



Document Number DI-044	Page 1 of 24
Supersedes: Rev: C	Rev Date: 04/18/05 Rev: D
Issued By: MARK MATHENY	

## COMMUNICATION PROTOCOL FOR THE MOUSEMAT REV. 1.5

\*\*\*\*\*

**THE MOUSEMAT HAS A USB TO SERIAL CONVERTER BUILT INTO THE CIRCUITRY. YOU MUST INSTALL THE USB TO SERIAL DRIVERS BEFORE YOU CAN COMMUNICATE WITH THE MOUSEMAT. THE SERIAL PORT WILL BE ASSIGNED AT THE TIME OF DRIVER INSTALLATION.**

### Installing the USB drivers onto your computer

Plug the Mousemat into the USB port located on your computer. The computer will recognize that a new device has been added and will prompt the user with the "Found New Hardware Wizard". Check the box named "Install from a list" and then hit the "Next" button. Check the box "Search Removable Media and then hit the "Next" button. The computer should find the USB drivers on the CD and install them to the computer. This process will be completed twice. The first is to register the USB device. The second is to install the USB to serial drivers. After completion of the second installation process, the Mousemat USB drivers should be loaded onto your computer. Windows will now prompt you with a small message indicating the hardware is ready for use.

\*\*\*\*\*

**SET MOUSEMAT APPLICATION SERIAL PARAMETERS AS FOLLOWS.**

- A) 57600 BAUD
- B) 8 DATA BITS
- C) NO PARITY
- D) 1 STOP BIT
- E) NO HARDWARE OR SOFTWARE HANDSHAKE

Document Number DI-044	Page 2 of 24	Revision D
------------------------------	-----------------	---------------

\*\*\*\*\*

## **IDLE LOOP 1**

**THE MOUSEMAT STAYS HERE IN THIS IDLE LOOP WAITING FOR A SINGLE BYTE COMMAND FROM THE APPLICATION.**

**80** HEX FROM THE MOUSEMAT APPLICATION  
BRANCHES TO THE **“READ”** SEQUENCE

**90** HEX FROM THE MOUSEMAT APPLICATION  
BRANCHES TO THE WRITE **“TAGIT ASCII”** SEQUENCE

**91** HEX FROM THE MOUSEMAT APPLICATION  
BRANCHES TO THE WRITE **“TI ISO ASCII”** SEQUENCE

**92** HEX FROM THE MOUSEMAT APPLICATION  
BRANCHES TO THE WRITE **“ICODE ASCII”** SEQUENCE

**93** HEX FROM THE MOUSEMAT APPLICATION  
BRANCHES TO THE WRITE **“PHILIPS ISO ASCII”** SEQUENCE

**94** HEX FROM THE MOUSEMAT APPLICATION  
BRANCHES TO THE WRITE **“TAGIT HEX”** SEQUENCE

**95** HEX FROM THE MOUSEMAT APPLICATION  
BRANCHES TO THE WRITE **“ICODE HEX”** SEQUENCE

**96** HEX FROM THE MOUSEMAT APPLICATION  
BRANCHES TO THE WRITE **“PHILIPS ISO HEX”** SEQUENCE

**97** HEX FROM THE MOUSEMAT APPLICATION  
BRANCHES TO THE WRITE **“TI ISO HEX”** SEQUENCE

**A0** HEX FROM THE MOUSEMAT APPLICATION  
BRANCHES TO THE SEND **“REVISION LEVEL”** SEQUENCE

**A1** HEX FROM MOUSEMAT APPLICATION  
BRANCHES TO THE **DISABLE BEEPER** FUNCTIONS

**A2** HEX FROM THE MOUSEMAT APPLICATION  
BRANCHES TO THE **ENABLE BEEPER** FUNCTIONS

Document Number DI-044	Page 3 of 24	Revision D
------------------------------	-----------------	---------------

**A3** HEX FROM THE MOUSEMAT APPLICATION  
BRANCHES TO THE **BEEP** FUNCTION

**D0** HEX FROM THE MOUSEMAT APPLICATION  
BRANCHES TO THE **REBOOT MOUSEMAT** FUNCTION

\*\*\*\*\*

\*\*\*\*\*

## **READ SEQUENCE**

MOUSEMAT SENDS AN ASCII “OK” TO THE APPLICATION INDICATING IT RECEIVED THE READ COMMAND. IF A RESPONSE IS NOT RECEIVED BY THE APPLICATION, AN ERROR SHOULD BE DISPLAYED INDICATING A COMMUNICATION ERROR.

