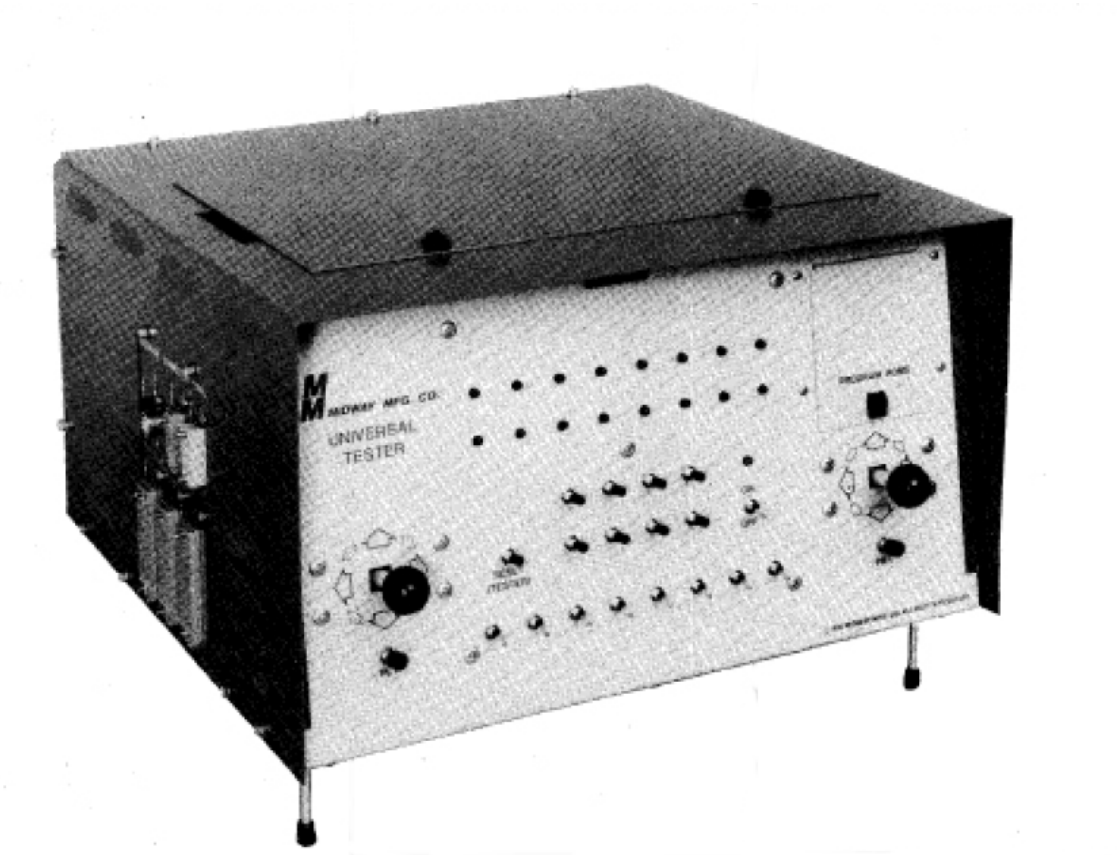
Midway Universal Tester

Test ROM Programming Guide



Version 1.0.2
August 2005

Revision History

1.0.0	Release
1.0.1	Formatting Updates
1.0.2	Formatting Updates, TOC Added, Samples moved to appendices

Table of Contents

Purpose:

To describe the layout of the test ROMs for the Midway Universal Tester system in sufficient enough detail to allow for editing of existing test ROMs and creation of new test ROMs.

Contact Information:

The original version was authored by, and is still maintained by James Marous. Email jmarous@woh.rr.com.

Scope:

This document covers basic input/output reading and writing, bit masking, bit shifting operations, tester setup defaults and encoder emulation functions of the test ROM. This document deals with version 2.0 of the test ROM (Marked UTEX20A6 on the ROM label).

To be done:

Continue documentation of encoder emulation.

Introduction:

1. ROM layout:
    a. Checksum header.
    b. Internal ROM address table.
    c. Individual subroutine data.
    d. Executable setup code.

2. Tester interface to ROM:
    a. Tester memory maps the test ROM starting at memory location 2000H. Test ROM 0 lies from 2000H to 2FFFH and Test ROM 1 lies from 3000H to 3FFFH.
    b. For the purposes of the remainder of this document, the ROM will be described in terms of its memory mapped address as visible by the Tester. To convert to the actual address of just the ROM, subtract 2000H from the address.

