

MDG Bytesmasher 0.04

Here's an example for a simple bytepacker using 5 different packcodes:

Y-BYTE - packs 0 byte rows with a length of \$03-\$FF bytes to 2 bytes and \$100-\$FFFF bytes to 4 bytes

X-BYTE - packs ? byte rows with a length of \$04-\$FF bytes to 3 bytes and \$100-\$FFFF bytes to 5 bytes

Z-BYTE - packs ? byte rows with a length of \$03 bytes to 2 bytes

V-BYTE - packs 0 byte rows with a length of \$02 bytes to 1 byte

P-BYTE - stores packcode + controlbyte

Purpose of this packer was to handle byte rows larger than \$FF bytes quite well and to optimize and speed up compression using a more complicated approach like huffman.

Written in 1994 by The Human Code Machine / Masters' Design Group.

```

PROCESSOR 6502
;BASIC ZEILE
;ZEROPAGES

ZP EQU $50

DESTLO EQU ZP+$00
DESTHI EQU ZP+$01

YBYTE EQU ZP+$02;0BYTE FOLGEN VON $03-$FF UND $100-$FFFF
XBYTE EQU ZP+$03;XBYTE FOLGEN VON $04-$FF UND $100-$FFFF
ZBYTE EQU ZP+$04;XBYTE FOLGEN VON $03 BYTES
VBYTE EQU ZP+$05;0BYTE FOLGEN VON $02 BYTES
PBYTE EQU ZP+$06;PACKBYTE + CONTROLBYTE

BUF1 EQU ZP+$07
BUF2 EQU ZP+$08

SOURCELO EQU ZP+$09
SOURCEHI EQU ZP+$0A

TESTLO EQU ZP+$0B
TESTHI EQU ZP+$0C

COUNTERLO EQU ZP+$0D
COUNTERHI EQU ZP+$0E

LASTLO EQU ZP+$0F
LASTHI EQU ZP+$10

SOURCELOBUF EQU ZP+$11
SOURCEHIBUF EQU ZP+$12

```

```
ENDLO EQU ZP+$13
ENDHI EQU ZP+$14

;MEMORYS

TESTMEM EQU $FF00
FILE EQU $0701
DESTINATION EQU $FFFF
SOURCE EQU $FEFF
CLEARSTART EQU $0700
FILESTART EQU $0801
DATASTART EQU DEPACKER_END-DEPACKER+$11+$0701
PUT EQU $EDDD
GET EQU $EE13

LOADNAME EQU $06C0
SAVENAME EQU $06E0
LOADLENGTH EQU $06DF
SAVELENGTH EQU $06FF
PROGSTARTLO EQU $06BE
PROGSTARHI EQU $06BF
SEIFLAG EQU $06BD
IOVALUE EQU $06BC

ORG $0200
;-----
LOADFILE SUBROUTINE
LDA #$01
LDX #$08
LDY #$00
JSR $FFBA

LDA LOADLENGTH
LDX #<LOADNAME
LDY #>LOADNAME
JSR $FFBD
JSR $FFC0

LDX #$01
JSR $FFC6

JSR GET
STA SOURCELO ;STARTADDRESS LOW
STA SOURCELOBUF
JSR GET
STA SOURCEHIBUF
TAX
DEX
STX SOURCEHI ;STARTADDRESS HIGH -1 (SECURITY BUFFER)
```

```

.NEXTBYTE  LDA $90
           BNE .EXIT
           JSR GET
           STA $D020
           SEI
           LDY #$00
           STY $D020
           STY $01

           STA (SOURCELO),Y
           TAX
           LDA TESTMEM,X
           CMP #$FF
           BEQ .OVER
           INC TESTMEM,X

.OVER     INC SOURCELO
           BNE .OVER1
           INC SOURCEHI
.OVER1    LDX #$37
           STX $01
           BNE .NEXTBYTE
.EXIT     JMP EXITLOAD

```

```

;-----
ORG $0259
DC.B $00,$00,$00,$00,$00,$00,$00,$00
DC.B $00,$00,$00,$00,$00,$00,$00,$00
DC.B $00,$00,$00,$00,$00,$00,$00,$00
DC.B $00,$00,$00,$00,$00,$00,$00,$00