THE MOUSEMAT NOW CHECKS THE “**TYPE OF TAG**” IN THE FIELD AND RESPONDS WITH THE BYTE LISTED BELOW.

**NOTE: THE MOUSEMAT CAN TAKE UP TO 8 SECONDS TO RESPOND WITH A “TYPE OF TAG” BYTE IF THE SCANNER DOES NOT IMMEDIATELY FIND A TAG IN THE FIELD. THE APPLICATION MUST WAIT UP TO 8 SECONDS FOR A RESPONSE.**

**15** HEX - NO TAG IN FIELD, NOW GO TO “**NO TAG IN FIELD ROUTINE**”

**C2** HEX - TAGIT TAG IN FIELD, NOW GO TO THE “**TAGIT READ ROUTINE**”

**C3** HEX – ISO TAG IN THE FIELD, NOW GO TO THE “**ISO READ ROUTINE**”

**C4** HEX - ICODE IN FIELD, NOW GO TO THE “**ICODE READ ROUTINE**”

**NOTE: THERE IS SEVERAL OTHER RESPONSE CODES DEFINED FOR FUTURE USE. AT THIS TIME, ALL OTHER RESPONSE CODES SHOULD GO TO IDLE LOOP 1**

\*\*\*\*\*

### **NO TAG IN FIELD ROUTINE**

SCANNER NOW GOES TO **IDLE LOOP 1** AND WAITS FOR THE NEXT READ/WRITE COMMAND FROM THE APPLICATION.

Document Number DI-044	Page 4 of 24	Revision D
------------------------------	-----------------	---------------

\*\*\*\*\*

### TAGIT READ ROUTINE

THE MOUSEMAT NOW READS THE TAG AND RESPOND WITH AN ACK (**06 HEX**) FOR A “GOOD READ” OR A NAK (**15 HEX**) FOR A “BAD READ”. THE APPLICATION MUST WAIT UP TO 8 SECONDS FOR A RESPONSE FROM THE SCANNER.

IF A NAK (**15 HEX**) HAS BEEN RECEIVED, GO TO **IDLE LOOP1** AND WAIT FOR THE NEXT COMMAND.

IF AN ACK (**06 HEX**) HAS BEEN RECEIVED GO TO THE NEXT LINE

104 CHARACTERS WILL BE SENT IMMEDIATELY FOLLOWING THE **06 HEX** FROM THE PREVIOUS SEQUENCE. THEY ARE LISTED BELOW.

8 CHARACTERS (SERIAL NUMBER) CONVERTED TO ASCII AND SENT.

64 CHARACTERS (DATA IN TAG) CONVERTED TO ASCII AND SENT.

32 CHARACTERS (DATA IN TAG) SENT AS THEY WERE ORIGINALLY PROGRAMMED.

MOUSEMAT NOW GOES TO **IDLE LOOP 1** AND WAITS FOR THE NEXT COMMAND FROM THE APPLICATION.

\*\*\*\*\*

### ICODE READ ROUTINE

THE SCANNER WILL NOW READ THE TAG AND RESPOND WITH AN ACK (**06 HEX**) FOR A “GOOD READ” OR A NAK (**15 HEX**) FOR A “BAD READ”. THE APPLICATION MUST WAIT UP TO 8 SECONDS FOR A RESPONSE FROM THE SCANNER.

IF AN NAK (**15 HEX**) HAS BEEN RECEIVED, GO TO **IDLE LOOP1** AND WAIT FOR THE NEXT COMMAND.

IF AN ACK (**06 HEX**) HAS BEEN RECEIVED GO TO THE NEXT LINE

160 CHARACTERS WILL BE SENT IMMEDIATELY FOLLOWING THE **06 HEX** FROM THE PREVIOUS SEQUENCE. THEY ARE LISTED BELOW.

Document Number DI-044	Page 5 of 24	Revision D
------------------------------	-----------------	---------------

16 CHARACTERS (SERIAL NUMBER) CONVERTED TO ASCII AND SENT.

96 CHARACTERS (DATA IN TAG) CONVERTED TO ASCII AND SENT

48 CHARACTERS (DATA IN TAG) SENT AS THEY WERE ORIGINALLY PROGRAMMED.

MOUSEMAT NOW GOES TO **IDLE LOOP 1** AND WAITS FOR THE NEXT COMMAND FROM THE APPLICATION.

