

<h1>Quio</h1>	<b>SPECIFICATION</b>	Model No.	QU-TK-F3
		Date	2010/01/28
	<b>DLL</b>	Ver.	v20100128
		Page	1/16

## DLL SPECIFICATION

### QU-TK-F3 Card Issuing Machine

#### -Windows DLL Specification:

Windows32 DLL on windows platform, files names as follow :

1. QU-TK\_571.H                32 digit dynamic library header files
2. QU-TK\_571.LIB            32 digit input dynamic libraries
3. QU-TK\_571.DLL          Windows32 DLL file

#### -Applicable Card Type:

1. RFID Card: ISO/IEC 14443 TYPE A / B, MIFARE one (S50, S70, UL)
2. IC Card:
  - SLE4428, SLE4442
  - 24C01A, 24C02, 24C04, 24C08, 24C16, 24C32, 24C64
  - Contact CPU Card (T=0/T=1)
  - SAM/SIM Card (T=0/T=1)

#### Notes:

1. Code examples of DLL which is running on VC6, VB6, DELPHI7, C++BUILD6, PB9, VB2005.NET, V#2005.NET environment are provided. Please contact with the corresponding sales to obtain the file
2. Technical support for DLL running on Linux/Unix (.so) can be provided for a large demand.
3. Technical support for JAVA environment can be provided for a large demand.
4. Technical support for OCX design and Web invocation can be provided for a large demand.

<h1>Quio</h1>	<b>SPECIFICATION</b>	Model No.	QU-TK-F3
		Date	2010/01/28
	<b>DLL</b>	Ver.	v20100128
		Page	2/16

## CONTENTS

1	Function Specification .....	3
1.1	QU-TK-F3 Address Setting .....	3
1.2	Function Parameter Specification .....	4
2	API Specification .....	5
2.1	Com Port Open Function (Default Baud 38400bps).....	5
2.2	Open Com Function By Specified Baud.....	6
2.3	Close Com Port Function.....	7
2.4	Execute Command Function.....	8
2.5	IC Card Transportation Function.....	9
3	Sample Code .....	10
3.1	Parts Command List.....	10
3.2	Sample Code (Contact CPU Card Operation).....	11

<h1>Quio</h1>	<b>SPECIFICATION</b>	Model No.	QU-TK-F3
		Date	2010/01/28
	<b>DLL</b>	Ver.	v20100128
		Page	3/16

## 1 Function Specification

### 1.1 QU-TK-F3 Address Setting

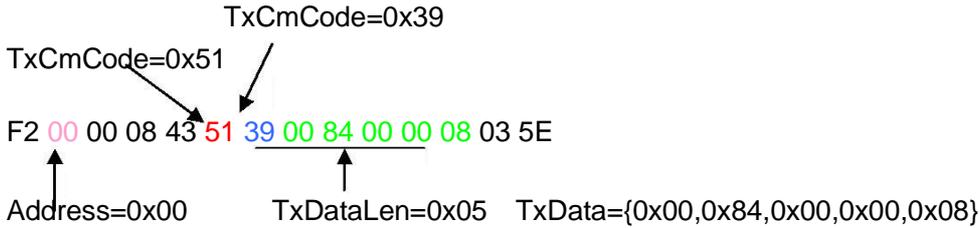
The following table shows the Dip setting of Unit Address.

DIP				Address
4	3	2	1	
ON	ON	ON	ON	0x00
ON	ON	ON	OFF	0x01
ON	ON	OFF	ON	0x02
ON	ON	OFF	OFF	0x03
ON	OFF	ON	ON	0x04
ON	OFF	ON	OFF	0x05
ON	OFF	OFF	ON	0x06
ON	OFF	OFF	OFF	0x07
OFF	ON	ON	ON	0x08
OFF	ON	ON	OFF	0x09
OFF	ON	OFF	ON	0x0A
OFF	ON	OFF	OFF	0x0B
OFF	OFF	ON	ON	0x0C
OFF	OFF	ON	OFF	0x0D
OFF	OFF	OFF	ON	0x0E
OFF	OFF	OFF	OFF	0x0F