3. Checksum Header:
    a. First 5 bytes of test ROM contain the data used by the tester to checksum the ROM.

ROM Address	Tester Mapped Address	Data
0000	2000	00
0001	2001	20
0002	2002	00
0003	2003	08
0004	2004	AA

    b. Bytes 2000H and 2001H contain the start address to checksum, in this example 2000H.

c. Bytes 2002H and 2003H contain the # of memory locations to checksum, in this example 0800H locations.
d. Byte 2004H contains the checksum to compare to.
e. Checksum is computed by a logical OR of the current memory location to the previous (starting with 00H) ignoring any carries.

4. Start of Test ROM data:
   a. Test ROM data starts at memory location 202DH as pointed to by ignored header bytes from 2008H to 202CH.

5. Address table:
   a. Starting at 202D, a list of 2 byte addresses follows, until two 00H bytes are encountered. For the PacMan test Rom, the table is:

202d	37
202e	20
202f	39
2030	20
2031	44
2032	20
2033	55
2034	20
2035	00
2036	00

   b. This table decodes into 4 addresses within the test ROM, 2037H, 2039H, 2044H and 2055H. These addresses are copied into the tester's ROM starting at memory locations 401AH.
   c. Each address represents a subroutine address held in the test ROM.
   d. A maximum of 24 addresses (48 bytes) can be stored in this header table.

6. The first address, in this example 2037H, points to another 2 byte address which holds the address of the setup executable section of the test ROM.

2037	67
2038	20

   a. In this case, 2067H is where the executable code starts.

7. The remaining addresses in the table each point to a section of code data in the test ROM.

2039	C1
203A	0b
203B	01
203C	01
203D	05
203E	44
203F	FE
2040	0A
2041	01
2042	00

2043	00
2044	5E
2045	0C
2046	01
2047	0D
2048	00
2049	01
204A	44
204B	00
204C	0f
204D	00
204E	FF
204F	00
2050	44
2051	00
2052	03
2053	FC
2054	00

2055	5E
2056	0c
2057	01
2058	0B
2059	00
205A	02
205B	44
205C	00
205D	FF
205E	00
205F	0C
2060	00
2061	04
2062	44
2063	00
2064	FF
2065	00
2066	00

8. The first two bytes of each code section contain the address of a subroutine located in the tester's internal ROM to run. For this example, there are 2 distinct subroutines to run, 0BC1H and 0C5EH (run twice).

0BC1H Subroutine.

2039	C1
203A	0B
203B	01
203C	01
203D	05
203E	44
203F	FE
2040	0A
2041	01
2042	00
2043	00

1. This subroutine handles reading and writing to the LEDs on the front panel of the tester.
2. For this example, let's call address 2039H offset # 0 into this subroutine.
3. Offsets 4 and 5 (203DH and 203EH) are the address of the LEDs to write data to (stored in IY).
4. Offset 6 (203FH) is the default value to write to the LEDs (FEH).
5. Offset 7 (2040H) is the GPIO byte to read from, 0AH = 10 Decimal = 4410H . (0BH = 4411H, etc).
6. Offset 8 (2041H) is the bit mask to mask the data after it is read, this value (01H in this example) is logical ANDd with the read data before writing to the LEDs.
7. Offsets 9 contains the bit shifting instructions (if any). Data is shifted right or left after being read. Values greater than 80H are left shifts, and values less are right shifts. 01H means shift one right, 02H means shift 2 right, etc. FFH = 1 shift left, FBH= 5 shift left, FAH = 6 shift left, F9H = 7 shift left.
8. Offset 10 (2043H) contains a 00H byte to indicate the end of this code section.