\*\*\*\*\*

### **ISO READ ROUTINE**

THE SCANNER WILL NOW READ THE TAG AND RESPOND WITH AN ACK (**06 HEX**) FOR A “GOOD READ” OR A NAK (**15 HEX**) FOR A “BAD READ”. THE APPLICATION MUST WAIT UP TO 8 SECONDS FOR A RESPONSE FROM THE SCANNER.

IF A NAK (**15 HEX**) HAS BEEN RECEIVED, GO TO **IDLE LOOP1** AND WAIT FOR THE NEXT COMMAND.

IF AN ACK (**06 HEX**) HAS BEEN RECEIVED GO TO THE NEXT LINE

**NOTE: THIS READ SEQUENCE IS CAPABLE OF READING BOTH T.I. AND PHILIPS ISO TAGS. PLEASE NOTE THE DIFFERENCE IN TAG DATA. 112 CHARACTERS FOR THE PHILLIPS ISO TAG OR 256 CHARACTERS FOR THE T.I. ISO TAG**

358 OR 790 CHARACTERS WILL BE SENT IMMEDIATELY FOLLOWING THE **06 HEX** FROM THE PREVIOUS SEQUENCE. THEY ARE LISTED BELOW IN ORDER OF SENDING.

16 CHARACTERS (SERIAL NUMBER) CONVERTED TO ASCII AND SENT.

2 CHARACTERS (MFG OF TAG) CONVERTED TO ASCII AND SENT.

2 CHARACTERS (NUMBER OF BLOCKS) CONVERTED TO ASCII AND SENT

2 CHARACTERS (NUMBER OF BYTES PER BLOCK) CONVERTED TO ASCII AND SENT

Document Number DI-044	Page 6 of 24	Revision D
------------------------------	-----------------	---------------

224 (PHILIPS) OR 512 (T.I.) CHARACTERS (DATA IN TAG) CONVERTED TO ASCII AND SENT

**NOTE: IN THE TI ISO TAG, THERE IS A SHORT DELAY BETWEEN ASCII AND HEX DATA**

112 (PHILIPS) OR 256 (T.I.) CHARACTERS SENT AS THEY WERE ORIGINALLY PROGRAMMED.

SCANNER NOW GOES TO **IDLE LOOP 1** AND WAITS FOR THE NEXT COMMAND FROM THE APPLICATION.

\*\*\*\*\*  
\*\*\*\*\*

## WRITE SEQUENCES

**THE WRITE SEQUENCE ALLOWS YOU TO WRITE AS LITTLE AS ONE OR AS MANY BLOCKS AS THE TAG HOLDS. EVEN IF YOU SELECT ONE BLOCK TO WRITE, YOU MUST FOLLOW THE PROTOCOL BELOW FOR THE WRITE SEQUENCE TO COMPLETE.**

\*\*\*\*\*  
**T.I. ISO WRITE (ASCII)**

**NOTE: THE WRITE PROTOCOL FOR THIS TAG IS SET UP TO WRITE 4 PARTITIONS OF 16 BLOCKS TOTALING 64 BLOCKS OF DATA OR 256 CHARACTERS. EACH PARTITION REQUIRES 80 CHARACTERS TO BE SENT. THE BREAKDOWN FOR A PARTITION IS AS FOLLOWS.**

**EACH BLOCK OF DATA SENT TO THE SCANNER MUST BE STRUCTURED AS 5 BYTES. THE COMMAND BYTE FOLLOWED BY 4 DATA BYTES.**

**NO WRITE TO BLOCK        =   00H, 00H, 00H, 00H, 00H**  
**WRITE BLOCK                =   F0H, BYTE, BYTE, BYTE, BYTE**  
**WRITE BLOCK AND LOCK =   F1H, BYTE, BYTE, BYTE, BYTE**

**(16 BLOCKS/PARTITION) X (5 BYTES) = 80 CHAR PER PARTITION**

Document Number DI-044	Page 7 of 24	Revision D
------------------------------	-----------------	---------------

**THIS PROCESS MUST BE COMPLETED 4 TIMES FOR THE WHOLE TAG TO BE WRITTEN. ALL 4 PARTITIONS MUST BE SENT TO THE SCANNER FOR THE WRITE PROCESS TO BE COMPLETED. YOU CAN'T WRITE LESS THAN 4 PARTITIONS.**

**2 BYTES WILL BE RETURNED AS A RESPONSE CODE AFTER EACH BLOCK IN THE PARTITION HAS BEEN WRITTEN. THE BREAKDOWN OF THE RESPONSE CODES IS AS FOLLOWS.**