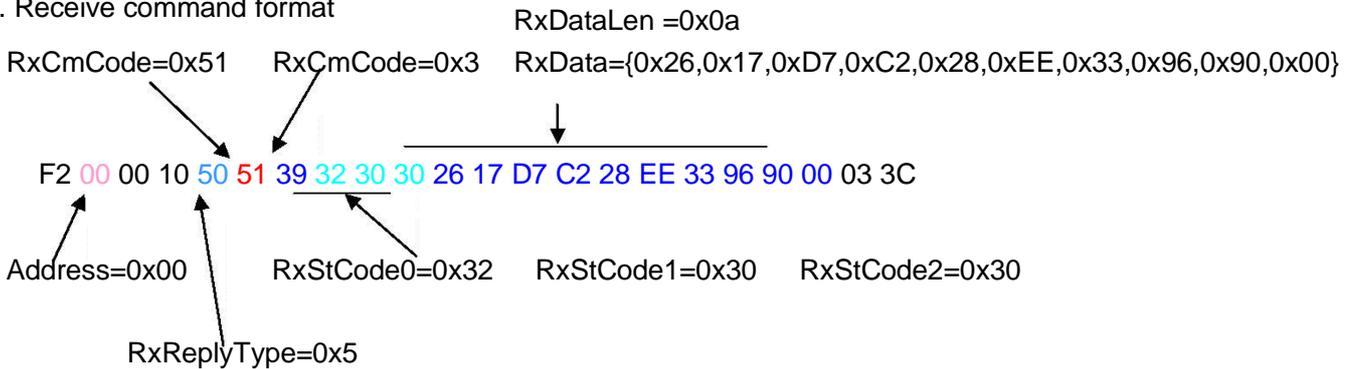
<h1>Quio</h1>	<b>SPECIFICATION</b>	Model No.	QU-TK-F3
		Date	2010/01/28
	<b>DLL</b>	Ver.	v20100128
		Page	4/16

## 1.2 Function Parameter Specification

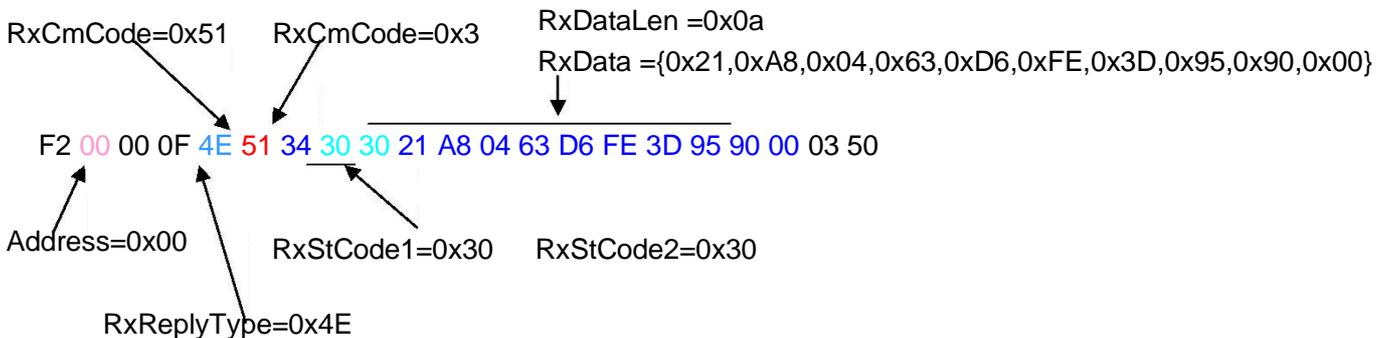
### 1. Send command format



### 2. Receive command format



### 3. Receive command format



### Notes :

Receive command (Successful) : RxReplyType=0x50. (RxStCode0, RxStCode1, RxStCode2 is state code).