```

```

;-----
CLEARMEMORY SUBROUTINE

```

```

           LDA #<CLEARSTART
           STA DESTLO
           LDA #>CLEARSTART
           STA DESTHI

           LDA #$34
           STA $01
.1        LDY #$00
           TYA
.2        STA (DESTLO),Y
           INY
           BNE .2

           INC DESTHI
           LDA DESTHI
           BNE .1

           LDA #$FF

```

```
    STA TESTMEM
    LDA #$37
    STA $01
    RTS

    ORG $02A1
    DC.B $00

;-----
MAIN2  SUBROUTINE
    JSR CLEARMEMORY
    JSR LOADFILE

    LDA #$34
    STA $01
    JSR TESTBYTES

    JSR PACK
    JSR MAKEPACKER
    LDA #$37
    STA $01

.2  LDA #$00
    STA $D020

.3  LDA $D020
    EOR #$0B
    STA $D020

    LDA $DC01
    CMP #$EF
    BNE .3
    JSR SAVEFILE

    JMP .2

;-----
TESTBYTES  SUBROUTINE
    LDY #$00

.TEST  LDX #$FF
    TXA
.1  CMP TESTMEM,X
    BCC .KLEINER ;VERZWEIGT WENN CMP KLEINER
    LDA TESTMEM,X
    STX .3+1
.KLEINER  DEX
    BNE .1
.3  LDX #$00
    LDA #$FF
    STA TESTMEM,X
```

```
STX YBYTE, Y
INY
CPY #$05
BNE .TEST
RTS

;-----
ORG $0314
;-----
---
ORG $0334

PACK    SUBROUTINE
LDA SOURCELOBUF
SEC
SBC #$02
STA ENDLO
LDA SOURCEHIBUF
SBC #$01
STA ENDHI

LDA SOURCELO
STA PUTIT+1+$11
SEC
SBC #$01
STA SOURCELO
LDX SOURCEHI
TXA
INX
STX PUTIT+2+$11
SBC #$00
STA SOURCEHI

LDX #$FF
STX DESTLO
STX DESTHI

.1 JSR LOADBYTE
BCS .END

INC $01
INC $D020
DEC $01

JSR CHECKBYTE
JMP .1

.END    LDA ZBYTE
        JSR STOREBYTE
        LDA #$00
        JMP STOREBYTE
```

```
;-----  
CHECKBYTE  SUBROUTINE  
    STA BUF1  
    CMP #$00  
    BEQ .TESTY  
    JMP .TESTX  
  
;-----  
  
.TESTY  JSR SETTEST      ;+SETCOUNTER  
  
.1  JSR GETTEST  
    BCS .2  
    CMP #$00  
    BNE .2  
    JSR INCCOUNTER  
    JMP .1  
  
.2  LDA COUNTERHI  
    BNE .SETYHI  
  
    LDA COUNTERLO  
    CMP #$02  
    BEQ .SETYTWO  
    CMP #$01  
    BNE .SETYLO  
    JMP .STOREONE  
  
.SETYHI LDA YBYTE  
    JSR STOREBYTE  
    LDA #$00  
    JSR STOREBYTE  
    LDA COUNTERHI  
    JSR STOREBYTE  
    LDA COUNTERLO  
    JSR STOREBYTE  
    JMP .ENDCHECK  
  
.SETYTWO  LDA VBYTE  
    JSR STOREBYTE  
    JMP .ENDCHECK  
  
.SETYLO  LDA YBYTE  
    JSR STOREBYTE  
    LDA COUNTERLO  
    JSR STOREBYTE  
    JMP .ENDCHECK  
  
;-----  
  
.TESTX  JSR SETTEST      ;+SETCOUNTER
```

```
.3 JSR GETTEST
   BCS .4
   CMP BUF1
   BNE .4
   JSR INCCOUNTER
   JMP .3

.4 LDA COUNTERHI
   BNE .SETXHI

   LDA COUNTERLO
   CMP #$04
   BCS .SETXLO
   BEQ .SETXLO

   CMP #$03
   BEQ .SETXTHREE
   JMP .STOREONE

.SETXHI LDA XBYTE
        JSR STOREBYTE
        LDA #$00
        JSR STOREBYTE
        LDA COUNTERHI
        JSR STOREBYTE
        LDA BUF1
        JSR STOREBYTE
        LDA COUNTERLO
        JSR STOREBYTE
        JMP .ENDCHECK

.SETXTHREE LDA ZBYTE
.SETROW3B JSR STOREBYTE
          LDA BUF1
          JSR STOREBYTE
          JMP .ENDCHECK

.SETXLO LDA XBYTE
.SETROW2 JSR STOREBYTE
         LDA COUNTERLO
         JSR STOREBYTE
         LDA BUF1
         JSR STOREBYTE
         JMP .ENDCHECK

.STOREONE LDA BUF1
          TAY
          LDX #$04

.9 CMP YBYTE,X
   BNE .OVER
```

```
LDA PBYTE
JSR STOREBYTE
TYA
.OVER DEX
BPL .9
.ENDP JSR STOREBYTE
JSR SETCOUNTER

.ENDCHECK INC SOURCELO
BNE ADDCOUNTER
INC SOURCEHI

;-----
ADDCOUNTER SUBROUTINE
LDA SOURCELO
SEC
SBC COUNTERLO
STA SOURCELO
LDA SOURCEHI
SBC COUNTERHI
STA SOURCEHI
RTS

;-----
GETTEST SUBROUTINE
STY .1+1
LDY #$00
LDA (TESTLO),Y
PHA
.1 LDY #$00

LDA TESTLO
SEC
SBC #$01
STA TESTLO
BCS .2
DEC TESTHI
.2 LDA TESTHI
CMP ENDHI
BNE .OVER
LDA TESTLO
CMP ENDLO
BNE .OVER
PLA
SEC
RTS

.OVER PLA
CLC
RTS

;-----
```