**THE RESPONSE CODES DEPEND ON A TAG PRESENT IN THE RF FIELD. THE APPLICATION MUST WAIT UP TO 8 SECONDS FOR A RESPONSE CODE IF A TAG IS NOT PRESENT IN THE FIELD.**

**BLOCK #**

- 
- BLOCK 1 = 30H**
  - BLOCK 2 = 31H**
  - BLOCK 3 = 32H**
  - BLOCK 4 = 33H**
  - BLOCK 5 = 34H**
  - BLOCK 6 = 35H**
  - BLOCK 7 = 36H**
  - BLOCK 8 = 37H**
  - BLOCK 9 = 38H**
  - BLOCK 10 = 39H**
  - BLOCK 11 = 3AH**
  - BLOCK 12 = 3BH**
  - BLOCK 13 = 3CH**
  - BLOCK 14 = 3DH**
  - BLOCK 15 = 3EH**
  - BLOCK 16 = 3FH**

**RESPONSE CODES**

- 
- BLOCK#, C9H= BLOCK ERROR ATTEMPTING TO RE WRITE**
  - BLOCK#, C8H= ERROR NO TAG FOUND**
  - BLOCK#, CAH= BLOCK NOT WRITTEN**
  - CBH, CBH= PARTITION COMPLETE, SEND NEXT PARTITION DATA**
  - BLOCK #, C5H = WRITE TO BLOCK COMPLETE (GOOD)**
  - BLOCK #, C6H = ERROR, BLOCK ALREADY LOCKED**
  - BLOCK #, C7H = BLOCK WRITTEN AND NOW LOCKED**
  - CCH, CCH= WRITE TO TAG GOOD AND COMPLETE**

Document Number DI-044	Page 8 of 24	Revision D
------------------------------	-----------------	---------------

#####

## **PROTOCOL**

AN ASCII “**OK**” IS RETURNED TO THE APPLICATION INDICATING THAT THE MOUSEMAT HAS ACKNOWLEDGED THE COMMAND.

### **FIRST PARTITION**

THE APPLICATION SENDS **80** CHARACTERS TO THE MOUSEMAT AND WAITS FOR A RESPONSE FROM THE MOUSEMAT.

THE MOUSEMAT RESPONDS WITH A **06 HEX** INDICATING IT RECEIVED THE 80 CHARACTERS OF DATA.

THE MOUSEMAT IMMEDIATELY ATTEMPTS TO WRITE THE TAG

A RESPONSE CODE WILL BE RECEIVED AFTER EACH BLOCK WRITE

### **SECOND PARTITION**

THE APPLICATION NOW SENDS THE NEXT **80** CHARACTERS TO THE MOUSEMAT AND WAITS FOR A RESPONSE FROM THE MOUSEMAT.

THE MOUSEMAT RESPONDS WITH A **06 HEX** INDICATING IT RECEIVED THE 80 CHARACTERS OF DATA.

THE MOUSEMAT IMMEDIATELY ATTEMPTS TO WRITE THE TAG

A RESPONSE CODE WILL BE RECEIVED AFTER EACH BLOCK WRITE

### **THIRD PARTITION**

THE APPLICATION NOW SENDS THE NEXT **80** CHARACTERS TO THE MOUSEMAT AND WAITS FOR A RESPONSE FROM THE MOUSEMAT.

THE MOUSEMAT RESPONDS WITH A **06 HEX** INDICATING IT RECEIVED THE 80 CHARACTERS OF DATA.

THE MOUSEMAT IMMEDIATELY ATTEMPTS TO WRITE THE TAG

A RESPONSE CODE WILL BE RECEIVED AFTER EACH BLOCK WRITE

Document Number DI-044	Page 9 of 24	Revision D
------------------------------	-----------------	---------------

**FOURTH PARTITION**

THE MOUSEMAT APPLICATION SENDS **80** CHARACTERS TO THE MOUSEMAT AND WAITS FOR A RESPONSE FROM THE MOUSEMAT.

THE SCANNER RESPONDS WITH A **06 HEX** INDICATING IT RECEIVED THE 80 CHARACTERS OF DATA.

THE MOUSEMAT IMMEDIATELY ATTEMPTS TO WRITE THE TAG

A RESPONSE CODE WILL BE RECEIVED AFTER EACH BLOCK WRITE

THE MOUSEMAT NOW GOES TO **IDLE LOOP 1** AND WAITS FOR THE NEXT COMMAND FROM THE APPLICATION.

