingerprint version], [Face version], [Face templates count],[dev support data]

[dev support data] : it is a string made of 1 and 0, stands for on/off of the device supported features:

Position 1	**Position 2**	**Position 3**
Fingerprint	Face	User Picture

PART VI: PUSH Commands from Server

1. Operation System Command

Command:

SHELL CMD_String

Function:

Run the operation system command.

Execution Result:

On device finishes command process, it sends devicecmd request to server reporting the result:

- If the command is executed without error, device sends devicecmd request to server reporting the result::

```
POST /iclock/devicecmd?SN=%s
ID=%d
SN=%s
Return=%s
CMD=Shell
FILENAME=shellout.txt
Content=%s
```

Where:

Return=%d is the return code (see Table 2.7);

Content=%s is the output of the command execution.

- If device finishes command process with error, it sends /devicecmd request to server reporting the result:

```
POST /iclock/devicecmd?SN=%s
ID=%d&Return=%d&CMD=Shell.
```

2. Check the Data Update

Command:

CHECK

Function:

This command asks device to check any updated option. Upon receiving the command, device check if any option updated, if so, device uploads the updated option data to server immediately with separated request.

Execution Result:

When device completes command execution, it sends devicecmd request to server reporting the result:

```
POST /iclock/devicecmd?SN=%s
ID=%d&SN=%s&Return=0&CMD=CHECK
```

3. Clear and Reset Data

Command:

CLEAR LOG

Function:

Clear the attendance records from the device

Command:

CLEAR DATA

Function:

Clear all the data from the device

Command:

CLEAR PHOTO

Function:

Clear the on-site captured pictures from the device.

****Command:**

CLEAR ALL USERINFO

Function:

- Clear all User Information from device with all associated templates.
- Replace by another command in **7.6** TFT series.

****Command:**

CLEAR ALL BELL

Function:

- Clear all defined Bell from device.
- Replace by another command in **7.16.2** TFT series.

****Command:**

RESET ALL FUNCKEYS

Function:

- Restore the short key settings to factory default.
- Only available on TFT series.

****Command:**

RESET ALL TIMEZONE

Function:

- Restore all timezone periods to 00:00-23:59 7 days a week.
- Only available on TFT series.

4. Get the Device Information

Command:

INFO

Function:.

It loads device's all option configure from options.cfg file to device with extra system information.

Result:

If the command is executed without error, the device will POST the following data back to the server:

POST /iclock/devicecmd?SN=%s ID=%d&Return=0&CMD=INFO [OPTIONS LIST]

OPTIONS includes the clock configurations in the format of name value pair as "Item=Value", the data comes from all entries of options.cfg and the following extra info:

Item	Description
TransactionCount	The number of attendance records
FPCount	The number of the registered fingerprints
UserCount	The number of the registered users
FWVersion	The device version number
…	…

See the following "Set the Device Option" for more options from options.cfg.

5. Set the Device Option

Command

SET OPTION ITEM=VALUE

Function:

Where ITEM is the device option; and VALUE is the option value. For example:

SET OPTION IPAddress=192.168.1.225

The above command sets the device IP address as 192.168.1.225.

**The currently supported items in TFT series are:

Item	Description
IPAddress	Device IP address
NetMask	Device subnet mask
GATEIPAddress	Device gateway address
VOLUME	Volume
MAC	Ethernet MAC address of the device, its C language format is: "%02X:%02X:%02X:%02X:%02X:%02X"

CardKey	Mifare card encrypted password
DeviceID	Device ID number
LockOn	LockOn duration
AlarmAttLog	Attendance record alarm
AlarmReRec	The shortest time interval between two repeated check-ins
RS232BaudRate	RS232/RS485 baud rate
AutoPowerOff	Automatic power-off time, with the format: Hours×256+Minutes The following time setting items all follow the above formats.
AutoPowerOn	Automatic power-on time
AutoPowerSuspend	Automatic power-suspended time
AutoAlarm1～AutoAlarm50	50 automatic alarm time
IdlePower	Idle setting
IdleMinute	Idle time (in minutes)
RS232On	Rs232 ON or OFF
RS485On	Rs485 ON or OFF
UnlockPerson	The number of unlocking persons
OnlyPINCard	Read-only Mifare ID card
HiSpeedNet	Network speed
Must1To1	Only 1:1 fingerprint authentication is allowed
ODD	Unlock time, overdue will cause alarm on
DSM	
AADelay	
DUHK	Forced alarm help key is ON or OFF
DU11	1:1 fingerprint authentication can cause forced alarm
DU1N	1:N fingerprint authentication can cause forced alarm
DUPWD	Password authentication can cause forced alarm
DUAD	The extended alarm time (in seconds) when the forced alarm is on
LockPWRButton	Power-off button lock
SUN	Whether sending out a message to other devices within the network, when the device starts up.
I1NFrom	Setting 1: The starting user ID number of the N fingerprint authentication mode
I1NTo	Setting 1: The ending user ID number of the N fingerprint authentication mode
I1H	1：H function is ON or OFF
I1G	1：G function is ON or OFF
KeyPadBeep	Key pad beep is ON or OFF
WorkCode	WorkCode function is ON or OFF
AAVOLUME	Alarm volume
DHCP	DHCP function is ON or OFF

