DSECASH
eCash Evaluation Kit

www.maxim-ic.com

GENERAL DESCRIPTION

The eCash Evaluation Kit demonstrates the speed, reliability, and security of a SHA-1 based iButton® eCash system. The provided eCash debit board is a complete stand-alone module that will perform monetary debits in a fast 100ms. The eCash debit board has a serial interface that allows a PC or microprocessor to monitor or provide manual control of the debit process. Utilizing the serial interface, this demo could easily be integrated into a real eCash or access control system.

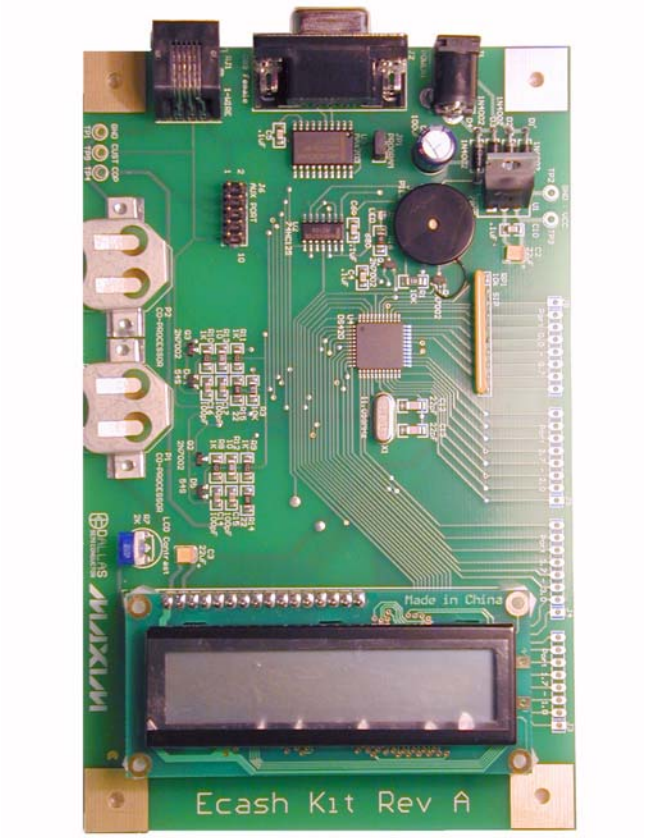
EVALUATION KIT CONTENTS

(3) DS1963S – Coprocessor or user tokens
(2) DS1961S – User tokens
(4) DS9093A – (2) black, (2) blue
(1) DS1402-DR8 – iButton Blue Dot™ receptor
(1) DS9097U-S09 – 1-Wire® PC serial port adapter for PC initialization of coprocessor and user tokens
(1) DB9 Serial cable – Connects eval board to serial port of a computer to monitor eval board
(1) eCash evaluation board
(1) Instruction sheet

FEATURES

- Stand-alone eCash evaluation circuit board with LCD display and Audible feedback.
- Supports both DS1963S and DS1961S SHA-1 iButtons as debit tokens.
- Once initialized, the DS1963S coprocessor on the debit board keeps the money secret secure
- Secure eCash debits in 100ms (approximate).
- 2 Java™ programs (compatible with Windows® and Linux) provided for download to initialize eCash coprocessors and tokens as well as to monitor evaluation board.
- Evaluation board can be used as a component in a larger control system (Service Control Unit).
- Complete firmware source provided in portable 'C' code.
- Schematic design and list of components of eval board provided.
- Evaluation board can be used for eCash code development with the onboard single-cycle 8051 compatible DS89C420.

Figure 1. eCash Evaluation Board



ORDERING INFORMATION

PART	DESCRIPTION
DSECASH	eCash Evaluation Kit

REQUIREMENTS:

- The external supply for the eCash evaluation board must be provided. Power supply requirements: AC/DC, 9-20 V, 200mA minimum. See *Power Connector* below for recommendations.
- Internet connection required to get initialization and monitoring software.

INTRODUCTION

The primary purpose of the eCash Evaluation Kit is to demonstrate SHA-1 iButtons by showing a fully qualified, SHA-1 authenticated, monetary debit in ~100ms using the file and security standards described in other Dallas Semiconductor Application Notes (see White Paper 1 in table below). Applications that this kit targets include: vending, parking meters, toll booths, pay phones, public transportation, gaming, and others requiring either secure payment or user authentication. The following Application Notes dealing with iButton and 1-Wire devices are as follows (it is recommended to start with White Paper 8: "1-Wire SHA-1 Overview"):

Table 1. SHA-1 Application Note List

White Paper 8: 1-Wire SHA-1 Overview
White Paper 4: Glossary of 1-Wire SHA-1 Terms
White Paper 3: Why are 1-Wire SHA-1 Devices Secure?
White Paper 1: SHA Devices Used in Small Cash Systems
App Note 150: Small Message Encryption using SHA Devices
App Note 151: Dallas Digital Monetary Certificates
App Note 152: SHA iButton Secrets and Challenges
App Note 154: Passwords in SHA Authentication
App Note 156: DS1963S SHA 1-Wire API Users Guide
App Note 157: SHA iButton API Overview

The kit incorporates an LCD display, two coprocessors (the DS1963S and DS1961S), external 1-Wire, and a serial port. The board can be operated independently of a PC after initialization of the coprocessors and user tokens. It can also be connected to a PC for configuration, monitoring, and control. The developer can take an already existing embedded system, add some simple serial code to control the eCash board, and quickly be up and running with an iButton-based debiting system, using the eCash evaluation board as a subassembly in a larger system. For this, a serial port and an IDC connector have been made available on the demo board for testing and bench development. Since the firmware's C source is provided with the kit, the developer is also free to extend the capabilities of the system and/or customize it.

The kit operating instructions and software are available online: http://www.ibutton.com/ibuttons/ecashkit.html. Two Java programs, eCashInit.java and eCashMonitor.java are included in the software downloads. The eCashInit program initializes iButtons for use as coprocessors and user tokens, and the eCashMonitor program communicates directly with the evaluation board through the included serial cable.

BASIC CONFIGURATION

To properly setup and configure the eCash evaluation board, the first step is to download and configure the software. Once that is done, then the hardware of the kit can be configured and basic SHA-1 monetary debits shown. Optionally, the evaluation board's firmware can be extended/ehanced/re-written by the developer and loaded onto the board's processor.

Software Configuration

Prior to first-time use, it is recommended that the software associated with the kit be installed and configured first.. Afterwards, initialization of the SHA-1 iButtons can take place using the software. At least one iButton will need to be setup as a coprocessor and one as a user token. This can be done easily on a PC with the Java Runtime Environment (JRE) pre-installed. If a JRE has not been installed, please visit http://java.sun.com to download and install one.

