Dogecode White Paper v0.1

Rob Myers rob@robmyers.org

23-12-2014

Overview

Dogecode (name to be finalized) is a system for representing computer programs using sequences of Dogeparty tokens in order to store, send, and execute their code via the Dogecoin blockchain.

Tokens

The encoding for programs is a minor variant of the "Brainfuck" programming language, chosen for its minimalism:

http://en.wikipedia.org/wiki/Brainfuck

Each Brainfuck programming language command is represented by an indivisible Dogeparty asset.

A single occurrence of a command in a program is represented by the sending of a single token, e.g. + is represented by sending 1 INCB tokens .

Multiple occurrences are represented by sending multiple tokens in a single transfer, e.g. +++++ is represented by sending 5 INCB tokens.

The order and quantities of sends are significant, not the total number of tokens held by an address or account.

Currently Used Tokens

Reading bytes into the pointer (the "," command in Brainfuck) isn't currently supported. To enter data, use "+" and ">".

INCP - > (Increment pointer)

DECP - < (Decrement pointer)

INCB - + (Increment byte at pointer)

DECB - - (Decrement byte at pointer)

PUTB - . (Write byte at pointer)

JFOR - [(Jump forward if byte at pointer is not zero)

JBAK -] (Jump back if byte at pointer is not zero)

Reserved Tokens

These tokens are reserved to support future functionality.

GETB - , (Read byte into pointer)

IDAT - (Identify data to be read by READ. See Input/Output below.)

RUNC - (This section of the code is complete, run it.)

BADC - (This section of the code has a problem, do not run.)

Encoding Of Programs

Programs are represented as a sequence of quantities of Dogecode tokens sent from a single address. Order, quantity and sending address are all significant to the system.

Ideally, programs would be represented as multiple sends in a single transaction. Until this is implemented, programs must be sent as an uninterrupted sequence of Dogecode tokens from a single address (no other tokens from the same address may interrupt the send).

Single Program Addresses

Single program addresses represent programs as a newly created Dogeparty address holding only the tokens sent to it in sequence that represent that single program's code.

No additional programs may be stored using that address.

If transferring the tokens fails or the program is found to be incorrect before it runs, a new address must be created and the correct sequence of tokens sent to it.

Program Queue Addresses (Not Yet Implemented)

Programs queued on an address for exection consist of a sequence of quantities of Dogecode tokens sent from a single address.

Since programs cannot be sent as multiple tokens in a single transaction two addresses may send programs at (approximately) the same time, leading to program token sends from two or more addresses becoming interleaved. This requires sorting token sends by address as well as by time when fetching programs to run.

Input/Output

Input Via +/>

Data can be entered into memory using the + and > commands. This is currently the only supported method for entering data into a program.

Input Via IDAT Token (Not Yet Implemented)

Details forthcoming.

Output For Single Program Addresses

Output is determined by running the program locally.

Output For Program Queue Address Runner (Not Yet Implemented)

Output is provided via broadcast messages on the Program Queue Address (details forthcoming).

Running Programs

Single Program Addresses

To execute the program, the system fetches all sends to the address in order and converts their token amounts to runs of Brainfuck commands. It then passes the resulting string to a Brainfuck interpreter. This will provide the final state and output of the program.

Program Queue Address (Not Yet Implemented)

Programs can be sent to an address that you control in order to be run in a queue.

Transactions from other addresses are ignored.

Used tokens can be returned to the sending address or, if the queue address is also the sending address, kept for re-use.

The runner polls the send table for new sends to the queue address. Sequences of tokens sent from a single address constitute programs to be executed. Programs

are terminated with a RUNN token to trigger their execution, or an ERRR token if they should not be executed because (e.g.) a bug was found or the upload was corrupted.

The output of the program can either be recorded locally or output via broadcast messages from the runner address (details forthcoming).

To prevent programs that do not complete in a reasonable amount of time from disrupting the operation of the queue, programs should be run on a thread with a short timeout (details forthcoming).

Program Queue Address As A Service (Not Yet Implemented)

For Dogecode as a service, a program queue address that you control can execute Dogecode programs sent from addresses that others control in return for payment.

Tokens sent to the runner address are not returned, they constitute part of the payment for the service. Additional payment is sent as a Dogecoin transaction following the program tokens.

This system does not use RUNN/ERRR tokens to control execution. Receiving the Dogecoin payment acts as the command to execute the program. Should the payment not be sent within two minutes the program is not run and the tokens are forfeit - unsuccessful or erronious program uploads can have their execution cancelled by not sending payment. Should the incorrect amount of payment be sent, it can be corrected by sending a supplementary payment before any further Dogecode tokens are sent. If a supplement is not sent within two minutes, the tokens and incorrect payment are forfeited to the runner account.

The output of the program is output via broadcast messages from the runner address (details forthcoming).

To prevent programs that do not complete in a reasonable amount of time from disrupting the operation of the queue, programs should be run on a thread with a short timeout. The length of the timeout can be extended by the sender incorporating an additional supplement into the Dogecoin fee that triggers execution (details forthcoming).

Dogecoin prices for running are announced by the runner account on its broadcast message feed (details forthcoming).

Token Transfer Speed

Dogecode programs must be sent as ordered sequences of individual Dogeparty asset “send” commands. To ensure that the token sends are incorporated into the blockchain in order, each send is performed only after the previous one is

confirmed. This means that it can take minutes for each token run to be sent to the receiving account.

Multiple sends from the same account in the same block could be enabled by having multiple inputs available to it, this does not ensure ordering however.

Ideally, Dogeparty would be extended to allow multi-token sends in a single Dogecoin transaction. This would solve both confirmation and ordering.