EnableProxyServer	HTTP proxy server is ON or OFF
ProxyServerIP	HTTP proxy server IP address
ProxyServerPort	HTTP proxy server port
PrinterOn	Printing mode is ON or OFF
DefaultGroup	Default group number
GroupFpLimit	Number limit of fingerprint in each group
WIFI	WiFi is ON
wifidhcp	DHCP function of WiFi is ON or OFF
AmPmFormatFunOn	AM/PM format is ON or OFF on the main interface
AntiPassbackOn	Anti-passback function is ON or OFF
MasterSlaveOn	Master-slave function is ON or OFF
ImeFunOn	T9 text input method is ON or OFF
WebServerIP	PUSH SDK Web server's IP address
WebServerPort	PUSH SDK Web server port
ApiPort	DataAPI SDK port
DelRecord	The number of automatically deleted old attendance records when the attendance records are full

- IsPushComKey

'1': value of this parameter adds one field in each http communication from client to server, named Comkey=%s. For example,

GET /iclock/cdata?SN=xxxxxx&ComKey=%s&type=time.

Purpose of this functionality is to take care of fake punches to server from unknown device. This also add benefits in terms of security. Default Communication key is 88888888.

'0': value of this parameter disables "Comkey=%s" field in http communication.

- PushComKey

Server can change default Communication key using this parameter.

- PINOrCardNonVal

This parameter configures whether device should work in validation mode or non validation mode.

0 sets device in validation mode. This means enduser has to verify himself using pin or card or biometric or password or "combination of them" to make punch.

1 sets device in "pin non validation mode". Anyone can key in any number to make punch.

2 sets device in "card non validation mode". Anyone can scan any card and make punch.

3 sets device in "card or pin non validation mode". Anyone can scan card or key in any number to punch.

****Note**: Some options are only functional on some particular devices, e.g. "AntiPassbackOn" is only for advanced access control devices, and "WIFI" can only be functional on the devices with built-in WiFi modules. The Non-validation function is currently not available on TFT series.

Result:

POST /iclock/devicecmd?SN=12345 ID=[command ID]&SN=12345&Return=0&CMD=SET OPTION

6. Reboot the Device

Command:

REBOOT

Function:

Reboot the device

Note:

If the server sends multiple commands to the device at one time, including this REBOOT command, it has to be the last one, otherwise, the other subsequent commands will be ignored.

7. Data Sync Commands

7.1. Data Command Format:

PUSH SDK allows server to sync the business data between clock and server.

Command Format:

DATA <SUBCMD>

Where:

<SUBCMD> are the sub-commands shown as follows:

- UPDATE tablename value :Add or delete tablename values
- DELETE tablename key :Delete tablename values according to the key

See Table 2-2 for tablenames.

7.2. Add or Modify User Information

Command:

UPDATE USERINFO PIN=%d\tName=%s\tPri=%d\tPasswd=%s\tCard=%s\tGrp=%d\tTZ=%d\tVerifyMode=%d

Parameters:

This is to sync user basic info from server to device, where:

- PIN：User/Employee ID, required;
- Name：User's name, optional;
- Pri：Privilege (**0 – Normal user, 1-inactive user, 2-Enroller, 14-admin)**, optional;
- Passwd：User's password, optional;
- Card：User's card number, optional;
- **Grp：Group (for access control), optional; Grp = 0 means the user applies user based lockout schedules. Grp= 1~99 means applying group based lockout schedules. Definition of group based lockout schedules refer

to **16**

- **TZ：User's lock out schedules (for access control), optional; TZ is represented by a 16 character string.

  The 1st 4 digits: 0000 means the user applies to user lockout schedule, non 0000(such as 0001) means the user applies to group lockout schedule. The non-zero value will be overlooked once Grp is being assigned with a Group ID.

  The 2nd 4 digits: The ID of the 1st schedule(or "timezone") defined in section 7.15, The ID of 1st schedule is transformed to a HEX representation of a decimal ID..

  The 3rd 4 digits: The ID of the 2nd schedule(or "timezone") defined in section 7.15. The ID of 2nd schedule is transformed to a HEX representation of a decimal ID.

  The 4th 4 digits: The ID of the 3rd schedule(or "timezone") defined in section 7.15. The ID of 3rd schedule is transformed to a HEX representation of a decimal ID.

  E.g TZ=000000010002000F represents the user applies to user based lockout schedules ID 1, 2,15 as defined using API in section 7.15

- VerifyMode: User verifying mode code, see Part III Table 2.3. Verifying mode table, optional.

Result:

Device post another request to server with result code defined below:

```
POST /iclock/devicecmd?SN=%s
ID=[command ID]&SN=%s&Return=[result code]&CMD=DATA
```

**7.3. Add or Modify User Extended Information

Command:

```
UPDATE USEREXTINFO PIN=%d\tRecheckMin=%d\tLockoutOverride=%dUser1to1FpThreshold=%d
```

Parameters:

This function to sync extended user info from server to device, where:

- PIN：User/Employee ID, required;
- RecheckMin：Prevent a duplicated entry/punch for the USER in seconds
- LockoutOverrider: 1 or 0. Enable or disable the overrider of lockout schedules.
- User1to1FpThreshold: Configure fingerprint verification threshold per user, optional. Value candidates are 10,15,20,25,30,35.