SETTEST SUBROUTINE

```
PHA
LDA SOURCELO
STA TESTLO
LDA SOURCEHI
STA TESTHI
PLA
```

SETCOUNTER SUBROUTINE

```
STX .1+1
LDX #$00
STX COUNTERHI
INX
STX COUNTERLO
.1 LDX #$00
RTS
```

;-----
INCCOUNTER SUBROUTINE

```
INC COUNTERLO
BNE .1
INC COUNTERHI
.1 RTS
```

;-----
STOREBYTE SUBROUTINE

```
PHA
STY .1+1
LDY #$00
STA (DESTLO),Y
.1 LDY #$00
LDA DESTLO
SEC
SBC #$01
STA DESTLO
BCS .2
DEC DESTHI
.2 PLA
RTS
```

;-----
LOADBYTE SUBROUTINE

```
STY .1+1

LDY #$00
LDA (SOURCELO),Y
PHA

.1 LDY #$00
LDA SOURCELO
SEC
SBC #$01
STA SOURCELO
BCS .2
```

```
DEC SOURCEHI

.2 LDA SOURCEHI
   CMP ENDHI
   BNE .3
   LDA SOURCELO
   CMP ENDLO
   BNE .3
   PLA
   SEC
   RTS

.3 PLA
   CLC
   RTS

;-----
SAVEFILE      SUBROUTINE
LDA SAVELENGTH
LDX #<SAVENAME
LDY #>SAVENAME
JSR $FFBD

LDA #$01
TAY
LDX #$08
JSR $FFBA
JSR $FFC0
LDX #$01
JSR $FFC9

LDA #<FILESTART
STA SOURCELO
JSR PUT
LDA #>FILESTART
STA SOURCEHI
JSR PUT
SEI

.NEXTBYTE    LDY #$00
             STY $D020
             STY $01

LDA (SOURCELO),Y

LDY #$37
STY $01

STA $D020

JSR PUT
```

```
SEI

INC SOURCELO
BNE .NEXT
INC SOURCEHI

.NEXT  LDA SOURCELO
      CMP DESTLO
      BNE .NEXTBYTE
      LDA SOURCEHI
      CMP DESTHI
      BNE .NEXTBYTE

;-----
EXITLOAD  JSR $FFCC
          LDA #$01
          JSR $FFC3
          SEI
          RTS

;-----
MAKEPACKER SUBROUTINE
          LDA DESTLO
          CLC
          ADC #$01
          STA SOURCELO
          LDA DESTHI
          ADC #$00
          STA SOURCEHI

          LDA #<DATASTART
          STA DESTLO
          LDA #>DATASTART
          STA DESTHI

          LDY #$00
.1      LDA (SOURCELO),Y
          STA (DESTLO),Y
          INC SOURCELO
          BNE .OVER1
          INC SOURCEHI
.OVER1  INC DESTLO
          BNE .OVER2
          INC DESTHI
.OVER2  CPY SOURCELO
          BNE .1
          CPY SOURCEHI
          BNE .1

          LDA PROGSTARTLO
          STA STARTADDRESS+1+$11
```

```
LDA PROGSTARHI
STA STARTADDRESS+2+$11

LDA SEIFLAG
STA FLAG+$11

LDA IOVALUE
STA IOVAL+1+$11

LDA DESTLO
STA DEPACKER+1+$11
STA GETIT+1+$11

LDA DESTHI
STA DEPACKER+2+$11
STA GETIT+2+$11

LDY #$00
.2 LDA DATASTART,Y
STA (DESTLO),Y
INY
BNE .2

LDX #$00
.4 LDA YBYTE,X
STA YBYTE2+1+$11,Y