\*\*\*\*\*

**T.I. ISO WRITE (HEX)**

**NOTE: THE WRITE PROTOCOL FOR THIS TAG IS SET UP TO WRITE 4 PARTITIONS OF 16 BLOCKS TOTALING 64 BLOCKS OF DATA OR 256 CHARACTERS. EACH PARTITION REQUIRES 160 CHARACTERS TO BE SENT. THE BREAKDOWN FOR A PARTITION IS AS FOLLOWS.**

**IN THIS MODE, AN ASCII TO HEX CONVERSION TAKES PLACE. 2 ASCII BYTES REPRESENT THE UPPER AND LOWER NIBBLE OF THE HEX BYTE. YOU MUST ENTER ONLY CHARACTERS 0-9 AND A-F FOR THE HEX CHARACTERS.**

**EXAMPLE: "AA" = AA HEX**

**EACH BLOCK OF DATA SENT TO THE SCANNER MUST BE STRUCTURED AS 10 BYTES. TWO COMMAND CHARACTERS FOLLOWED BY 8 DATA BYTES.**

**NO WRITE TO BLOCK = "0, 0, 0, 0, 0, 0, 0, 0, 0, 0"**  
**WRITE BLOCK = "F 0", (8 ASCII CHARS)**  
**WRITE BLOCK AND LOCK = "F1", (8 ASCII CHARS)**

**(16 BLOCKS/PARTITION) X (10 BYTES) = 160 CHAR PER PARTITION**

**THIS PROCESS MUST BE COMPLETED 4 TIMES FOR THE WHOLE TAG TO BE WRITTEN. ALL 4 PARTITIONS MUST BE SENT TO THE**

Document Number DI-044	Page 10 of 24	Revision D
------------------------------	------------------	---------------

**SCANNER FOR THE WRITE PROCESS TO BE COMPLETED. YOU CAN'T WRITE LESS THAN 4 PARTITIONS.**

**2 BYTES WILL BE RETURNED AS A RESPONSE CODE AFTER EACH BLOCK IN THE PARTITION HAS BEEN WRITTEN. EXCEPT WHEN A "NO WRITE TO BLOCK" COMMAND IS SENT. THERE IS NO RESPONSE FOR THIS COMMAND CODE. THE BREAKDOWN OF THE RESPONSE CODES IS AS FOLLOWS.**

**THE RESPONSE CODES DEPEND ON A TAG PRESENT IN THE RF FIELD. THE APPLICATION MUST WAIT UP TO 8 SECONDS FOR A RESPONSE CODE IF A TAG IS NOT PRESENT IN THE FIELD.**

**BLOCK #**

-----

- BLOCK 1 = 30H**
- BLOCK 2 = 31H**
- BLOCK 3 = 32H**
- BLOCK 4 = 33H**
- BLOCK 5 = 34H**
- BLOCK 6 = 35H**
- BLOCK 7 = 36H**
- BLOCK 8 = 37H**
- BLOCK 9 = 38H**
- BLOCK 10 = 39H**
- BLOCK 11 = 3AH**
- BLOCK 12 = 3BH**
- BLOCK 13 = 3CH**
- BLOCK 14 = 3DH**
- BLOCK 15 = 3EH**
- BLOCK 16 = 3FH**

**RESPONSE CODES**

-----

- BLOCK#, C9H= BLOCK ERROR ATTEMPTING TO RE WRITE**
- BLOCK#, C8H= ERROR NO TAG FOUND**
- BLOCK#, CAH= BLOCK NOT WRITTEN**
- CBH, CBH= PARTITION COMPLETE, SEND NEXT PARTITION DATA**
- BLOCK #, C5H = WRITE TO BLOCK COMPLETE (GOOD)**
- BLOCK #, C6H = ERROR, BLOCK ALREADY LOCKED**
- BLOCK #, C7H = BLOCK WRITTEN AND NOW LOCKED**
- CCH, CCH= WRITE TO TAG GOOD AND COMPLETE**

Document Number DI-044	Page 11 of 24	Revision D
------------------------------	------------------	---------------

#####

## **PROTOCOL**

AN ASCII “**OK**” IS RETURNED TO THE APPLICATION INDICATING THAT THE SCANNER HAS ACKNOWLEDGED THE COMMAND.

### **FIRST PARTITION**

THE APPLICATION SENDS **160** CHARACTERS TO THE SCANNER AND WAITS FOR A RESPONSE FROM THE SCANNER.