  Option configuration USAUser1to1FPThresholdFunOn=1 is required.

Result:

Device post another request to server with result code defined below:

```
POST /iclock/devicecmd?SN=%s
ID=[command ID]&SN=%s&Return=[result code]&CMD=DATA
```

7.4. Add or Modify User's Fingerprint

Command:

UPDATE FINGERTMP PIN=%d\tFID=%d\tSize=%d\tValid=%d\tTMP=%s *(The old version command: FP PIN=%d\tFID=%d\tSize=%d\tValid=%d\tTMP=%s)*

Parameters:

- UPDATE: Reserved keyword for data operation, add or modify the fingerprint template;
- FINGERTMP: Reserved keyword for fingerprint data field;
- PIN: User ID, required;
- FID: User's fingerprint template ID, required;
- Size: BASE64 encoded string length of the fingerprint template （Not available in the old versions）
- Valid: Flag for valid fingerprint, 1 – Valid, 0 – Invalid. Default to 1.
- TMP: BASE64 encoded fingerprint template

Result:

POST /iclock/devicecmd?SN=12345 ID=[command ID]&SN=12345&Return=[result code]&CMD=DATA

**7.5. Add or Modify User's Face Template

Command:

UPDATE FACE PIN=%d\tFID=%d\tSize=%d\tValid=%d\tTMP=%s

Parameters:

- UPDATE: Reserved keyword for data operation, add or modify the face template;
- FACE: Reserved keyword for face data object;
- PIN: User ID, required;
- FID: The face template;
- Size: BASE64 encoded string length of the face template;
- Valid: Flag for valid fingerprint, 1 – Valid, 0 – Invalid. Default to 1.
- TMP: BASE64 encoded face template

Result:

POST /iclock/devicecmd?SN=12345 ID=[command ID]&SN=12345&Return=[result code]&CMD=DATA

** 7.6. Delete User

Command:

DELETE USERINFO PIN=%d	:delete single user
DETETE USERINFO	:delete all users

Parameters:

- PIN: User identification Number

****Note:**

Once the user is removed from the device, all associated bio-templates including fingerprint template, face templates, extended user info should be removed as well from device altogether;

Result:

POST /iclock/devicecmd?SN=12345
ID=[command ID]&SN=12345&Return=[result code]&CMD=DATA

7.7. Delete User's Fingerprint

Command:

DELETE FINGERTMP PIN=%d\tFID=%d	: delete single template
DELETE FINGERTMP PIN=%d	: delete all template of one user;
DELETE FINGERTMP	: delete all template from device;

Parameters:

- PIN =user ID, optional; if PIN is absent from command, delete all templates from device;
- FID = fingerprint index, if FID is absent from command, delete all templates fo the user;

Result:

POST /iclock/devicecmd?SN=12345
ID=[command ID]&SN=12345&Return=[result code]&CMD=DATA

7.8. Delete User's Face Template

Command:

DELETE FACE PIN=%d	: delete single user face template

Parameters:

- PIN =user ID, optional; if PIN is absent from command, delete all templates from device;

Result:

POST /iclock/devicecmd?SN=12345 ID=[command ID]&SN=12345&Return=[result code]&CMD=DATA

7.9. Send User's Picture to Device

Command:

UPDATE USERPIC PIN=%d\tSize=%d\t Content=%s

Parameter:

- PIN: User ID, required.
- Size: BASE64 encoded string length of the image file
- **Content: BASE64 encoded image data limited by maximum size of 10K bytes for ZK TFT 20 series and 20K bytes for ZK TFT 30 series.

Result:

POST /iclock/devicecmd?SN=12345 ID=[command ID]&SN=12345&Return=[result code]&CMD=DATA

Where:

- 0 – Success
- -1 – Operation is not supported
- -2 – Download error (The image size may exceed the size limit.)
- -3 – Parameter error

7.10. Delete User's Picture

Command:

DELETE USERPIC PIN=%d

Parameter:

- PIN: User ID, required;

Result:

POST /iclock/devicecmd?SN=12345 ID=[command ID]&SN=12345&Return=[result code]&CMD=DATA

7.11. Manage Short Messages

7.11.1 Add/Update SMS.

Command:

UPDATE SMS MSG=%s\tTAG=%d\tUID=%d\tMIN=%d\tStartTime=%s

Parameters:

- MSG: SMS message
- TAG:
  - 253 represent Public SMS:-All enrolled users can see this message after successful verification.
  - 254 represent Private SMS:-Only assigned user can see this message after successful verification.
- UID: Short Message ID.
- MIN: The effective time span of the short message (in minutes).These many minutes, messages are displayed after successful verification.
- StartTime: Starting time of the short message in YYYY-MM-DD HH:MM:SS format.

Result:

POST /iclock/devicecmd?SN=12345 ID=[command ID]&SN=12345&Return=[result value]&CMD=DATA

7.11.2 Assign SMS to users.

Command:

UPDATE USER_SMS PIN=%d\tUID=%d

Description:

Assign the message to particular user.