Receive command (Error) : RxReplyType=0x4E. (RxStCode1, RxStCode2 is error state code).

<h1>Quio</h1>	<b>SPECIFICATION</b>	Model No.	QU-TK-F3
		Date	2010/01/28
	<b>DLL</b>	Ver.	v20100128
		Page	5/16

## 2 API Specification

### 2.1 Com Port Open Function (Default Baud 38400bps)

HANDLE APIENTRY CommOpen(char \*Port);

**Parameter:**

**Port:** Serial Number String

Eg :**ComOpen("COM1");**

**Return:**

- 0 (Com open error)  
Possible reason:
  - ① Invalid Serial Number String
  - ② This com occupy by other device
- <>0 Com Open Success

**Notes:** 1)Call this function before others.

2)Enable to open several com port to get serial number string, but disable to open one port twice at same time.

3) Close com port by using CommClose() after the completion of all operations.

<h1>Quio</h1>	<b>SPECIFICATION</b>	Model No.	QU-TK-F3
		Date	2010/01/28
	<b>DLL</b>	Ver.	v20100128
		Page	6/16

## 2.2 Open Com Function By Specified Baud

HANDLE APIENTRY CommOpenWithBaut(char \*Port, unsigned int Baudrate):

### Parameter:

Port: Serial Number String

Baud rate

Baud rate =9600, 19200, 38400, 57600,115200

Eg: CommOpenWithBaut ("COM1",38400);

### Return :

0 Com Open error

Possible reason:

① Invalid Serial Number String

② This com occupy by other device

<>0 Com Open Success

**Notes:** 1)Call this function before others.

2)Enable to open several com port to get serial number string, but disable to open one port twice at same time.

3) Close com port by using CommClose() after the completion of all operations.

<h1>Quio</h1>	<b>SPECIFICATION</b>	Model No.	QU-TK-F3
		Date	2010/01/28
	<b>DLL</b>	Ver.	v20100128
		Page	7/16

## 2.3 Close Com Port Function

**int APIENTRY CommClose(HANDLE ComHandle):**

**Parameter:**

ComHandle : Serial Number String

**Return:**

=0 Success

<>0 Fail

**Notes:**

CommOpen() or CommOpenWithBaut should use together to close port after the completion of all operations.

<h1>Quio</h1>	<b>SPECIFICATION</b>	Model No.	QU-TK-F3
		Date	2010/01/28
	<b>DLL</b>	Ver.	v20100128
		Page	8/16

## 2.4 Execute Command Function

```
int APIENTRY ExecuteCommand(HANDLE ComHandle, BYTE TxAddr,
                           BYTE TxCmCode, BYTE TxPmCode,
                           int TxDataLen, BYTE TxData[],
                           BYTE *RxReplyType,
                           BYTE *RxStCode0, BYTE *RxStCode1, BYTE *RxStCode2,
                           int *RxDataLen, BYTE RxData[])
```

### Function:

Execute command and return result

### Parameter:

ComHandle : Serial Number String

TxAddr: Unit Address (0x00~0x0f)

TxCmCode: Command Code

TxPmCode: Parameter Code

TxDataLen: Data Length

TxData: Data

RxReplyType: Return Reply Type

0x50 : Success

0x4E : Fail

0x10 : Cancel communication by machine (NAK )

0x20 : Communcation error

0x30 : Cancel communication by HOST(DLE, EOT)

RxStCode0: Return state code 0 // Pls see more detain from communication protocol

RxStCode1: Return state code 1 // Pls see more detain from communication protocol

RxStCode2: Return state code 2 // Pls see more detain from communication protocol

RxDataLen: Return package code

RxData: Return Data

### Return:

=0 Success

<>0 Fail

### Notes :

Receive command (Successful) : RxReplyType=0x50. (RxStCode0, RxStCode1, RxStCode2 is state code).

Receive command (Error) : RxReplyType=0x4E. (RxStCode1, RxStCode2 is error state code).