With Java correctly installed, both the eCashInit and eCashMonitor programs can be setup as Java "Web Start" programs. One of the advantages of "Web Start" technology is that every time one of the programs is run, a background process checks for updates to the program and automatically downloads and installs them from our website. The Java "Web Start" page for both of these programs can be found at the following URL: http://www.ibutton.com/ibuttons/ecashjava.html. This page contains complete download, installation, and troubleshooting instructions.

Manual Software Installation

If for some reason, a manual installation of the eCash software is necessary, instructions are available below. Specifically, see the sections entitled *RXTX Instructions, eCashInit Setup Instructions, and eCashMonitor Setup Instructions.*

RXTX Instructions

Both included Java programs, eCashInit and eCashMonitor, require RXTX 2.1 to be installed on the PC. RXTX is a cross-platform serial library used by eCashInit and eCashMonitor to communicate to the DS9097U and to the eCash eval board. The installation of RXTX is performed automatically through the Java "Web Start" process, but, just in case a manual install is necessary, the binaries for the Win32 build are included in the kit download (available from the above URL: http://www.ibutton.com/ibuttons/ecashkit.html). Specifically, RXTX binaries can be found in the software/rxtx folder of the download. If a platform other than 32-bit Windows is used, the binaries and/ or source to RXTX is downloadable from http://www.rxtx.org.

eCashInit Setup Instructions

Before running the eCashInit program, first install it either through "Web Start" as mentioned above (recommended) or install it by hand. Again, the "Web Start" version can be found at: http://www.ibutton.com/ibuttons/ecashkit.html. To compile and run by hand, follow the instructions below.

To compile the eCashInit:

```
javac -classpath "<path to 1-Wire API>/lib/OneWireAPI.jar;." *.java
```

To run the eCashInit:

```
java -classpath "<path to 1-Wire API>/lib/OneWireAPI.jar;." eCashInit
```

Please note that the OneWireAPI.jar can be downloaded from the eCash Kit's Java Web Start web page: http://www.ibutton.com/ibuttons/ecashjava.html.
It can also be found in the 1-Wire API for Java Software Development Kit (SDK) located at the following web address: http://www.ibutton.com/software/1wire/1wire_api.html.

Setting up RXTX and OneWireAPI.jar is handled automatically when installing/running the program with Java "Web Start".

eCashMonitor Setup Instructions

Similar to the eCashInit program above, it can either be installed through "Web Start" as mentioned above (recommended) or installed by hand. To compile and run by hand, follow the instructions below.

To compile the eCashMonitor:

```
javac -classpath "<path to 1-Wire API>/lib/OneWireAPI.jar;." *.java
```

To run the eCashMonitor:

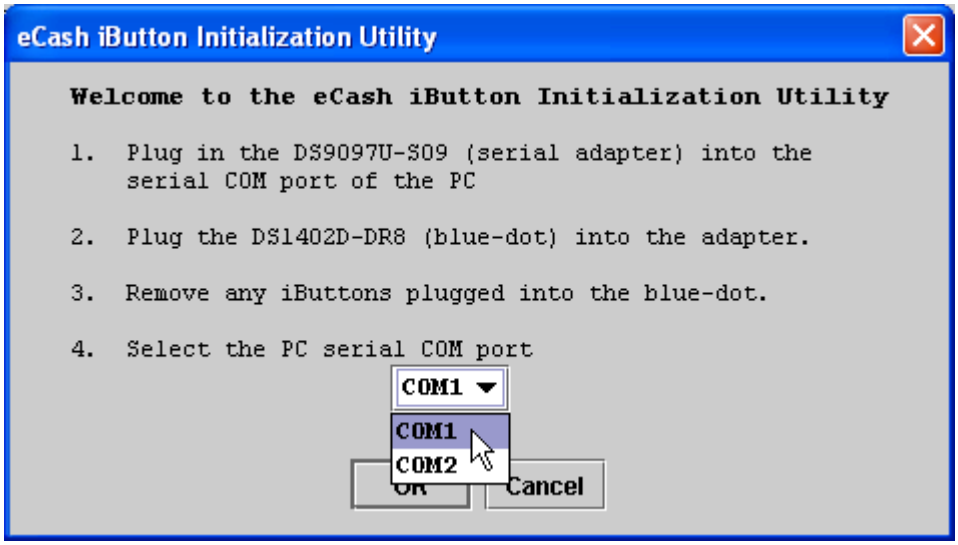
```
java -classpath "<path to 1-Wire API>/lib/OneWireAPI.jar;." eCashMonitor
```

Note that the eCashMonitor program is still in development. It can be used to monitor the evaluation board and any eCash transactions being performed by it. It should properly translate all 'events' and output them in plain English (i.e. whenever a token is debited, it should properly display the debit amount and current balance for that token). For a list of all the commands available for the eCash board, see the section entitled *APPENDIX A: THE ECASH PROCESSOR.*

Initializing iButtons as Coprocessors or User Tokens

With the DS9097U 1-Wire adapter plugged into the serial port and the DS1402-DR8 Blue Dot plugged into the adapter, run the eCashInit Java program. The following Window appears, asking for which COM port to use with the DS9097U:

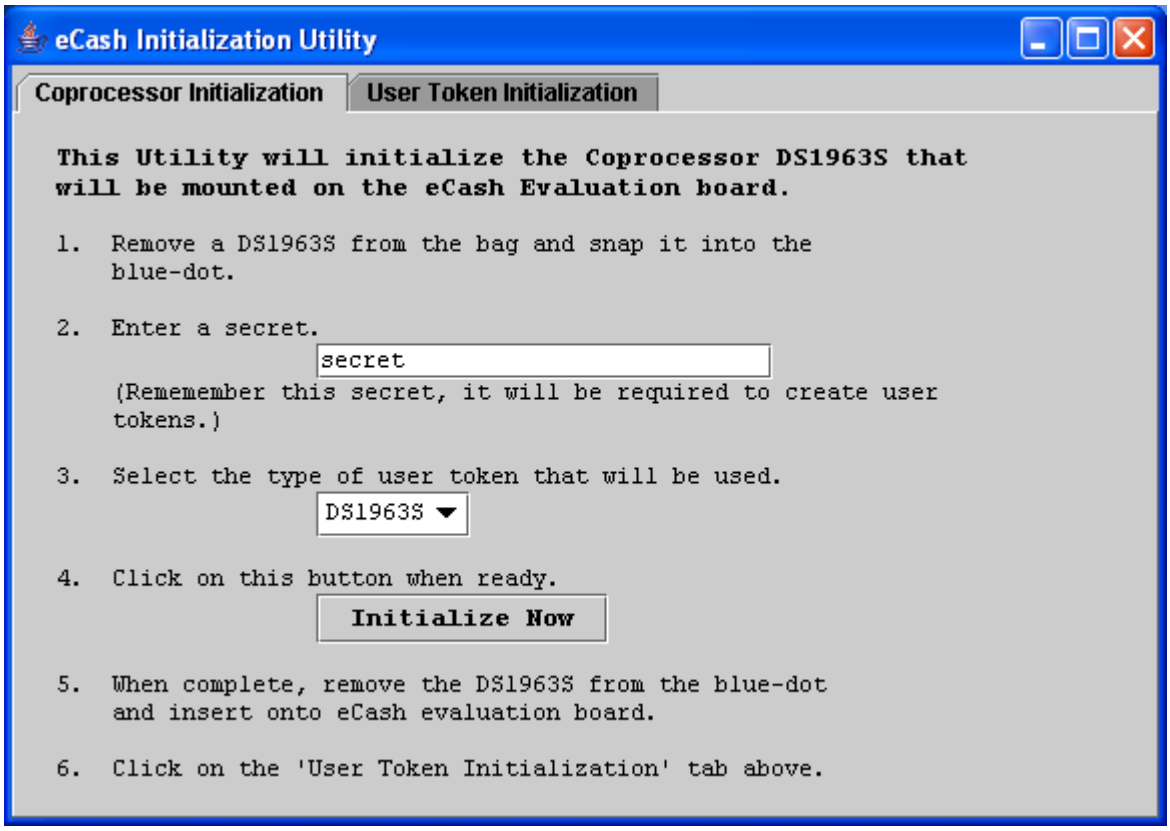
Figure 2. eCashInit Start Screen



Follow the steps presented and select the appropriate COM port. Click OK when finished.

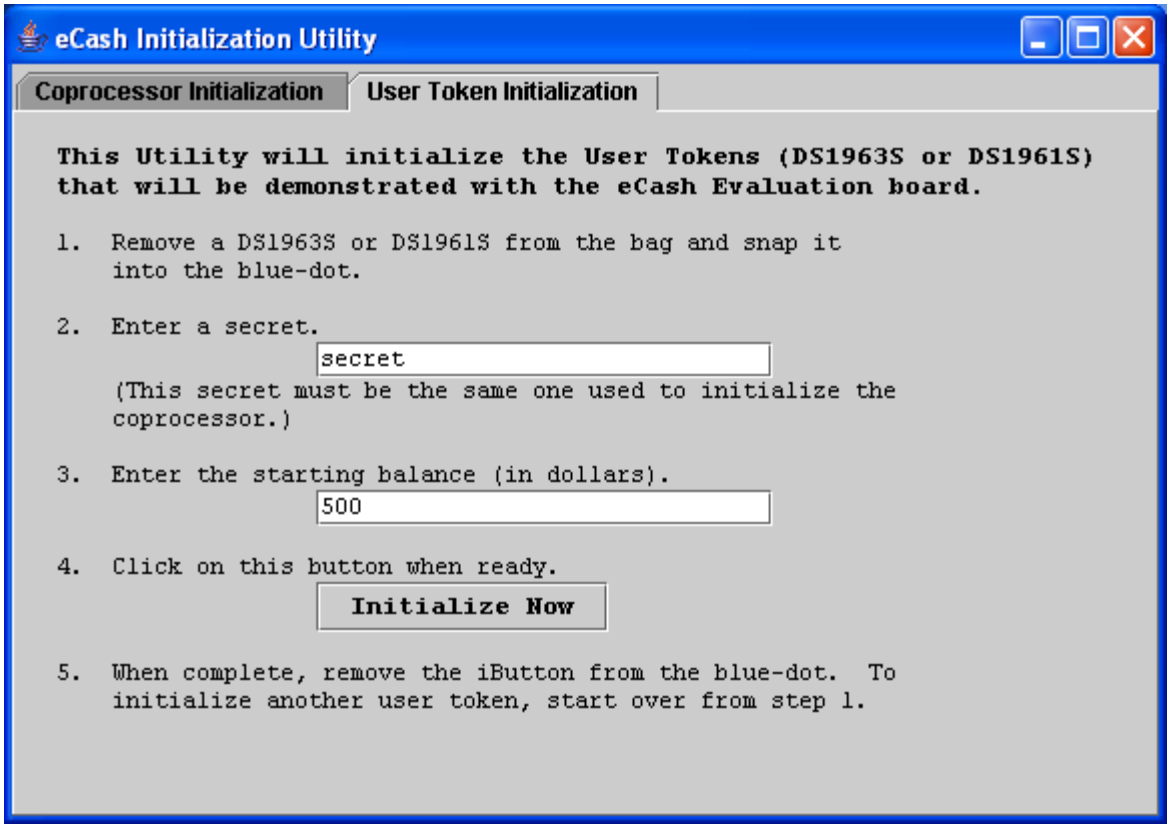
The next window to appear gives a choice between two tabs. Selecting the first tab starts the process for creating a SHA-1 coprocessor. Take a DS1963S iButton, snap it into the Blue Dot, and follow the instructions. When finished, snap the coprocessor into one of the iButton clips provided on the evaluation board. Note that the type of user token (DS1961S or DS1963S) needs to be specified. See screen capture below.

Figure 3. eCashInit Coprocessor Initialization Screen



The second tab of the eCashInit program starts the user token initialization process. Follow the instructions to create the user token. The user token actually carries the monetary value, and needs to be specified during the initialization process. The user token can either be a DS1961S or a DS1963S. Click the “Initialize Now” button to finish creating the user token. See below for screen shot.

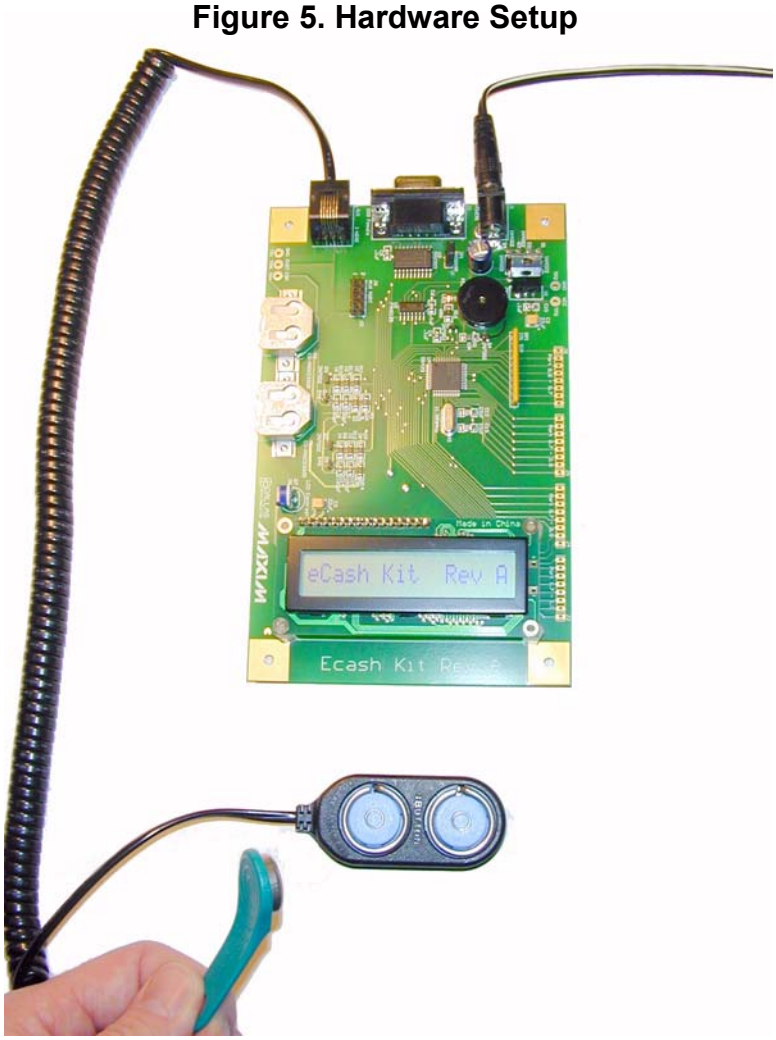
Figure 4. eCashInit User Token Initialization Screen