- PIN: User ID;
- UID: message ID;.

Result:

POST /iclock/devicecmd?SN=12345 ID=[command ID]&SN=12345&Return=[result code]&CMD=DATA

****7.11.3. Delete SMS (Short Messages)**

Command:

DATA DELETE SMS *DATA DELETE SMS UID=%d*

Parameters:

- UID: Short Message ID.

Description:

- Remove all SMS including public and private SMSs from the device
- Remove a specific SMS using UID: message ID.

Result:

POST /iclock/devicecmd?SN=12345 ID=[command ID]&SN=12345&Return=[result code]&CMD= DATA

7.12. Work Code

7.12.1. Add/update Work Code

Command:

DATA UPDATE WORKCODE CODE=%d\tName=%s

Parameter:

- CODE: - Work Code
- Name: - Work Code Name

Description:

To add or update work code record from host application.

Result:

POST /iclock/devicecmd?SN=12345 ID=[command ID]&SN=12345&Return=[result code]&CMD= DATA

7.12.2. Delete Work Code

Command:

DATA DELETE WORKCODE CODE=%d *DATA DELETE WORKCODE*

Parameter:

- CODE=%d: It is optional, if absent, device will delete all work codes.

Description:

To delete specified or all work code records.

Result:

POST /iclock/devicecmd?SN=12345 ID=[command ID]&SN=12345&Return=[result code]&CMD= DATA

**7.13. Job Code (Enhanced Work Code)

**7.13.1 Add/Update JOBCODEGROUP

Command:

DATA UPDATE JOBCODEGROUP ID=%d\tName=%s\tEnable=%d

Parameter:

- ID: JOBCODEGROUP ID starts from 1 to 6, a JOBCODE has to be associates with a specific JOBCODEGROUP, see **7.13.2** for details.
- Name: JobCode Group Name
- Enable : 1 or 0 to enable or disable one JOBCODEGROUP

Description:

To add or update JOBCODEGROUP for creating of jobcodes associated to the JOBCODEGROUP.

Result:

POST /iclock/devicecmd?SN=12345 ID=[command ID]&SN=12345&Return=[result code]&CMD= DATA

**7.13.2 Add/Update JOBCODE

Command:

DATA UPDATE JOBCODE Number=%d\tName=%s\tJobCodeGroupId%d

Parameter:

- Number: JOBCODE number.
- Name: JOBCODE name
- JobCodeGroupId: ID of a JOBCODEGROUP which JOBCODE is assigned to. The ID ranges from 1 to 6.

Description:

To add or update JOBCODE in a specified and pre-defined JOBCODEGROUP.

Result:

POST /iclock/devicecmd?SN=12345 ID=[command ID]&SN=12345&Return=[result code]&CMD= DATA

**7.13.3. Delete JOBCODEGROUP

Command:

DATA DELETE JOBCODEGROUP ID=%d

Parameter:

- ID: JOBCODEGROUP ID

Description:

To undefine a JOBCODEGROUP.

Result:

POST /iclock/devicecmd?SN=12345 ID=[command ID]&SN=12345&Return=[result code]&CMD= DATA

**7.13.4. Delete JOBCODE in a JOBCODEGROUP

Command:

DATA DELETE JOBCODE Number=%d\tJobCodeGroupId =%d

Parameter:

- *Number*: JOBCODE number.
- *JobCodeGroupId*: The ID ranges from 1 to 6.

Description:

To delete a specific JOBCODE *in a specific* JOBCODEGROUP

Result:

POST /iclock/devicecmd?SN=12345 ID=[command ID]&SN=12345&Return=[result code]&CMD= DATA

**7.14 Tip Code(Tip Entry)

7.14.1 Add/Update TIPCODE

Command:

DATA UPDATE TIPCODE ID=%d\tName=%s\tEnable=%d

Parameter:

- ID: TIPCODE ID(range 1 to 3)
- Name: the description of a tip entry
- Enable: 1 or 0 to enable and disable the TIPCODE.

Description:

- To activate or deactivate a TIPCODE with range from ID 1 to ID 3
- To define the name of this TIPCODE

Result:

POST /iclock/devicecmd?SN=12345 ID=[command ID]&SN=12345&Return=[result code]&CMD= DATA

**7.15 User TimeZone/Lockout Schedules

7.15.1 Add/Update Lockout Schedule(Timezone)

Command:

DATA UPDATE TIMEZONE TZID=%d\tTIME=%s

Parameter:

- TZID: Lockout schedule ID ranges from 1 to 50.
- TIME: string defined for time zone covering from Sunday to Saturday, format as:*[Start_HHMM][End_HHMM][Start_HHMM][End_HHMM]...*The Start/End time repeats 7 times for 7 days.

E.g:

DATA UPDATE TIMEZONE TZID=1 TIME=1123123300001533000012330000123300001233000012330000123300001233

The lockout schedule for ID 1 is

Sunday 11:23-12:33; Monday 00:00-15:33; Tuesday 00:00 -12:33; Wednesday 00:00-12:33; Tuesday 00:00 – 12:33; Friday 01:00-12:33; Saturday 00:00- 12:33

Description:

To configure the lockout schedules.