THE MOUSEMAT RESPONDS WITH A **06 HEX** INDICATING IT RECEIVED THE 160 CHARACTERS OF DATA.

THE MOUSEMAT IMMEDIATELY ATTEMPTS TO WRITE THE TAG

A RESPONSE CODE WILL BE RECEIVED AFTER EACH BLOCK WRITE

### **SECOND PARTITION**

THE APPLICATION SENDS **160** CHARACTERS TO THE MOUSEMAT AND WAITS FOR A RESPONSE FROM THE MOUSEMAT.

THE MOUSEMAT RESPONDS WITH A **06 HEX** INDICATING IT RECEIVED THE 160 CHARACTERS OF DATA.

THE MOUSEMAT IMMEDIATELY ATTEMPTS TO WRITE THE TAG

A RESPONSE CODE WILL BE RECEIVED AFTER EACH BLOCK WRITE

### **THIRD PARTITION**

THE APPLICATION SENDS **160** CHARACTERS TO THE MOUSEMAT AND WAITS FOR A RESPONSE FROM THE MOUSEMAT.

THE MOUSEMAT RESPONDS WITH A **06 HEX** INDICATING IT RECEIVED THE 160 CHARACTERS OF DATA.

THE MOUSEMAT IMMEDIATELY ATTEMPTS TO WRITE THE TAG

A RESPONSE CODE WILL BE RECEIVED AFTER EACH BLOCK WRITE

Document Number DI-044	Page 12 of 24	Revision D
------------------------------	------------------	---------------

**FOURTH PARTITION**

THE APPLICATION SENDS **160** CHARACTERS TO THE MOUSEMAT AND WAITS FOR A RESPONSE FROM THE MOUSEMAT.

THE MOUSEMAT RESPONDS WITH A **06 HEX** INDICATING IT RECEIVED THE 160 CHARACTERS OF DATA.

THE MOUSEMAT IMMEDIATELY ATTEMPTS TO WRITE THE TAG

A RESPONSE CODE WILL BE RECEIVED AFTER EACH BLOCK WRITE

MOUSEMAT NOW GOES TO **IDLE LOOP 1** AND WAITS FOR THE NEXT COMMAND FROM THE APPLICATION.

\*\*\*\*\*

**PHILIPS ISO WRITE (ASCII)**

**NOTE: THE WRITE PROTOCOL FOR THIS TAG IS SET UP TO WRITE 1 PARTITION OF 28 BLOCKS OF DATA OR 112 CHARACTERS. THE PARTITION REQUIRES 140 CHARACTERS TO BE SENT. THE BREAKDOWN FOR THE PARTITION IS AS FOLLOWS.**

**EACH BLOCK OF DATA SENT TO THE MOUSEMAT MUST BE STRUCTURED AS 5 BYTES. THE COMMAND BYTE FOLLOWED BY 4 DATA BYTES.**

**NO WRITE TO BLOCK       = 00H, 00H, 00H, 00H, 00H**  
**WRITE BLOCK               = F0H, BYTE, BYTE, BYTE, BYTE**  
**WRITE BLOCK AND LOCK = F1H, BYTE, BYTE, BYTE, BYTE**

**(28 BLOCKS/PARTITION) X (5 BYTES) = 140 CHAR PER PARTITION**

**2 BYTES WILL BE RETURNED AS A RESPONSE CODE AFTER EACH BLOCK IN THE PARTITION HAS BEEN WRITTEN. THE BREAKDOWN OF THE RESPONSE CODES IS AS FOLLOWS.**

**THE RESPONSE CODES DEPEND ON A TAG PRESENT IN THE RF FIELD. THE APPLICATION MUST WAIT UP TO 8 SECONDS FOR A RESPONSE CODE IF A TAG IS NOT PRESENT IN THE FIELD.**

**BLOCK #**  
**-----**  
**BLOCK 1 = 30H**

Document Number DI-044	Page 13 of 24	Revision D
------------------------------	------------------	---------------

**BLOCK 2 = 31H**  
**BLOCK 3 = 32H**  
**BLOCK 4 = 33H**  
**BLOCK 5 = 34H**  
**BLOCK 6 = 35H**  
**BLOCK 7 = 36H**  
**BLOCK 8 = 37H**  
**BLOCK 9 = 38H**  
**BLOCK 10 = 39H**  
**BLOCK 11= 3AH**  
**BLOCK 12 = 3BH**  
**BLOCK 13 = 3CH**  
**BLOCK 14 = 3DH**