0C5E Subroutine

2044	5E
2045	0C
2046	01
2047	0D
2048	00
2049	01
204A	44
204B	00
204C	0F
204D	00
204E	FF
204F	00
2050	44
2051	00
2052	03

2053	FC
2054	00

1. Used for general reading/writing (front panel reads to GPIO writes):
    a. May write to one GPIO address for multiple.
    b. May read from single or multiple front panel controls.
2. Bytes 0 and 1: address of subroutine in tester ROM to run.
3. Offset 3 (2047H in this example) is the first GPIO address to write to:
    a. 0DH = 4413H = GPIO D0-D7.
    c. 0CH = 4412H = GPIO C0-C7.
    d. 0BH = 4411H = GPIO B0-B7.
    e. 0AH – 4410H = GPIO A0-A7.
4. Offsets 5 and 6 (2049H and 204AH in this example) contains the front panel address to read from (4401H).
5. Bytes 7, 8 and 9 are bit handling for the data:
    a. Byte 7 is loaded into accumulator A and then or'd with itself before the data is read into the accumulator.
    b. Byte 8 is a bit mask for the read data, data is read from the front panel control and then logically ANDd to this value before being stored or written to the output port.
    c. Bit 9 is a bit shifting mask, data is shifted right or left after being read. Values greater than 80H are left shifts, and values less are right shifts. 01H means shift one right, 02H means shift 2 right, etc. FFH = 1 shift left, FBH= 5 shift left, FAH = 6 shift left, F9H = 7 shift left.
6. Finally, Bit 10 or the 4th byte after the address:
    a. If FFH, do not change output address, read another input.
    b. If 00H, end of code segment.
    c. If other value, change output port to value, i.e. if 0AH output = 4410H, if 0BH, output = 4411H, etc.

7. Example #1 (from PacMan test ROM):

2055	5E	Subroutine address (0C5EH)
2056	0C	
2057	01	
2058	0B	Output address 4411H (GPIO B0-B7)
2059	00	
205A	02	Input 4402H (Buttons PR1-5)
205B	44	
205C	00	Load into accumulator, OR'd w/ self
205D	FF	Mask bits w/ FFH (pass all bits)
205E	00	No shifting
205F	0C	Output address 4413H (GPIO D0-D7)
2060	00	End First Segment
2061	04	Input 4404H (Buttons PL1-5)
2062	44	
2063	00	Clears accumulator

2064	FF	Mask w. FFH (Pass all bits)
2065	00	No bit shifting
2066	00	End Segment

8. Example #2 (from Gorf test ROM):

2051	5E	Subroutine address 0C5EH
2052	0C	
2053	01	
2054	0B	Use address 4413H (GPIO D0-D7)
2055	00	
2056	01	Input address 4401H (Buttons PB1-8)
2057	44	
2058	00	Clears accumulator A
2059	1F	Mask, pass only bits 0-4
205a	00	no shift
205b	FF	no increment of output address
205c	00	Input address 4400H, (Switches S1-S8)
205d	44	
205e	00	Clear accumulator A
205f	01	Mask, only pass bit 0
2060	FB	Shift 5 to the left
2061	FF	no increment of output address
2062	04	Input address 4404H, (Buttons PL1-5)
2063	44	
2064	00	Clear accumulator A
2065	01	Mask, only pass bit 0
2066	FA	Shift 6 to the left
2067	FF	no increment of output address
2068	02	Input address 4402H, (Buttons PR1-5)
2069	44	
206a	00	Clear accumulator A
206b	01	Mask, only pass bit 0
206c	F9	Shift 7 to the left
206d	00	end, go ahead and write to output address (4413H)

0EA4H and 02C7H Subroutines (Encoder Emulation)

1. Reads from RAM, writes to Port.
2. First segment – output writing subroutine.
3. Next segments, input reading and data translation subroutine.

0EA4 Subroutine

2078	A4	Output writing function, writes data from RAM to a port
2079	0E	
207a	00	
207b	11	GPIO B0-B7
207c	44	Output port (also saved to 4002H)
207d	93	
207e	40	Data read from this RAM loc (4093H)
207f	00	Clear Accum A
2080	FF	mask (and'd w/ 4093H) {Pass all bits}
2081	00	shift bits {No shifting here}
2082	00	End of Segment

1. Writes data to a port based on data in a specified RAM location.
2. Bytes 1 and 2, subroutine address (0EA4H).
3. Bytes 4 and 5, Output port to send data to (4411H).
4. Bytes 6 and 7, RAM address holding data to write (4093H).
5. Bytes 8, 9 and 10. Same as 0C5EH subs, first is clear accumulator, then read mask and finally bit shifting.
6. Last byte is 00H to indicate end of this segment.
7. This segment's address in the test ROM is referred to in the 02C7H subroutine.