Result:

POST /iclock/devicecmd?SN=12345 ID=[command ID]&SN=12345&Return=[result code]&CMD= DATA

7.15.2 Delete Lockout Schedule(Timezone)

Command:

DATA DELETE TIMEZONE	Reset single time zone by TZID
DATA DELETE TIMEZONE TZID=%d	Reset all time zone

Parameter:

- TZID: Lockout schedule ID ranges from 1 to 50.

Description:

To reset schedules to 00:00-23:59 in 7 days a week.

7.15.3 Update User Lockout Schedule.

See details to 7.2

7.15.4 Update Group Lockout Schedule.

Command:

DATA UPDATE AccGroup ID=%d Verify=%d Validholiday=0 TZ=%d

Parameters:

- ID: Group number
- Verify: Verifying mode, details in table 2.3.
- Validholiday : Holiday function is still under development, use 0 for any input.
- TZ:
  - The 1st 4 digits: The ID of the 1st schedule(or “timezone”) defined in section 7.15, The ID of 1st schedule is transformed to a HEX representation of a decimal ID..
  - The 2nd 4 digits: The ID of the 2nd schedule(or “timezone”) defined in section 7.15. The ID of 2nd schedule is transformed to a HEX representation of a decimal ID.
  - The 3rd 4 digits: The ID of the 3rd schedule(or “timezone”) defined in section 7.15. The ID of 3rd schedule is transformed to a HEX representation of a decimal ID.

E.g:

TZ= 000100020010, The group lock out schedules are schedules(Time Zone) 1 , 2 ,16

Result:

Device posts another request to server with result code defined below:

POST /iclock/devicecmd?SN=%s ID=[command ID]&SN=%s&Return=[result code]&CMD=DATA

**7.16. Ring Bell Schedule

**7.16.1 Sync Ring Bell Schedule

Command:

UPDATE BELL BellID=%d\tTime=%s\tMusicID=%d\tVolume=%d\tTimes=%d\tstate=%d\tSUN=%d\tMON=%d\tTUE=%d\tWED=%d\tTHU=%d\tFRI=%d\tSAT=%dBellType=%d\tBellLifeTime=%d

Parameters:

- BellID: Id of Bell to be added from Push Sever.
- Time: What time of day bell should ring is decided from this value. Time should be sent in 24 hour format.
- MusicID: - Value of this parameter can be 0-9. Each value represents unique wav file which is played from device when bell triggers.
- Volume: Value of this parameter can be 0-99. Higher number, intensity of volume is increased. default value of this parameter is 67.
- State: - Value of this parameter can be 0 or 1. Value 0 disables specific Bell while Value 1 enables relevant Bell.
- Day: - Value of SUN, MON….SAT can be either 0 or 1. 0 does not play specific Bell on Specific day while 1 activates specific bell for specific day.
- **BellType :- Value of BellType can be 0, 1 or 2. 0: Internal bell(DEFAULT), 1:external bell, 2 :internal and external bell.
- **BellLifeTime:- The life time of the effectiveness of the bell. The parameter ranges from 0-255.

Description:

Add or Update Bell Settings to device using Push Server.

Result:

POST /iclock/devicecmd?SN=12345 ID=[command ID]&SN=12345&Return=[result code]&CMD=DATA

**7.16.2. Delete Ring Bell Schedule

Command:

DELETE BELL BellID =% d	Delete a bell schedule
DELETE BELL	Delete all bell schedules

Parameters:

- BellID:- Specific Bell which is needed to be deleted from Push server. If absent, delete all ring bell schedules.

Description:

Delete specific Bell Settings from clock using Push server.

Result:

POST /iclock/devicecmd?SN=12345

ID=[command ID]&SN=12345&Return=[result code]&CMD=DATA

7.17. Shortcut Key Setup

7.17.1 Shortkey settings (deprecated)

Command:

UPDATE FUNCTIONKEY ShortKeyID=%d\tShortKeyFun=%d\tStateCode=%d\tStateName=%s\t StateAutoChange=%d\tTime =%s\tSUN=%d\ tMON=%d\tTUE=%d\tWED=%d\tTHU=%d\tFRI=%d\tSAT=%d

7.17.2 Shortkey mappings (deprecated)

Command:

DATA UPDATE ATTSTATE AttStateKey=%d\tAttStateValue=%d

Parameters:

- AttStateKey: same as StateCode set in 7.17.1.
- AttStateValue: the mapped value set in attendance log.

Description:

Map the ATT State (Check in/out, break in/out...) to customer ATT State value.

**7.17.3 New Shortkey settings

Command:

UPDATE SHORTKEY keyID=%d\tkeyFun=%d\tstateCode=%d\tkeyLabel=%s\tmapValue=%d\tautoChange=%d\tTime =%s\tSUN =%d\tMON =%d\tTUE =%d\tWED =%d\tTHU=%d\tFRI =%d\tSAT=%d\tJobGroup=%s\tTipGroup=%s

Parameters:

- KeyID: ID of a shortcut key. The mappings are as listed below:F1=1, F2=2, F3=3, F4=4, F5=5, F6=6, F7=7, F8=8
- KeyFun: Function of a shortcut key.