**RESPONSE CODES**

-----  
**BLOCK#, C9H= BLOCK ERROR ATTEMPTING TO RE WRITE**  
**BLOCK#, C8H= ERROR NO TAG FOUND**  
**BLOCK#, CAH= BLOCK NOT WRITTEN**  
**CBH, CBH= PARTITION COMPLETE, SEND NEXT PARTITION DATA**  
**BLOCK #, C5H = WRITE TO BLOCK COMPLETE (GOOD)**  
**BLOCK #, C6H = ERROR, BLOCK ALREADY LOCKED**  
**BLOCK #, C7H = BLOCK WRITTEN AND NOW LOCKED**  
**CCH, CCH= WRITE TO TAG GOOD AND COMPLETE**

#####

**PROTOCOL**

AN ASCII “OK” IS RETURNED TO THE APPLICATION INDICATING THAT THE MOUSEMAT HAS ACKNOWLEDGED THE COMMAND.

THE APPLICATION SENDS **140** CHARACTERS TO THE MOUSEMAT AND WAITS FOR A RESPONSE FROM THE MOUSEMAT.

THE MOUSEMAT RESPONDS WITH A **06 HEX** INDICATING IT RECEIVED THE **140** CHARACTERS OF DATA.

THE MOUSEMAT IMMEDIATELY ATTEMPTS TO WRITE THE TAG

A RESPONSE CODE WILL BE RECEIVED AFTER EACH BLOCK WRITE

Document Number DI-044	Page 14 of 24	Revision D
------------------------------	------------------	---------------

THE MOUSEMAT NOW GOES TO **IDLE LOOP 1** AND WAITS FOR THE NEXT COMMAND FROM THE APPLICATION.

\*\*\*\*\*

**PHILIPS ISO WRITE (HEX)**

**NOTE: THE WRITE PROTOCOL FOR THIS TAG IS SET UP TO WRITE 28 BLOCKS OF DATA OR 280 CHARACTERS. THE BREAKDOWN FOR A PARTITION IS AS FOLLOWS.**

**IN THIS MODE, AN ASCII TO HEX CONVERSION TAKES PLACE. 2 ASCII BYTES REPRESENT THE UPPER AND LOWER NIBBLE OF THE HEX BYTE. YOU MUST ENTER ONLY CHARACTERS 0-9 AND A-F FOR THE HEX CHARACTERS.**

**EXAMPLE: "AA" = AA HEX**

**EACH BLOCK OF DATA SENT TO THE MOUSEMAT MUST BE STRUCTURED AS 10 BYTES. THE 2 COMMAND CHARACTERS FOLLOWED BY 8 DATA BYTES.**

**NO WRITE TO BLOCK = "0, 0, 0, 0, 0, 0, 0, 0, 0, 0"**  
**WRITE BLOCK = "F 0", (8 ASCII CHARS)**  
**WRITE BLOCK AND LOCK = "F1", (8 ASCII CHARS)**

**(28 BLOCKS/PARTITION) X (10 BYTES) = 280 CHAR PER PARTITION**

**2 BYTES WILL BE RETURNED AS A RESPONSE CODE AFTER EACH BLOCK IN THE PARTITION HAS BEEN WRITTEN. THE BREAKDOWN OF THE RESPONSE CODES IS AS FOLLOWS.**

**THE RESPONSE CODES DEPEND ON A TAG PRESENT IN THE RF FIELD. THE APPLICATION MUST WAIT UP TO 7 SECONDS FOR A RESPONSE CODE IF A TAG IS NOT PRESENT IN THE FIELD.**

**BLOCK #**

- **BLOCK 1 = 30H**  
**BLOCK 2 = 31H**  
**BLOCK 3 = 32H**  
**BLOCK 4 = 33H**  
**BLOCK 5 = 34H**  
**BLOCK 6 = 35H**

Document Number DI-044	Page 15 of 24	Revision D
------------------------------	------------------	---------------

**BLOCK 7 = 36H**  
**BLOCK 8 = 37H**  
**BLOCK 9 = 38H**  
**BLOCK 10 = 39H**  
**BLOCK 11= 3AH**  
**BLOCK 12 = 3BH**  
**BLOCK 13 = 3CH**  
**BLOCK 14 = 3DH**