0C27H Subroutine

2083	C7	encoder emulation executable subroutine address
2084	02	
2085	02	Push Buttons PR1-PR5
2086	44	
2087	04	Panel command: left, medium speed (increment)
2088	10	Panel command: right, medium speed (decrement)
2089	1E	read mask, only allowing 4 joystick bits through
208a	1E	read mask, only allowing 4 joystick bits through
208b	00	If bit 7 of IY+4 is 1, add this to byte
208c	00	
208d	00	If bit 7 of IY+4 is 0, add this to byte
208e	00	
208f	02	Speed byte (# of major interrupts), if command bit 0 is 0
2090	00	

2091	00	Value to load into (IY+3) when byte reaches 0
2092	0B	Command byte (0000 1011)
2093	A0	Ram Loc to store data (40A0H)
2094	40	
2095	78	Output writing function address (2078H)
2096	20	
2097	00	Address loaded into (DE)
2098	00	

1. Bytes 1 and 2, subroutine address.
2. Bytes 3 and 4, address of front panel control to read data from.
3. Bytes 5 and 6, input compare bytes, data read from control is first compared to these bytes, used to set flags in IY+4 to set increment/decrement flag.
4. Bytes 7 and 8, input mask, data read from control is ANDd with these bytes before processing.
5. Bytes 9 and 10 are used if increment by byte command and an increment flag.
6. Bytes 11 and 12, are used if decrement by byte command and a decrement flag.
7. Bytes 13 and 14, are the speed byte, equates to # of minor interrupts.
8. Byte 15 is the starting value of (IY+3) {generally 406EH}, also loaded when (IY+3) reaches 00H.
9. Byte 16 is the command byte:
    a. Bit 0 signals to decrement (IY+3) {generally 406DH}:
        1. If (IY+3) reaches zero, load (IY+2) byte 15.
    b. Bit 1 signals increment or decrement based on bytes 9 through 12 (doesn't work in 2.0 release, but I have a ROM version where this is fixed).
    c. bit 3 signals to load the byte data from a table (works).
    d. Bit 4 signals to use the table at 02FDH if 1, 03FFH is 0.
10. Bytes 17 and 18, Ram location to store data in (40A0H).
11. Bytes 19 and 20, address of output writing subroutine in test ROM.
12. Byte 21, 00H to indicate end of code segment.

Executable setup code

1. Runs once at beginning of main program, loads defaults into I/O VREF and Load Control registers.
2. Compiled, executable Z80 subroutine.

Example #1 (PacMan)

1. Machine code

2067	3e
2068	ee
2069	32
206a	15
206b	44
206c	32
206d	04
206e	40
206f	3e
2070	fe
2071	32
2072	16
2073	44
2074	3e
2075	ff
2076	32
2077	17
2078	44
2079	c9

2. Decompiled Code

```
;First program I/O VREF Controls
0067 3E EE          LD A,0EEH ;Load A w/ EEH
0069 32 15 44       LD (4415H),A ;Write EEH to (4415H)
006C 32 04 40       LD (4004H),A ;Save A in RAM @ 4004H
;Now program Load Control A0-A7
006F 3E FE          LD A,0FEH ;A = FEH
0071 32 16 44       LD (4416H),A ;(4416H) = FEH
;Now program Load Control B0-B7
0074 3E FF          LD A,0FFH ; A = FFH
0076 32 17 44       LD (4417H),A; (4417H) = FFH
0079 C9             RET ;Done
```

Example #2 (Gorf)

1. Machine Code

```
3e
fc
32
15
44
32
04
40
3e
ff
32
16
44
32
17
44
32
18
44
32
1a
44
3e
f9
32
19
44
c9
```

2. Decompiled Code

```
LD      A,0FCH      A = FCH
LD      (4415H),A   (4415H) = FCH (I/O VREF)
LD      (4004H),A   (4004H) = FCH (RAM)
LD      A,0FFH      A = FFH
LD      (4416H),A   (4416H) = FFH (Load Control A)
LD      (4417H),A   (4417H) = FFH (Load Control B)
LD      (4418H),A   (4418H) = FFH (DR Bus Interface)
LD      (441AH),A   (441AH) = FFH  (Audio Enable)
LD      A,0F9H      A = F9H
LD      (4419H),A   (4419H) = F9H (Video Enable)
RET
```

Executable Code Details

1. I/O VRef Control (4415) for GPIO A – D:
    a. Bits 0 to 3, Port direction (to/from tester), 0 = input, 1 = output.
        i. Bit 0 = 4410H = GPIOA.

ii. Bit 1 = 4411H = GPIOB.
iii. Bit 2 = 4412H = GPIOC.
iv. Bit 3 = 4413H = GPIOD.

b. Bits 4 to 7 control the reference voltage that the I/O lines are pulled up to (through resistors):
   i. Bit 4 = GPIOA and DRA.
   ii. Bit 5 = GPIOB and DRB.
   iii. Bit 6 = GPIOC.
   iv. Bit 7 = GPIOD .
   v. Writing a zero pulls a line to VRef1, writing a 1 pulls to either +5V or VRef2 depending on jumpers on the I/O card.
   vi. The two possible reference voltage values are VRef1 and VRef2 at pins 33 and 34 of the backplane bus.