Hardware Configuration

The hardware setup of the eCash evaluation board requires that at least a coprocessor and a user token be created before beginning (see *Software Configuration*). Once created, plug a coprocessor into the iButton clip of the eCash evaluation board. Next, unplug the DS1402-DR8 Blue Dot from the 1-Wire adapter, and plug it into the RJ11 socket of the eval board. Finally, power up the eCash evaluation board by plugging in the AC adapter. It should look like the picture below.

Figure 5. Hardware Setup



Please note that the user token iButton is easier to handle when snapped into a key fob. Four key fobs come with the kit. If difficulty is encountered when snapping the iButton into the key fob, then please note that the plastic of the key fob can be loosened with warm water enough so that the iButton snaps in easily. Please make an effort to keep water away from the iButton.

The eCash system should now be setup. To debit from the system, just touch the user token iButton to the Blue Dot and watch the debit take place on the LCD screen.

Quick Start

1. Plug in DS9097U to the PC. + Blue Dot to PC
2. Select the DS1963S or DS1961S as the user tokens.
3. Initialize a DS1963S as a token type specific coprocessor.
4. Initialize the user tokens selected in step 2 above.
5. Put the coprocessor into an iButton clip on the eCash board.
6. Plug the Blue Dot into the eCash board.
7. Power up eCash board.
8. eCash board as a power-on default will do a fully autonomous debit.
9. Debit user iButtons, view the result on the LCD.

EVALUATION KIT USAGE

Many experiments can be performed with the evaluation kit. Here are a few suggestions:

Experiment 1:

See *Installation Overview* above.

Experiment 2:

1. Remove DS9097U from the PC.
2. Plug in the straight through serial cable from the PC to the eCash board.
3. Run eCashMonitor program.
4. Change and view the different operation modes of the eCash board.
5. Debit iButtons using various modes of eCash board, monitor status.

Experiment 3:

1. Optionally recycle one of the DS1963S user tokens to make both types of coprocessors. Clip both coprocessors into the eCash board to support both user token types.
2. Debit both the single DS1963S and the DS1961S's with the eCash board, view the results on the LCD.

Experiment 4:
1. Optionally connect to some other host/micro using the RS232 port or development connector and implement the eCash serial protocol.

Experiment 5:
1. Add the firmware loader jumper and do development on the eCash evaluation board.

Experiment 6:
1. Use the firmware source code and design to integrate the eCash functionality into a different board.

FIRMWARE

The firmware running the evaluation board comes pre-loaded. However, we have made it available in the download package, with source, so that developers can extend or modify the functionality of the evaluation board. The firmware was built using the Keil C51 compiler. Both versions 5.10 and 7.05 of the compiler were tested. The makefile builds all the necessary files and links them. Unfortunately, the Keil linker returns an error status after linking when it issues warnings (just warnings about unused segments). Execute 'make' with the '-i' option to ignore this error and convert the binary to the '.hex' file, ready to load on the DS89C420 eCash board.

If the Keil C51 compiler is unavailable to the developer, the pre-built hex file has been included in the download as well.

Loader Instructions

The firmware loader program is a simple command-line serial loader. It was built and tested on Windows using the Cygwin shell. Cygwin provides a POSIX interface (just like Unix/Linux) for serial I/O, so either Cygwin or another Unix to is needed compile and use this application.

Cygwin is available for download from:
http://www.cygwin.com/

To build the loader:
```
gcc -o load420 load420.c
```

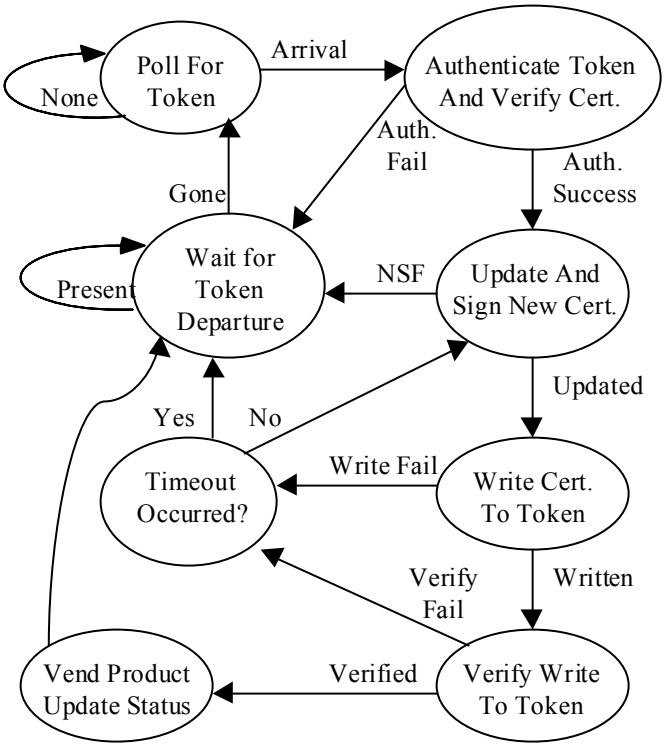
To run the loader:
```
./load420 COM1 ../Firmware/ecash.hex
```

After the loader process completes, the board has all the software ready for performing eCash debits.

Firmware State Diagram

See Figure 6 below for the firmware state diagram.

Figure 6. Firmware State Diagram



HARDWARE SPECIFICATIONS

A description of the evaluation board's hardware components are found in the sections below. Covered items are the Development Connector, the RJ11 "Customer" 1-Wire Interface, the DB9 Serial Interface, the Firmware Load Enable Jumper, and the Power Connector.

Development Connector

IDC (Insulation Displacement Connector) 100 mil spacing.
The development connector can be used to remotely monitor and control the eCash evaluation board.