Code	Description
0	Function key disabled
1	Attendance State Key
2	Work Code

3	View SMS
4	**Help(deprecated)
5	**Query ATTLog(deprecated)
6	View Last Punch
7	**Job Code
8	**Self-service
9	**Tip Code

- StateCode: Value of Attendance state, see Part III, Table 2.6.
  - If value of ShortkeyFun is set to 1, firmware considers value of this parameter. This represents value for ATT State key. For example you can set value of Check-In (ATT state) to 15 using this parameter.
  - If value of ShortkeyFun is not set to 1, firmware ignores this parameter.
  - ****Note**: StateCode is deprecated and replaced by mapValue. Assign StateCode=1 to 8 for ZKTFT30 series
- StateName: Attendance State key label.
  - If value of ShortKeyFun is 1, this parameter sets label name of relevant ATT State. If value of ShortKeyFun is not set to 1, firmware ignores this parameter. For example you can set label "Lunch Out" of function key F-3 using this parameter. It supports up to 24 charactors.
- StateAutoChange: Enable/Disable attendance state auto switching per schedule.
  - ZK firmware has functionality of scheduling ATT state automatically. For example you can schedule "Check-In" to display from 9:00 o' clock to 11:00 o clock. You can schedule "Lunch-Out" to display from 12:00 to 2:00 o clock and so on.
  - Value 1 of this parameter enables this functionality. Value 0 of this parameter disables this functionality.
  - If value of ShortkeyFun is not set to 1, firmware ignores this parameter. The field is only valid in the scenario KeyFun =1
- StateAutoChangeTime
  - For example value of this parameter can be "08:30;09:00;08:00;12:00;11:12;00:00;00:00;"
  - This explains

    Sunday 8:30 in morning ATT state should be changed to "Check-In" automatically.

    Monday 09:00 in morning ATT state should be changed to "Check-In" automatically.

    Friday and Saturday there is no need to change ATT state value automatically.

    00:00 defines there is no need change ATT state on that particular day.
  - Hours and minutes are separated by ":"
  - Time for each day are separated by ";".
  - Change Time for each day has to be defined.
  - If value of ShortkeyFun is not set to 1, firmware ignores this parameter.
- **JobGroup: A String defined in a format that 6 digits(1 or 0) separates by :

  E.g. JobGroup =1:0:1:0:1:1 means the JobGroup ID=1, ID=3 ,ID=5, ID=6 are activated with association to the attendance keys(KeyFun=1) as well as functions keys(KeyFun=7).
- **TipGroup: A String defined in a format that 3 digits(1 or 0) separates by :

  E.g. TipGroup =1:0:1 means the TipCode ID=1, ID=3 are activated with association to the attendance keys(KeyFun=1) as well as functions keys (KeyFun=9).

Description:

This new API is the combination of old function key setup and attendance state mapping, which eases server side implementation.

8. Output Unlock Signal(deprecated)

Command:

AC_UNLOCK

Description：

The server requests the device to output an unlock signal.

Returns：

POST ID=%d&SN=%s&Return=0&CMD=AK_UNLOCK

9. Cancel Alarm Signal(deprecated)

Command:

AC_UNALARM

Description：

The server requests the device to cancel the alarm signal.

Returns：

POST ID=%d&SN=%s&Return=0&CMD=AC_UNALARM

10. File Operation

10.1. Get Files from the Device

Command:

GetFile SourceFilePath

Descriptioin：

The server requests the device to send the device file or package to the server, and the source file path is specified by “SourceFilePath”. The SourceFilePath is always relative to /mnt/mtdblock directory.

For example:

GetFile attlog.dat

Which return data from /mnt/mtdblock/attlog.dat

GetFile data/ssruser.dat

Which returns data from /mnt/mtdblock/data/ssruser.dat

Result：

Device will post the following data in separated request：

POST Message
ID=%d *SN=%s* *FILENAME=%s* *CMD=GetFile* *Return=%d* *Content=%s*

Where:

- ID=%d: represents the command ID;
- SN=%s: represents the device serial number;
- FILENAME=%s: represents the file name;
- Return=%d: represents the file size (in bytes);
- Content=%x: represents the file contents. “%x” can be multiple line text, or binary data.

Result:

POST ID=%d&SN=%s&Return=0&CMD=GetFile

10.2. Send Files to the Device

Command:

PutFile SourceFilePath DestFilePath

Result:

POST ID=%d&SN=%s&Return=[return value]&CMD=PutFile

- Return value is the downloaded file size in bytes.

Description：

This command allows srver send the file or package into device, and device downloads the file to the local location defined by DestFilePath. The server uses the absolute URL path file for file download by device.

- SourceFilePath: The absolute URL path for source file or package, it is required;
- DestFilePath: The destination location where the file is saved on device. It is optional, if in absence, the default location is /mnt/mtdblock on device.

For example,

- PutFile http://server/iclock/file/fw/X938/main.tgz

  The device will dowload main.tgz and save to default location at /mnt/mtdblock.