<h1>Quio</h1>	<b>SPECIFICATION</b>	Model No.	QU-TK-F3
		Date	2010/01/28
	<b>DLL</b>	Ver.	v20100128
		Page	9/16

## 2.5 IC Card Transportation Function

```
int APIENTRY ICCardTransmit (HANDLE ComHandle, BYTE TxAddr,
                             BYTE TxCmCode, BYTE TxPmCode,
                             int TxDataLen, BYTE TxData[],
                             BYTE *RxReplyType,
                             BYTE *RxCmCode, BYTE *RxPmCode,
                             BYTE *RxStCode0, BYTE *RxStCode1, BYTE *RxStCode2,
                             int *RxDataLen, BYTE RxData[]);
```

### Function :

Execute command and return result

**Notes :** Add two parameters RxCmCode and RxPmCode comparing to Execute Command

### Parameter:

ComHandle : Serial port handle

TxAddr: Unit Address (0x00~0x0f)

TxCmCode: Command Code

TxPmCode: Parameter Code

TxDataLen: Data length

TxData: Data

RxReplyType: Return reply type

0x50 : Successful

0x4E : Fail

0x10 : Cancel Communication by machine NAK (15H)

0x20 : Communication Fail

0x30 : Cancel Communication by HOST (DLE, EOT (10H, 04H))

RxCmCode: Return Command Code

RxPmCode: Return parameter Code

RxStCode0: Return state code0 // Pls see more detail from communication protocol

RxStCode1: Return state code 1 // Pls see more detail from communication protocol

RxStCode2: Return state code 2 // Pls see more detail from communication protocol

RxDataLen: Return package length

RxData: Return data

### Return:

=0 Success

<>0 Fail

### Notes :

Receive command (Successful) : RxReplyType=0x50. (RxStCode0, RxStCode1, RxStCode2 is state code).

Receive command (Error) : RxReplyType=0x4E. (RxStCode1, RxStCode2 is error state code).

<h1>Quio</h1>	<b>SPECIFICATION</b>	Model No.	QU-TK-F3
		Date	2010/01/28
	<b>DLL</b>	Ver.	v20100128
		Page	10/16

## 3 Sample Code

### 3.1 Parts Command List

- cm: Command code pm: Parameters

Command	cm	Function	pm	Details of operation
INITIALIZE	30H	Initialize ICRW	30H	If card is inside ICRW, move card to gate
			31H	If card is inside ICRW, capture card to bin
			33H	If card is inside ICRW, does not move the card
			34H	Same as 30H. And Retract counter will work
			35H	Same as 31H. And Retract counter will work
			37H	Same as 33H. And Retract counter will work
STATUS REQUEST	31H	Inquire status	30H	Report presence of card and its position
			31H	Report presence of sensor status in detail
CARD MOVE	32H	Card move	30H	Move card to Gate and with hold card
			31H	Move card to IC contact position
			32H	Move card to RF
			33H	Capture card to BIN
			39H	Move card to Gate and without hold card
CARD IN	33H	Disable/Enable	30H	Accepts all card In
			31H	Disable
TEST CARD TYPE	50H	AutoTest card type	30H	AutoTest IC card type
			31H	AutoTest RF card type
CPU CARD	51H	CPU Card Operation	30H	CPU card cold reset
			31H	CPU card power off
			32H	CPU Cards Status Inquiry
			33H	CPU cards T = 0 protocol APDU data exchange
			34H	CPU cards T = 1 protocol APDU data exchange
			38H	CPU Card Hot Reset
			39H	Automatically distinguish between T = 0 / T = 1 protocols CPU card APDU data exchange
RFCARD	60H	RF Card Operation	30H	RF Card Activation
			31H	RF Card Deactivation
			32H	RF card operation status query
			33H	Mafare one Card Standard Operation
			34H	Type A standard T = CL card APDU data exchange
			35H	Type B standard T = CL card APDU data exchange

<h1>Quio</h1>	<b>SPECIFICATION</b>	Model No.	QU-TK-F3
		Date	2010/01/28
	<b>DLL</b>	Ver.	v20100128
		Page	11/16