Figure 7. IDC Connector

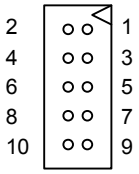


Table 2. IDC Connector Pin Out

PIN	Signal	Description
1	VCC	5V power
2	RESET	reset to the micro
3	COP	coprocessor 1-Wire data
4	RX1	Serial port 1 receive TTL
5	TX1	Serial port 1 transmit TTL
6	INT0	Interrupt 0 on DS89C420
7	INT1	Interrupt 1 on DS89C420
8	P3.4	Port 3 bit 4 input/output
9	CUST	customer 1-Wire data
10	GND	signal ground

RJ-11 1-Wire Interface

See Figure 8 below for the "Customer" 1-Wire Interface. This is the 1-Wire interface designed to be external to the board and accessible by the user or "customer". It provides a place for a DS1402-DR8 Blue Dot to be plugged in thus allowing customers easy access to make contact with their user tokens for debiting purposes.

Figure 8. RJ-11 "Customer" 1-Wire Interface

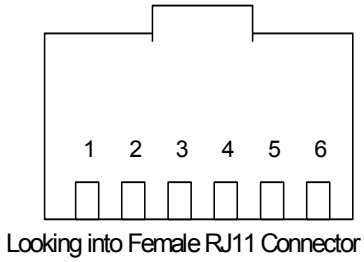


Table 3. RJ-11 "Customer" 1-Wire Interface Pin Out

Pin	Signal name	Description
1	VDD	+5 VDC output
2	GND	Power ground
3	OW	1-Wire Data
4	OW_GND	1-Wire ground return
5	no-connect	
6	no-connect	

DB9 Serial Interface

See Figure 9 and Table 4 below for the DB9 Serial Interface pin out. A standard, straight-through serial cable (included in this kit) can be used to connect the evaluation board to a PC serial port. The above-mentioned eCashMonitor program then can be used to communicate with the board. See *Appendix A: Advanced*. This is the 1-Wire interface designed to be external to the board and accessible by the user or "customer". It provides a place for a DS1402-DR8 Blue Dot to be plugged in thus allowing customers easy access to make contact with their user tokens for debiting purposes.

Figure 9. DB9 Serial Interface

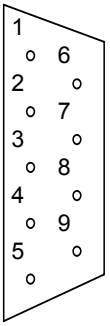


Table 4. DB9 Serial Interface Pin Out

Pin	Signal name	Description
1	no-connect	
2	RX12	RS232 Receive
3	TX12	RS232 Transmit
4	DTR	Data Terminal Ready
5	GND	Ground
6	no-connect	
7	no-connect	
8	no-connect	
9	no-connect	

Figure 10. Firmware Load Enable Jumper

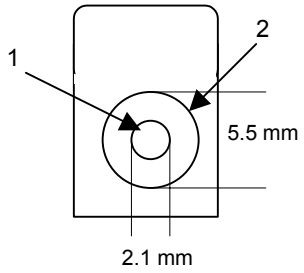
Table 5. Firmware Load Enable Jumper Pin Out

State	Description
JUMPER ON	Firmware loading is controlled by DTR on RS232 serial interface
JUMPER OFF	Firmware loading disabled (default)

Power Connector

Power supply requirements: AC/DC, 9-20 V, 200mA
Recommendations: Stancor Model STA-300R
(Newark Electronics Stock No. 84F2081, Allied
Electronics Stock No. 928-9895)

Figure 11. Power Connector



DSECASH INFORMATION

For more information about the DSECASH, including software downloads, please consult the kit's web page located on our website at http://www.ibutton.com/ibuttons/ecashkit.html.

TECHNICAL SUPPORT RESOURCES

Dallas Semiconductor MAXIM provides many technical support resources for developers. In addition to data sheets, we provide a large number of application notes and white papers, software development tools, and a web-based discussion forum where technical questions can be posted for follow-up.

iButton Product Data Sheets: http://para.maxim-ic.com/iButton.htm

1-Wire Product Data Sheets: http://para.maxim-ic.com/1Wire.htm

Application Notes and White Papers: http://www.maxim-ic.com/appnotes10.cfm/ac_pk/1

Software Development Tools and SDK's: http://db.maxim-ic.com/ibutton/example/

Post technical questions to our web-based discussion forum: http://discuss.dalsemi.com/

SCHEMATICS

See below for some the board schematics. Schematics covered are: Processor LCD and Piezo, Coprocessor and 1-Wire Customer, Power Supply, and the RS232 Serial Interface.