- PutFile http://server/iclock/ffile/fw/X938/desktop.jpg /mnt/mtdblock/image

  The device will download desktop.jpg and save to /mnt/mtdblock/image folder;

Note:

If the source package is in tgz format with proper file name, it can be decompressed by device. With this approach, PutFile API can be applied for firmware upgrade process. The standard tgz name for firmware is: main.tgz, res.tgz or language.tgz. But the decompress process needs the support from /mnt/mtdblock/auto.sh or /mnt/mtdblock/nand script file. The script file must include the tar command to decompress the package.

**11. DATA QUERY

11.1 Query User Information

Command:

QUERY USERINFO PIN=%d *QUERY USERINFO*

Description：

The server requests user info from the terminal by PIN number or all user info.

Returns：

The server posts the user info and returns value below:

POST ID=%d&Return=0&CMD=DATA

Parameters:

- PIN: User identification Number

Return Value:

**Appendix 1.

11.2 Query Attendance Log

Command:

QUERY ATTLOG StartTime=%s\tEndTime=%s

Description：

The server requests attendance logs in a time period from “StartTime” to “EndTime”.

Returns：

The server posts all attendance logs in the time period and returns value below:

POST ID=%d&Return=0&CMD=DATA

Parameters:

- StartTime: Starting time point of the time period. Time format is YYYY-MM-DDThh:mm:ss

- EndTime: Ending time of the time period. Time format is YYYY-MM-DDThh:mm:ss

Return Value:

**Appendix 1

11.3 Query Fingerprint Templates.

Command:

QUERY FINGERTMP PIN=%d\tFID=%d
QUERY FINGERTMP PIN=%d
QUERY FINGERTMP

Description：

The server requests user's fingerprint by PIN number and FID number or all fingerprint templates.

Returns：

POST ID=%d&Return=0&CMD=DATA

Parameters:

- PIN: User identification Number
- FID: Fingerprint ID ranges from 1 to 10

Return Value:

**Appendix 1

**12. Remote Enrollment

**12.1 Enrolling User Fingerprint

Host application initiates the fingerprint enrollment request remotely by sending the following command

Command:

ENROLL_FP PIN=%d\tFID=%d\tRETRY=%d\tOVERWRITE=%d

Parameters:

- PIN : user ID;
- FID : user's fingerprint template ID number;
- RETRY: the number of times to retry after failed;
- OVERWRITE : whether to overwrite the old fingerprint data or not (1 – Overwrite; 2 – Not overwrite).

Description：

Remote control the device to start the fingerprint enrollment process.

Return Value:

**Appendix 1

**13. Self Service

**13.1 Add/Update Self Service(deprecated)

Command:

UPDATE SELFSERVICEINFO ServiceNum=%d\tServiceName=%s\tEnable=%d

Parameters:

- ServiceNum: The ID of the self-service.
- ServiceName: The name of the self-service
- Enable: Enable/Disable the self-service.

Description:

Add or update a self-service instance to notify the host application once being triggered by self-service function key(see details in 7.11.3) that the service is triggered and expecting further response to display certain information for end user's acknowledgement, such as view schedules, view time period, view time card etc.

Result:

Device posts another request to server with result code defined below:

POST /iclock/devicecmd?SN=%s ID=[command ID]&SN=%s&Return=[result code]&CMD=DATA

**13.2 Self service (deprecated)

The self service action can be triggered by a punch by "Self-service" function key and the device sends the following command to notify the host application.

Command:

GET /iclock/cdata?SN=%s&table=SELFSERVICEDATA&PIN=%d&ServiceNum=%d HTTP/1.1

Server response: (post body):

DATA UPDATE SELFSERVICEDATA ServiceNum=%d\tPIN=%d?\tStyle=%d?\tData=%s?

Parameters:

- SN : Device serial number.
- PIN : User ID;
- ServiceNum: The identification number of the self-service.
- Style : 0- The clock decodes Data string using CSV.
- Data : The BASE64 encoded string of returning message in text using a predefined format, such as CSV.

Result:

Device parses the data encrypted by BASE64, displays the original info and sends another request to server with result code defined below:

POST /iclock/devicecmd?SN=%s ID=[command ID]&SN=%s&Return=[result code]&CMD=DATA

**13.3 Self Service Dialogue

The self service action can be triggered by a punch by "Self-service" function key and the device sends the following command to notify the host application and initiate the dialogue.

Command:

GET /iclock/cdata?SN=%s&table=SELFSERVICEDATA&PIN=%d&ServiceNum=%d HTTP/1.1

Server response: (post body):

DATA UPDATE SELFSERVICEDATA ServiceNum=%d\tPIN=%d?\tStyle=%d?\tData=%s?

Parameters:

- SN : Device serial number.
- PIN : User ID;
- ServiceNum: Server side defined identifier to track the service selected in the 2nd interaction with the host. This number is fixed at 0 as default in the 1st interaction with the host to notify the request of starting a dialogue with the host
- Style :
  - The clock decodes DATA=%s as TEXT; no 2nd interaction with the host will be triggered
  - The clock decodes DATA=%s as CSV, no 2nd interaction with the host will be triggered
  - The clock decodes DATA=%s in the format ID=%d&CMD=%s; ID=%d&CMD=%s; ID=%d&CMD=%s; ID=%d&CMD=%s; ID=%d&CMD=%s;
    - ✓ ID=%d : ID is the ServiceNum defined by the service.
    - ✓ CMD=%s: is the name associated with ServiceNum.
- Data : The BASE64 encoded string of returning message

Result:

Device parses the data encrypted by BASE64, displays the original info and sends another request to server with result code defined below:

POST /iclock/devicecmd?SN=%s ID=[command ID]&SN=%s&Return=[result code]&CMD=DATA

**Self service dialogue example will be provided in Appendix 2.

**14. View Last Punch

The view last punch action can be triggered by a punch by “Last Punch” function key and the device sends the following command to notify the host application.