2. I/O VRef Control (4435H) for GPIO E-H:
   a. Bits 0 to 3, Port direction (to/from tester), 0 = input, 1 = output.
      i. Bit 0 = 4430H = GPIOE.
      ii. Bit 1 = 4431H = GPIOF.
      iii. Bit 2 = 4432H = GPIOG.
      iv. Bit 3 = 4433H = GPIOH.

   b. Bits 6 AND 7 control the reference voltage that the I/O lines are pulled up to (through resistors):
      i. Bit 6 = GPIOG.
      ii. Bit 7 = GPIOH.
      iii. Writing a zero pulls a line to VRef1, writing a 1 pulls to either +5V or VRef2 depending on jumpers on the I/O card.
      iv. The two possible reference voltage values are VRef1 and VRef2 at pins 33 and 34 of the backplane bus.

3. Load Control (4416H and 4417H):
   a. Addresses:
      i. 4416H = GPIO A0-A7.
      ii. 4417H = GPIO B0-B7.
   b. Usage:
      i. Bits written determine the default state of the I.O port. i.e. writing a 1 means I/O line is normally high, writing a 0 means line is normally low. (PacMan sets A0 low so that the coin counter LED is normally on).

4. Video Output Enable 4419H.:
   a. writing a 1 on bit 0 enables video output (B-Y,R-Y format to RGB decode).

5. Audio Output Enable 441AH, (writing FFH enables all 3 amps):

a. 1 = enable amp, 0 = disable amp.
b. Bit 0, enable audioin1 and audioin2.
c. Bit 1, enable audio in.
d. Bit 2, enable 1KW signal on backplane bus.

6. DRA0-7, DRB0-7 to A0-7 and B0-7 interface (use undetermined) (4418H):
    a. 1 = enable, 0 = disable.
    b. Bit 0, enable DRA0-3 to A0-A3.
    c. Bit 1, enable DRA4-7 to A4-7.
    d. Bit 2, enable DRB0-3 to B0-3.
    e. Bit 3, enable DEB4-7 to B4-7.

Appendix: Complete Sample ROMs

PacMan

2000	00	Checksum Header
2001	20	
2002	00	
2003	08	
2004	AA	
2005	00	
2006	00	
2007	00	
2008	00	Header
2009	00	
200a	aa	
200b	00	
200c	2d	
200d	20	
200e	2d	
200f	20	
2010	2d	
2011	20	
2012	2d	
2013	20	
2014	2d	
2015	20	
2016	2d	
2017	20	
2018	2d	
2019	20	
201a	2d	
201b	20	
201c	2d	
201d	20	
201e	2d	
201f	20	
2020	2d	
2021	20	
2022	2d	
2023	20	
2024	2d	
2025	20	
2026	2d	
2027	20	
2028	2d	
2029	20	
202a	2d	
202b	20	
202c	10	
202d	37	(2037H)

202e	20	
202f	39	(2039H)
2030	20	
2031	44	(2044H)
2032	20	
2033	55	(2055H)
2034	20	
2035	00	
2036	00	
2037	67	Points to setup executable code
2038	20	
2039	c1	Subroutine 0BC1
203a	0b	
203b	01	
203c	01	
203d	05	LEDs LA1-8
203e	44	
203f	fe	Default value written to LEDs (saved in 4000H)
2040	0a	(4410H) GPIO A0-A7
2041	01	bit mask
2042	00	no shift
2043	00	end of segment
2044	5e	Subroutine 0C5E
2045	0c	
2046	01	
2047	0d	(4413H) GPIO D0-D7
2048	00	
2049	01	Button PB1
204a	44	
204b	00	clear accum
204c	0f	bit mask
204d	00	no shift
204e	ff	use same I/O output address
204f	00	Switches S1
2050	44	
2051	00	clear accum
2052	03	bit mask
2053	fc	shift left 4 times before writing
2054	00	end of segment
2055	5e	Subroutine 0C5E
2056	0c	
2057	01	
2058	0b	(4411H) GPIO B0-B7
2059	00	
205a	02	Button PR1 (Joystick 1)
205b	44	
205c	00	clear accum
205d	ff	bit mask
205e	00	no shift