Figure 12. Processor LCD and Piezo

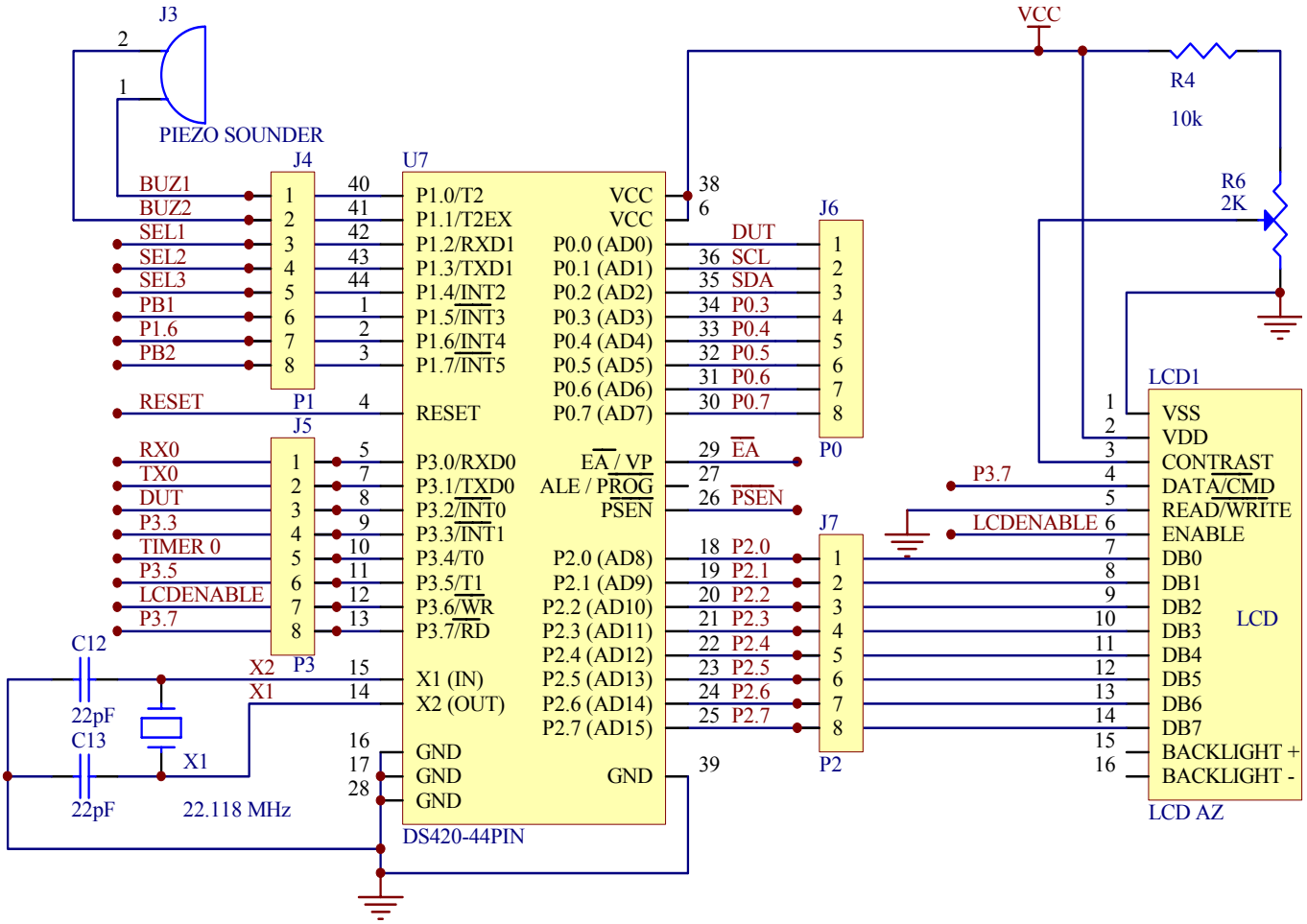


Figure 13. Co-Processor And 1-Wire Customer

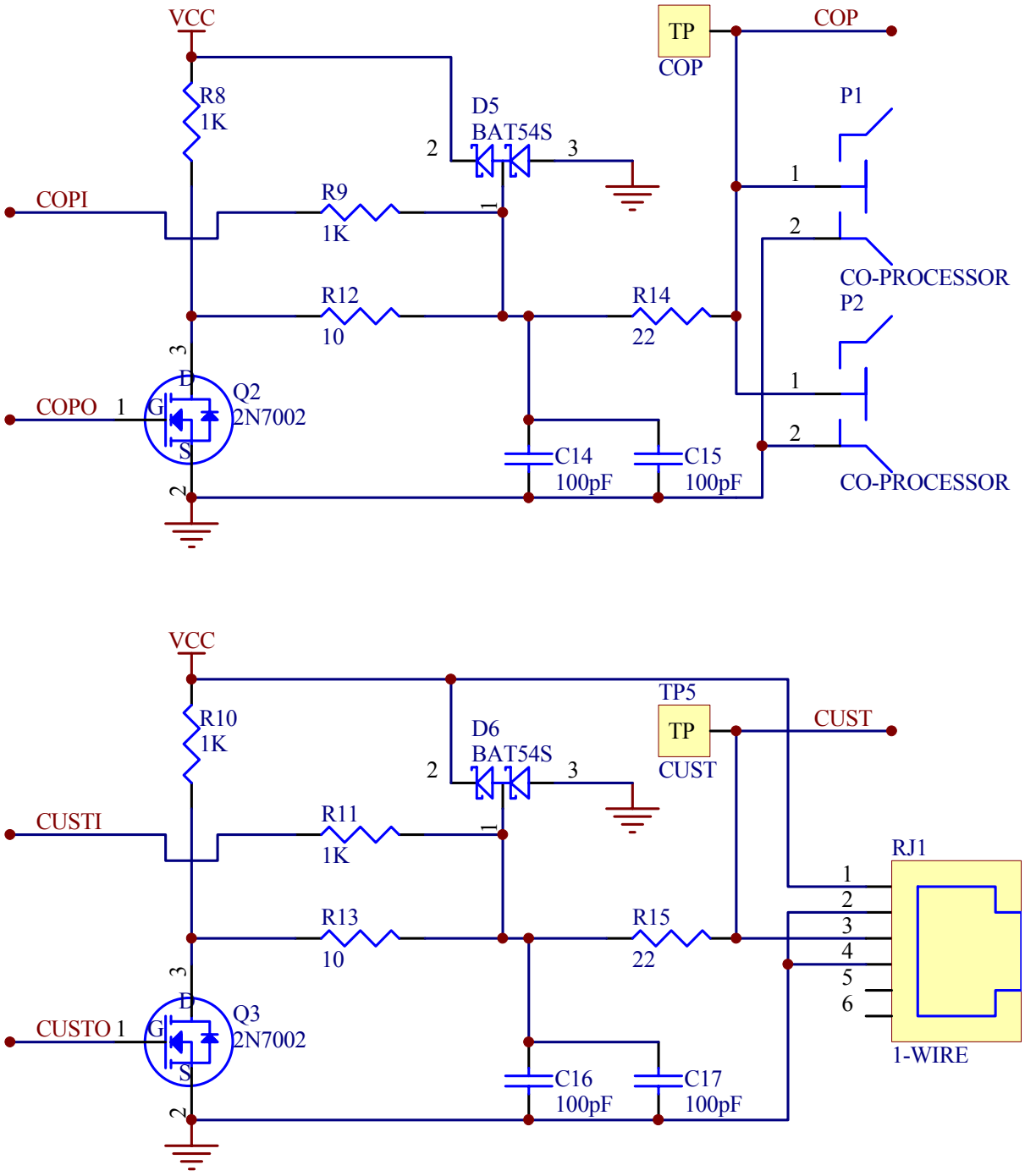


Figure 14. Power Supply

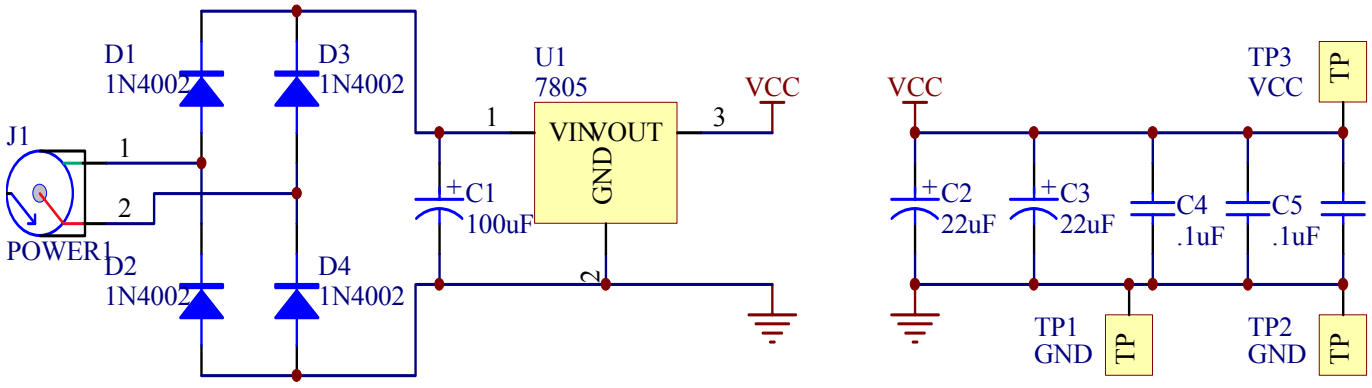
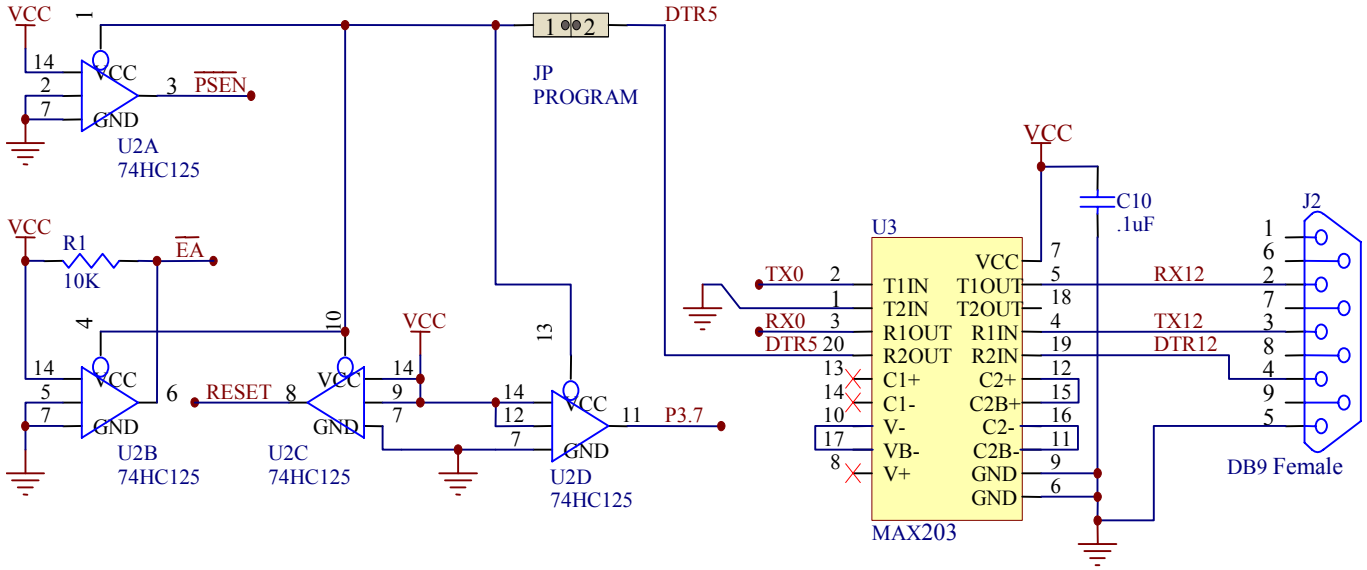


Figure 15. RS232 Serial Interface

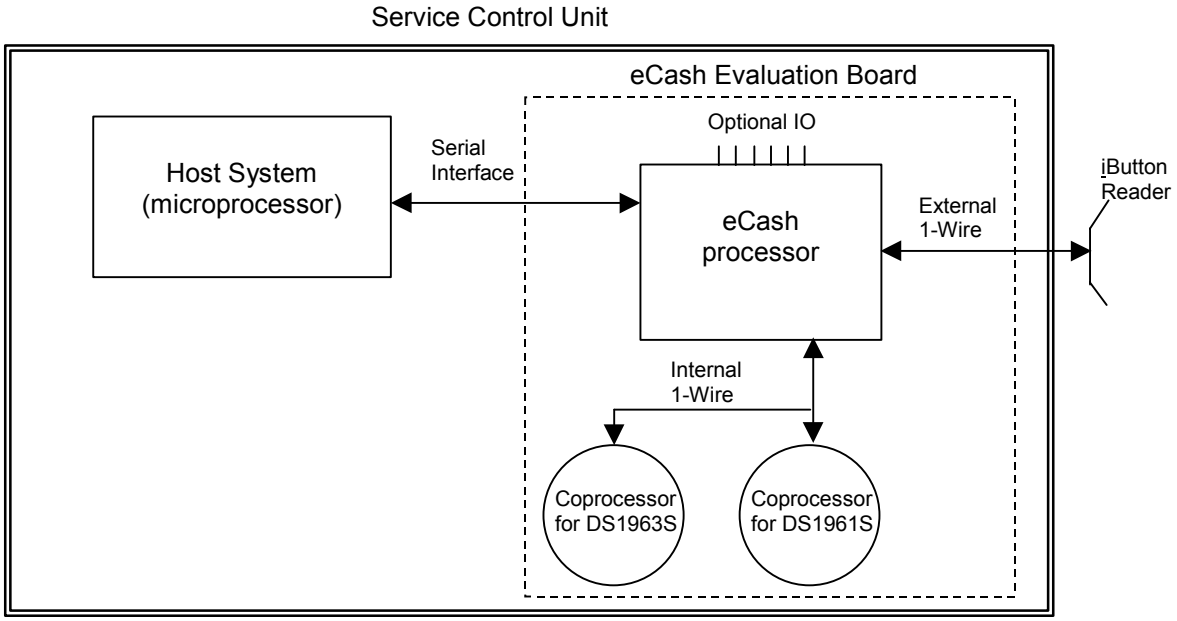


APPENDIX A: THE ECASH PROCESSOR

One of the purposes of the eCash processor is to offload the difficult 1-Wire SHA operations to a dedicated microprocessor. The eCash processor on the eCash evaluation board utilizes the DS1963S in an internal 1-Wire network to perform the SHA-1 authentication and signing of service data. This has speed advantages and also securely keeps the secrets in a stainless steel iButton package.

Alternately, the firmware written in 'C' can be ported to many different processors. There are three main operating modes to the eCash processor: Autonomous, Lock-Step, and Manual. Autonomous mode allows the eCash processor to work without any interaction with the host processor. However, status information will be sent out the serial interface when an event occurs. Lock-Step mode allows the eCash processor to do all of the 1-Wire authentication and/or debiting work but allows the host to approve key steps in the process. Manual mode allows the host to instruct the eCash processor to do any individual operation. The Manual mode commands can also be used while the eCash processor is in Autonomous or Lock-Step mode.

Figure 16. eCash Processor Application Example



Speed considerations: Autonomous is the fastest, Lock-Step is next and full Manual mode is the slowest due to the overhead constraint of serial communication before each 1-Wire operation.

Table 6. Mode Flags

Sequence (and bit #)	Operation Description
0	Automatic 1-Wire polling for user token
1	Automatic read and authentication
2	Automatic debit
3	Automatic Pulse of I/O bit
4	Automatic update of the LCD
5	Enable Overdrive operation
6	not used
7	not used

These autonomous operations are enable or disabled with the manual set command.
The inbound command format has two different types: ‘get’ or ‘set’. The get format provides a single character designator. The set parameter provides a single character designator and also a data field preceded with a length character.