### 3.2 Sample Code (Contact CPU Card Operation)

```

HANDLE hCom;    // Target device's HANDLE
int rc=0;      //Result
unsigned char Addr;
unsigned char CmCode;
unsigned char PmCode;
int  CmDataLen;
unsigned char CmData[1024];

unsigned char ReType;
unsigned char St2;
unsigned char St1;
unsigned char St0;
int  ReDataLen;
unsigned char ReData[1024];
// Open Comm. port
{
    hCom=CommOpenWithBaut("COM1",38400);
    if(hCom<=0)
    {
        // failed
        ...
    }
    else
    {
        // successfully.
        ...
    }
}
// Initialize
{
    memset(CmData,0x00,sizeof(CmData));
    Addr=0x00;
    CmCode=0x30; // Initialize command
    PmCode=0x30; // Parameter code
    CmDataLen=1; // Data size (bytes)
    CmData[0]=0x30;
    rc=ExecuteCommand(hCom,Addr,CmCode,PmCode,CmDataLen,CmData,
        &ReType,&St0,&St1,&St2,&ReDataLen,ReData);
    if(rc==0)
        // Initialize command successfully finished.

```

<h1>Quio</h1>	<b>SPECIFICATION</b>	Model No.	QU-TK-F3
		Date	2010/01/28
	<b>DLL</b>	Ver.	v20100128
		Page	12/16

```

if (ReType==0x50)
{
    // Received positive reply
    ...
}
else (ReType==0x4e)
{
    // Received negative reply
    ...
}
else
{
    //Communication Error
    // Initialize command failed.
    ...
}
}

```

**// Setting Card In**

```

{
    memset(CmData,0x00,sizeof(CmData));
    CmCode= 0x33; // Card In
    PmCode= 0x30; // Enable
    CmDataLen= 0; // Data size
}

```

or

**// Move card to IC contact position**

```

{
    memset(CmData,0x00,sizeof(CmData));
    CmCode= 0x32; // Move Card to
    PmCode= 0x31; // IC contact position
    CmDataLen= 0; // Data size
}

```

```

rc=ExecuteCommand(hCom,Addr,CmCode,PmCode,CmDataLen,CmData,
                  &ReType,&St0,&St1,&St2,&ReDataLen,ReData);

```

```

if(rc!=0 || ReType != 0x50)

```

```

{
    // Command sending failed or command execution failed
    ResErrMsg(St1,St2); // ResErrMsg is a function to show the error reason
                       // Pls see communication protocol specification.to more detail
    ...
}
}

```

**// Request Status**

```

{
    memset(CmData,0x00,sizeof(CmData));
    CmCode= 0x31; // Status request command
    PmCode= 0x30; // Parameter code
    CmDataLen=0; // Data size
    rc=ExecuteCommand(hCom,Addr,CmCode,PmCode,CmDataLen,CmData,

```

<h1>Quio</h1>	<b>SPECIFICATION</b>	Model No.	QU-TK-F3
		Date	2010/01/28
	<b>DLL</b>	Ver.	v20100128
		Page	13/16

```
&ReType,&St0,&St1,&St2,&ReDataLen,ReData);
```

```
if(rc==0)
    if (ReType==0x50)
    {
        //Execute Ok
        if(St0 == '2')    // There is a card at IC contact position
        {
            // status code="2xx"
            // Detected a card inside of Card Reader/Writer
            ...
        }
    }
    else (ReType==0x4e)
    {
        //Command execution failed
        ResErrMsg(St1,St2);
        ...
    }
else
{
    //Communication Error
    ...
}
}
```

```
// CPU card cold reset
```

```
{
    memset(CmData,0x00,sizeof(CmData));
    CmCode= 0x51; // CPU card
    PmCode=0x30; // Cold reset
    CmDataLen=1; // Data size
    If (true)    //Choice Mode
    {
        CmData[0]=0x30;    //EMV2000 V4.0 Vcc=5v
    }
    else
    {
        CmData[0]=0x33;    // ISO/IEC7816-3
    }
    rc=ExecuteCommand(hCom,ICRWaddr,CmCode,PmCode,CmDataLen,CmData,&ReType,&St0,&St1,&St2,&ReDataLen,ReData);
    if(rc==0)
        if (ReType==0x50)
        {
```