Command:

GET/ iclock /cdata?SN=xxxxxx&type=LastPunch&PIN=%d

Server response: (post body):

status = [0 | n] *name =%s\ tpin =%s\ tstate =%s\tworkcode=%s\tdate=% s\ttime =%s*

Parameters:

- PIN: - Identified User who wants to see his last punch
- **status: 0=indicates the success. Other value=indicate failure;**
- **name: Name of the user;**
- **pin: UserID of person who request last Punch;**
- **state: Attendance state of Last Punch;**
- **workcode: The workcode value selected or input in last punch;**
- **date: The date of last punch;**
- **time: time of last punch.**

Description:

To use this functionality first one needs to defined one short cut key for last Punch functionality.

End user has to hit “Last Punch” function key. Upon doing so firmware ask for verification.

Upon successful verification firmware sends below get request to server.

GET /iclock/cdata?SN=%S&type=LastPunch&PIN=%d

Server responses to above Get request as discussed above

Firmware displays return value from server to LCD.

****Note:**

When Push functionality is enabled, it allows firmware to fetch Last Punch information from web server or local database if PUSH is disabled or device is offline. ZK TFT 30 series only supports to retrieve last punch when device is online

**15. Manually Add Punches

Description:

The function is designed to enable authorized user (e.g. Admin) to add punches in the device to make up attendance record which failed to be created for some reason and submit the punch to the host application.

**Appendix 1

Error Code	Description
0	Successful
2	Enrolling User Print, the fingerprint of corresponding user already exists
4	Enrolling User Print, enrollment failure, which is usually due to poor quality or not the same fingerprint enrolled for three times
5	Enrolling User Print, the enrolled fingerprint already exists in the database
6	Enrolling User Print, enrollment is cancelled
7	Enrolling User Print, the equipment is busy and enrollment cannot be conducted
-1	The parameter is incorrect
-2	The transmitted user photo data does not match the given size

-3	Reading or writing is incorrect
-9	The transmitted template data does not match the given size.
-10	The user specified by PIN does not exist in the equipment
-11	The fingerprint template format is illegal
-12	The fingerprint template is illegal
-1001	Limited capacity
-1002	Not supported by the equipment
-1003	Command execution timeout
-1004	The data and equipment configuration are inconsistent
-1005	The equipment is busy
-1006	The data is too long
-1007	Memory error

**Appendix 2

Self Service dialogue example with user ID 1999

1. User 1999 selects function key "Self-Service" and verified identity.
2. The device will start 1st interaction with the service by sending out the exact following command.

```
GET /iclock/cdata?SN=123456789&table=SELFSERVICEDATA&PIN=1999&ServiceNum=0
```

3. The service responses to the cdata request by the exact POSTBODY

```
POST /iclock/devicecmd?SN=123456789
DATA UPDATE SELFSERVICEDATA ServiceNum=0    PIN=1999    Style=2    Data=
SUQ9MSZDTUQ9VmFjYXRpb247SUQ9MiZDTUQ9UFRPO0lEPTMmQ01EPUFjY3VyYWw7SUQ9NC
```

ZDTUQ9U2NoZWR1bG

*Data is "ID=1&CMD=Vacation;ID=2&CMD=PTO;ID=3&CMD=Accural;ID=4&CMD=Schedules" before BASE64 encode.

4. User 1999 will find out 4 selections on the UI as Vacation/ PTO/Accural/Schedules after the clock successfully parses the response of the service.
5. The device will start 2nd interaction with the service by pushing out the exact following command after user 1999 selects "Schedules"

GET /iclock/cdata?SN=123456789&table=SELFSERVICEDATA&PIN=1999&ServiceNum=4

6. The service responses to the cdata request by the exact following command

POST /iclock/devicecmd?SN=123456789

DATA UPDATE SELFSERVICEDATA ServiceNum=4 PIN=1999 Style=1
Data=TW9uZGF5LDE6MDAtMjowMCBwbSwzOjAwLTQ6MDAgcG0sMTA6MDAtMTE6MDAgcG0NClR
1ZXNkYXksMTowMC0
yOjAwIGFtLDM6MDAtNDowMCBhbSwxMDowMC0xMTowMCBhbQ==

*Data is "Monday,1:00-2:00 pm,3:00-4:00 pm,10:00-11:00 pm Tues,1:00-2:00 am,3:00-4:00 am,10:00-11:00 am" before BASE64 encode.

7. The device will display the following data on the screen and return in POSTBODY "ID=567&SN=123456789&Return=0&CMD=ServiceNum"

Monday,1:00-2:00 pm,3:00-4:00 pm,10:00-11:00 pm
Tues,1:00-2:00 am,3:00-4:00 am,10:00-11:00 am