205f	0c	(4412H) GPIO C0-C7
2060	00	spacer
2061	04	Button PL (joystick 2)
2062	44	
2063	00	clear accum
2064	ff	bit mask
2065	00	no shift
2066	00	end of segment
2067	3e	executable starts here
2068	ee	
2069	32	
206a	15	I/O VREF Control = EEH
206b	44	
206c	32	
206d	04	Ram address = EEH
206e	40	
206f	3e	
2070	fe	
2071	32	
2072	16	Load Control (A0-7) = FEH
2073	44	
2074	3e	
2075	ff	
2076	32	
2077	17	Load Control (B0-7) = FFH
2078	44	
2079	c9	Ret

Tron

0	
20	2000 checksum start address
0	
8	0800 # positions to checksum
aa	checksum value
0	
0	
0	
0	
0	
aa	
0	
2d	
20	
2d	
20	
2d	
20	
2d	
20	
2d	
20	
2d	
20	
2d	
20	
2d	
20	
2d	
20	
2d	
20	
2d	
20	
2d	
20	
2d	
20	
2d	
20	
2d	
20	
2d	
20	
2d	
20	
ec	
3d	start
20	
3f	

20	
4a 20	
61 20	
83 20	
99 20	
af 20	
0	
0	
c5 20	Executable Address in test ROM
c1 0b	Led Read/Write Subroutine
1	
1	
05 44	LEDs
fe	default write value
0a	4410
ff	bit mask
0	no shift
0	
5e 0c	Standard I/O Read/Write Subroutine
1	
0d	4413
00	
01 44	Push Button PB1-PB8
00	Clear Accum A
3f	input mask
00	no shift
ff	no increment output address
00 44	Switches S1-S8
00	Clear Accum A
1	bit mask
fa	shift left 6
ff	no increment output address
02 44	Push Buttons PR1-PR5 (Only reading bit 0, fire button)
00	Clear Accum A
1	bit mask
f9	shift left 7
00	end

5e	Standard I/O Read/Write Subroutine
0c	
01	
0c	4412
00	
04	Push Buttons PL1-PL5
44	
00	Clear Accum A
1e	bit mask
00	no shift
ff	
04	Push Buttons PL1-PL5
44	
00	Clear Accum A
0e	bit mask
fc	shift left 4
ff	no increment output address
04	Push Buttons PL1-PL5
44	
00	Clear Accum A
10	bit mask
04	shift right 4
00	
a4	Output writing function, writes data from RAM to a port
0e	
00	
11	GPIO B0-B7
44	Output port (also saved to 4002H)
93	Data read from this RAM loc
40	
00	Clear Accum A
ff	mask (and'd w/ 4093H) {Pass all bits}
00	shift bits {No shifting here}
00	End of Segment
c7	encoder emulation executable subroutine address
02	
02	Push Buttons PR1-PR5
44	
04	left, medium speed
10	right, medium speed
1e	read mask, only allowing 4 joystick bits through
1e	read mask, only allowing 4 joystick bits through
00	decrement value
00	
00	Increment Value
00	
02	# of major interrupts to change value (n x 10 interrupts)
00	(IY+2) starting value
00	(IY+3) starting value

0b	Command Byte
a0 40	address loaded into (HL)
78 20	Output writing function address
00	Loaded into 40A0
00	Loaded into 40A1
c7 02	encoder emulation executable subroutine address
02 44	Push Buttons PR1-PR5
0c	Left, fast speed
18	Right, fast speed
1e	read mask, only allowing 4 joystick bits through
1e	read mask, only allowing 4 joystick bits through
00 00	decrement value
00 00	Increment Value
01	# of major interrupts to change value (n x 10 interrupts)
00	(IY+2) starting value
00	(IY+3) starting value
0b	Command Byte
a0 40	address loaded into (HL)
78 20	Output writing function address
00	Loaded into 40A0
00	Loaded into 40A1
c7 02	encoder emulation executable subroutine address
02 44	Push Buttons PR1-PR5
06	Left, slow speed
12	Right, slow speed
1e	read mask, only allowing 4 joystick bits through
1e	read mask, only allowing 4 joystick bits through
00 00	decrement value
00 00	Increment Value
06	# of major interrupts to change value (n x 10 interrupts)
00	(IY+2) starting value
00	(IY+3) starting value
0b	Command Byte
a0 40	address loaded into (HL)
78 20	Output writing function address