<h1>Quio</h1>	<b>SPECIFICATION</b>	Model No.	QU-TK-F3
		Date	2010/01/28
	<b>DLL</b>	Ver.	v20100128
		Page	14/16

```

CString str,t;
CString TempBuf="";
for(int n=1; n<ReDataLen; n++) //
    {
        t.Format("%02x",ReData[n]);
        TempBuf += t;
        t="";
    }
if (ReData[0]==48)
{
    MessageBox("CPU Card(T=0) Deactivate Succeeded and CPU card information meet the
EMV mode ATR ");
}
else
{
    MessageBox("CPU Card(T=1) Deactivate Succeeded and CPU card information meet the
EMV mode ATR");
}
}
else if((ReType==0x4e) && (ReDataLen>0))
{
    CString str,t;
    CString TempBuf="";
    for(int n=1; n<ReDataLen; n++) //
        {
            t.Format("%02x",ReData[n]);
            TempBuf += t;
            t="";
        }
    if (ReData[0]==48)
    {
        MessageBox("CPU Card(T=0) Deactivate Succeeded,But CPU card information does not
meet the EMV mode ATR ");
    }
    else
    {
        MessageBox("CPU Card(T=1) Deactivate Succeeded,But CPU card information does not
meet the EMV mode ATR");
    }
}
else
{
    SANKYOErrMsg(St1,St2);
}

```

<h1>Quio</h1>	<b>SPECIFICATION</b>	Model No.	QU-TK-F3
		Date	2010/01/28
	<b>DLL</b>	Ver.	v20100128
		Page	15/16

```

    }
else
{
    MessageBox("Communication Error");
}
}
// Exchanges data between the Host Computer and IC card
{
    memset(CmData,0x00,sizeof(CmData));
    CmCode= 0x51; // IC card control
    PmCode=0x39; // Deactivate
    CmDataLen=5; // Data size
    CmData[0]=0x00;
    CmData[1]=0x84;
    CmData[2]=0x00;
    CmData[3]=0x00;
    CmData[4]=0x08;
    rc=ExecuteCommand(hCom,Addr,CmCode,PmCode,CmDataLen,CmData,
        &ReType,&St0,&St1,&St2,&ReDataLen,ReData);
    if(rc!=0)
    {
        // ICCardTransmit failed
        ...
    }
else
{
    if (ReType==0x50)
    {
        CString t;
        CString TempBuf="";
        for(int n=0; n<ReDataLen; n++) // R-APDU= start from n=0
        {
            t.Format("%02x",ReData[n]);
            TempBuf += t;
            t="";
        }
        // TempBuf is the buffer of R-APDU data
    }
else //(ReType==0x4e)
{
    // Command execution failed
    SANKYOErrMsg(St1,St2);
}
}
}

```

<h1>Quio</h1>	<b>SPECIFICATION</b>	Model No.	QU-TK-F3
		Date	2010/01/28
	<b>DLL</b>	Ver.	v20100128
		Page	16/16

```

    }
}
// CPU card power off
{
    memset(CmData,0x00,sizeof(CmData));
    CmCode=0x51; // CPU card power off
    PmCode=0x31; // Power off
    CmDataLen=0; // Data size
    rc=ExecuteCommand(hCom,Addr,CmCode,PmCode,CmDataLen,CmData,
        &ReType,&St0,&St1,&St2,&ReDataLen,ReData);
    if(rc!=0 || ReType != 0x50)
    {
        // Command sending failed or command execution failed
        ...
    }
}
_EXIT1:
// Closes communications between the Host Computer and the Card Reader/Writer
CommClose(hCom);
_EXIT:

```