Table 7. Inbound Format

Format	Description
GX	Get command where X is a one character designator
SX<D>	Set command where X is a one-character designator and <D> represents the data payload that the set command requires. If the length of the set command is inconsistent with the format then it will be disregarded.

Example: GF
Get the F parameter data

Example: SQ3ABC
Sets the Q parameter with a three character data field of ABC.

There are two types of outbound messages. The first type is as a result of a get command and provides the one byte designator and data length followed by the data. The second message type is an asynchronous event. The event also has a type designator, length and data.

Table 8. Outbound Format

Format	Description
RXhh<D>	Response of get command designator X with data following of hex length hh
EXhh<D>	Event type X occurred with data following of hex length hh

Example: EQ04ABCD
Event type Q occurred and four characters of data are provided ‘ABCD’.

Example: RQ01Z
Response type Q occurred and one character of data is provided ‘Z’.

Table 9 lists all of the ‘get’ and ‘set’ commands.

Table 9. Get Commands

Command + Format	Description
GD	Get time of last debit
GL	Get the last message sent (response or event).
GP	Read pages(specified by SP) of Coprocessor specified with SN and return
GS	Get status (dump of Memory State Table 6)
GU	Read pages(specified by SU) of user token
GV	Get firmware version
GW	Get ROM of coprocessor, if more then one coprocessor, use SN command to select

Table 10. Set Commands

Command + Format	Description
SAzy	Enables (y=1) or disables (y=0) mode flag (z=0 for AutoPolling, 1 for AutoReading, 2 for AutoDebiting, 3 for Pulsing IO bit, 4 for Update LCD, 5 for Use Overdrive Speed)
SBz	Abort the Lock-Step event (z=0 for Arrival, z=1 for read, z=2 for Debit)
SCy	Enable/Disable (y 1 enable, 0 disable) CRC16 of all packets sent from the eCash processor to the host. The inverted CRC16 is calculated over the entire packet and appended as a 4 character hex value.
SFaaaa	Set service filename aaa (ASCII file name, extension always 102 dec)
SGhhaaa…	Set the LCD to the text message (aaa…) of length hh (hex, max 16). If length N is 0 then clear display.
SIppx	Set the port pin ‘pp’ to the value ‘x’.
SKz	Acknowledge the Lock-Step event X so the next autonomous operation can be done (0 – poll, 1 – read, 2 – debit)
SLxy	Enable/Disable (y=1 enable, y=0 disable) a Lock-Step state. (x=0 for Arrival, 1 for Read, 2 for Debit). Will reset current Lock-Step status
SMhhhh	Set debit amount (hhhh hex number in hundredth of a unit)
SNx	Set the coprocessor number to read pages from (x=0 for DS1963S coprocessor or x=1 for DS1961S coprocessor)
SPhh	Set the coprocessor page to read with the GP command (hh hex page number)
SR	Soft Reset of eCash processor, will get a reset event when complete
SShh	Play beep (Sound) with (hh hex number of beeps)
STxb	Toggle port x bit b, where x and b are 4-bit ascii-encoded hex nibbles.
SUhh	Set the user token page to read (hh page hex number)
SYhh	Set the debit timeout value in hh seconds

Note that the Status State below does not contain all of the state information. State information that can be read directly was omitted (e.g. GV for get version).

Table 11. Status State

Designator	Length (hex)	Example Data (hex)	Description
A	02	01	Autonomous operation mode flags (1 ASCII encoded hex byte) (See Table 1)
C	01	1	CRC16 enable flag (1 enabled, 0 disabled)
D	02	64	Time of last debit in milliseconds
F	04	ABCD	ASCII Service filename
G	10	eCash Demo V0.8	Current LCD text message
L	02	0102	Lock-step acknowledge flags (see Table 7)
M	06	020000	Amount to debit composed of 3 ASCII-encoded hex bytes, representing amount to debit (LSB first)
N	01	0	Coprocessor number selected (0=DS1963S or 1=DS1961S)
P	02	01	Page number to read from the current coprocessor
T	02	32	Port and bit number for the I/O pulse (1 ASCII encoded hex byte)
U	02	08	Page number to read from user Token
W	10	1800000000001122	Coprocessor ROM ID
Y	02	10	Debit timeout in seconds
Z	02	01	Status flags (see Table 8)

Table 12. Acknowledge Flags

Flag (bit position)	Description
0	Ack_PollEnabled
1	Ack_PollWaiting
2	Ack_ReadEnabled
3	Ack_ReadWaiting
4	Ack_DebitEnabled
5	Ack_DebitWaiting
6	Don't Care
7	Don't Care

Table 13. Status Flags

Flag (bit position)	Description
0	Stat_Poll
1	Stat_Read
2	Stat_DebitCert
3	Stat_UpdateToken
4	Stat_Verify
5	Stat_WaitUntilGone
6	Don't Care
7	Don't Care

Table 14. Events

Code	Operation Description	Length + Data Payload Format
R	Soft reset complete	00
A	1-Wire arrival reporting, data is ROM	10RRRRRRRRRRRRRRR
L	1-Wire departure reporting, data is ROM	10RRRRRRRRRRRRRRR
U	Automatic read of select pages from user	40AA55AA55AA55AA55AA55AA55AA55AA55AA55AA55AA55AA55AA55AA55AA55AA55
S	Authenticated raw service data (8 bytes)	10CTCMCMMBMBMBIDID
M	Authenticated monetary balance service data	06MBMBMB
D	Debited final balance	06MBMBMB
I	IO bit pulse after completion of D. (If D is not set then pulse on completion of S or M).	00
C	Status displayed on LCD, length variable	0155
E	Error (see Table 10)	010A

Table 15. Error Codes

Code (hex)	Description
01	No valid token found during poll
02	Tear occurred, the device contents have not changed
03	Tear occurred, the device contents are in an unknown state
04	Tear occurred, but a fix has been applied. Retry verification.
05	Failed to erase coprocessor scratchpad.
06	Failed to find account file
07	Failed to get challenge from coprocessor
08	During Read: Failed to issue challenge to token
09	During Verify: Failed to issue challenge to token
0A	Failed to read token's account data, or data is bad
0B	Failed to compute the UTS
0C	Device failed authentication
0D	Money file signature failed validation
0E	Failed to update the device
0F	Insufficient funds for debit.
10	Failed to erase scratchpad.

Example Responses:

GU:

RU200FAA008039070000444C534D660A0100BEED520001040094CE0000000000E3F

GP:

RP200FAA008063750000434F5052180104004AE1520001040094CEAAAAAAAAAAAAAA

GS: (CR's and spaces added for clarity)

RS63
A02 37
C01 0
D02 00
F04 DLSM
G0F iButton eCash
L02 00
M06 01000
N01 0
P02 00
T02 00
U02 00
W10 18F6AA0200000043
Y02 10
Z02 01

GD:

RD0268