INY
INY
INY
INY
INX
CPX #$05
BNE .4

LDY #$00
.3 LDA $0100,Y
STA FILESTART,Y
INY
CPY #DEPACKER_END-DEPACKER+$11
BNE .3

INC DESTHI

RTS

REND

ORG $0700
;-----
---
```

```
MAIN    SUBROUTINE
```

```
SEI
CLD
LDA #$37
STA $01

LDX #$FF
TXS
INX
STX $D011
STX $D020
STX $D021
STX $D418

LDY #$1F
.A LDA $FD30,Y
STA $0314,Y
DEY
BPL .A

LDA #$7E
STA $0314

INY

.B LDA DATEN,Y
STA $0100,Y
INY
BNE .B
JMP MAIN2

;-----
;DEPACKER
;-----
DATEN
    RORG $0801

DEZIP  SUBROUTINE
    SEI
    INC $01
    LDY #DEPACKER_END-DEPACKER
.1 LDA PACKER-1,Y
STA DEPACKER-1,Y
DEY
BNE .1
JMP DEPACKER
PACKER
    REND

    RORG $0100
```

```
;-----  
;DECRUNCHER ROUTINEN  
;-----  
DEPACKER  SUBROUTINE  
.1  LDA $FFFF,Y  
    STA DATASTART,Y  
    INY  
    BNE .1  
  
;-----  
DECRUNCH  SUBROUTINE  
    LDX #$01  
    INY  
    JSR GETBYTE  
YBYTE2  CMP #$00  
    BEQ DECY  
XBYTE2  CMP #$00  
    BEQ DECX  
ZBYTE2  CMP #$00  
    BEQ PUTZ  
VBYTE2  CMP #$00  
    BEQ DECV  
PBYTE2  CMP #$00  
    BNE PUTREST  
  
DECP     JSR GETBYTE  
  
PUTREST  PHA  
    LDA PUTIT+1  
    BNE .OVER  
    DEC PUTIT+2  
.OVER   DEC PUTIT+1  
    PLA  
PUTIT   STA $FFFF  
  
    DEX  
    BNE PUTREST  
    DEY  
    BEQ DECRUNCH  
  
;-----  
DECY     SUBROUTINE  
;YBYTE  0BYTE FOLGEN VON $03-$FF UND $100-$FFFF  
    JSR GETBYTE      ;GETLENGTH  
    BNE .DEPACK  
  
.DECLONG  JSR GETBYTE      ;GET HI  
    TAY  
    JSR GETBYTE      ;GET LO  
  
.DEPACK  TAX  
.DOV     LDA #$00
```

```

    BEQ PUTREST

DECV   INX
    BNE .DOV

;-----
DECX   SUBROUTINE
;XBYTE XBYTE FOLGEN VON $04-$FF UND $100-$FFFF
    JSR GETBYTE   ;ANZAHL LO ODER 0
    BNE .DEPACK
    JSR GETBYTE   ;GET HI
    TAY
    JSR GETBYTE   ;GET LO
.DEPACK TAX
    JSR GETBYTE
    BNE PUTREST
;-----
PUTZ   LDX #$03
    JSR GETBYTE
    BNE PUTREST

ENDE   SUBROUTINE
IOVAL  LDA #$37
    STA $01
FLAG   SEI
STARTADDRESS JMP $9000
;-----
GETBYTE SUBROUTINE
    LDA GETIT+1
    BNE .OVER
    DEC GETIT+2
.OVER  DEC GETIT+1
GETIT  LDA $FFFF
    RTS
;-----
DEPACKER_END
    REND

```

From: <https://codebase64.org/> - **Codebase 64 wiki**

Permanent link: https://codebase64.org/doku.php?id=base:mdg_bytesmasher_0.04

Last update: **2015-04-17 04:32**