00	Loaded into 40A0
00	
3e	executable
fe	
32	
15	(4415H) = FEH
44	
32	
16	(4416H) = FEH
44	
32	
4	(4004H) = FEH (RAM)
40	
3e	
ff	
32	
17	(4417H) = FFH
44	
3e	
fd	
32	
19	(4419H) = FDH (unknown address)
44	
c9	ret

Gorf

0	0	CHECKSUM HEADER
1	20	(2000H) starting address
2	0	
3	8	0800H Locations to checksum
4	aa	Checksum
5	0	
6	0	
7	0	
8	0	
9	0	
0000a	aa	
0000b	0	
0000c	2d	Header
0000d	20	
0000e	2d	
0000f	20	
10	2d	
11	20	
12	2d	
13	20	
14	2d	
15	20	
16	2d	
17	20	
18	2d	
19	20	
0001a	2d	
0001b	20	
0001c	2d	
0001d	20	
0001e	2d	
0001f	20	
20	2d	
21	20	
22	2d	
23	20	
24	2d	
25	20	
26	2d	
27	20	
28	2d	
29	20	
0002a	2d	
0002b	20	
0002c	5e	
0002d	39	Start address table
0002e	20	

0002f	3b	
30	20	
31	46	
32	20	
33	51	
34	20	
35	6e	
36	20	
37	00	
38	00	
39	85	Executable location in test ROM
0003a	20	
0003b	c1	
0003c	0b	Address (0BC1) (Located in program ROM)
0003d	01	
0003e	01	
0003f	05	
40	44	Address (4405H)
41	7c	Default Value
42	0a	address to write out to
43	83	read data and'd with this value
44	00	tells to write value to IY
45	00	end
46	c1	
47	0b	Address (0BC1) (Located in program ROM)
48	01	
49	01	
0004a	06	
0004b	44	Address (4406H)
0004c	81	Default Value
0004d	0b	address to write out to
0004e	7e	value read from (4411H) is and'd with this
0004f	00	tells to write value to IY
50	00	end
51	5e	
52	0c	Address (0C5E) (Located in program ROM)
53	01	
54	0d	address to write out to
55	00	
56	01	
57	44	Address (4401H)
58	00	Clear accum A
59	1f	bit mask
0005a	00	no shifting
0005b	ff	no increment of output address
0005c	00	
0005d	44	Address (4400H)
0005e	00	Clear accum A
0005f	01	bit mask

60	fb	5 shift left
61	ff	no increment of output address
62	04	
63	44	Address (4404H)
64	00	Clear accum A
65	01	bit mask
66	fa	6 shift left
67	ff	no increment of output address
68	02	
69	44	Address (4402H)
0006a	00	Clear accum A
0006b	01	bit mask
0006c	f9	7 shift left
0006d	00	end, go ahead and write to address
0006e	5e	
0006f	0c	Address (0C5E) (Located in program ROM)
70	01	
71	0c	address to write out to
72	00	
73	04	
74	44	Address (4404H)
75	00	Clear accum A
76	1e	bit mask
77	00	no shifting
78	ff	no increment of output address
79	02	
0007a	44	Address (4402H)
0007b	00	Clear accum A
0007c	0e	bit mask
0007d	fc	4 shift left before writing
0007e	ff	no increment of output address
0007f	02	
80	44	Address (4402H)
81	00	Clear accum A
82	10	bit mask
83	04	shift right 4
84	00	end
85	3e	executable
86	fc	
87	32	
88	15	
89	44	(4415H) = FCH
0008a	32	
0008b	04	
0008c	40	(4004H) = FCH
0008d	3e	
0008e	ff	
0008f	32	
90	16	

91	44	(4416H) = FFH
92	32	
93	17	
94	44	(4417H) = FFH
95	32	
96	18	
97	44	(4418H) = FFH
98	32	
99	1a	
0009a	44	(441AH) = FFH
0009b	3e	
0009c	f9	
0009d	32	
0009e	19	(4419H) = F9H
0009f	44	
000a0	c9	