**RESPONSE CODES**

-----  
**BLOCK#, C9H= BLOCK ERROR ATTEMPTING TO RE WRITE**  
**BLOCK#, C8H= ERROR NO TAG FOUND**  
**BLOCK#, CAH= BLOCK NOT WRITTEN**  
**CBH, CBH= PARTITION COMPLETE, SEND NEXT PARTITION DATA**  
**BLOCK #, C5H = WRITE TO BLOCK COMPLETE (GOOD)**  
**BLOCK #, C6H = ERROR, BLOCK ALREADY LOCKED**  
**BLOCK #, C7H = BLOCK WRITTEN AND NOW LOCKED**  
**CCH, CCH= WRITE TO TAG GOOD AND COMPLETE**

#####

**PROTOCOL**

AN ASCII “**OK**” IS RETURNED TO THE APPLICATION INDICATING THAT THE MOUSEMAT HAS ACKNOWLEDGED THE COMMAND.

THE APPLICATION SENDS **280** CHARACTERS TO THE MOUSEMAT AND WAITS FOR A RESPONSE FROM THE MOUSEMAT.

THE MOUSEMAT RESPONDS WITH A **06 HEX** INDICATING IT RECEIVED THE 280 CHARACTERS OF DATA.

THE MOUSEMAT IMMEDIATELY ATTEMPTS TO WRITE THE TAG

A RESPONSE CODE WILL BE RECEIVED AFTER EACH BLOCK WRITE

THE MOUSEMAT NOW GOES TO **IDLE LOOP 1** AND WAITS FOR THE NEXT COMMAND FROM THE APPLICATION.

\*\*\*\*\*

**TAGIT WRITE (ASCII)**

Document Number DI-044	Page 16 of 24	Revision D
------------------------------	------------------	---------------

**NOTE: THE WRITE PROTOCOL FOR THIS TAG IS SET UP TO WRITE 8 BLOCKS TOTALING 32 CHARACTERS.**

**EACH BLOCK OF DATA SENT TO THE MOUSEMAT MUST BE STRUCTURED AS 5 BYTES. THE COMMAND BYTE FOLLOWED BY 4 DATA BYTES.**

**NO WRITE TO BLOCK = 00H, 00H, 00H, 00H, 00H**  
**WRITE BLOCK = F0H, BYTE, BYTE, BYTE, BYTE**  
**WRITE BLOCK AND LOCK = F1H, BYTE, BYTE, BYTE, BYTE**

**(8 BLOCKS) X (5 BYTES) = 40 CHARS**

**2 BYTES WILL BE RETURNED AS A RESPONSE CODE AFTER EACH BLOCK HAS BEEN WRITTEN. THE BREAKDOWN OF THE RESPONSE CODES IS AS FOLLOWS.**

**THE RESPONSE CODES DEPEND ON A TAG PRESENT IN THE RF FIELD. THE APPLICATION MUST WAIT UP TO 8 SECONDS FOR A RESPONSE CODE IF A TAG IS NOT PRESENT IN THE FIELD.**

**BLOCK #**

-----  
**BLOCK 1 = 30H**  
**BLOCK 2 = 31H**  
**BLOCK 3 = 32H**  
**BLOCK 4 = 33H**  
**BLOCK 5 = 34H**  
**BLOCK 6 = 35H**  
**BLOCK 7 = 36H**  
**BLOCK 8 = 37H**

**RESPONSE CODES**

-----  
**BLOCK#, C9H= BLOCK ERROR ATTEMPTING TO RE WRITE**  
**BLOCK#, C8H= ERROR NO TAG FOUND**  
**BLOCK#, CAH= BLOCK NOT WRITTEN**  
**CBH, CBH= PARTITION COMPLETE, SEND NEXT PARTITION DATA**  
**BLOCK #, C5H = WRITE TO BLOCK COMPLETE (GOOD)**  
**BLOCK #, C6H = ERROR, BLOCK ALREADY LOCKED**  
**BLOCK #, C7H = BLOCK WRITTEN AND NOW LOCKED**  
**CCH, CCH= WRITE TO TAG GOOD AND COMPLETE**

Document Number DI-044	Page 17 of 24	Revision D
------------------------------	------------------	---------------

#####

## PROTOCOL

AN ASCII "OK" IS RETURNED TO THE APPLICATION INDICATING THAT THE MOUSEMAT HAS ACKNOWLEDGED THE COMMAND.

THE APPLICATION SENDS 40 CHARACTERS TO THE MOUSEMAT AND WAITS FOR A RESPONSE FROM THE MOUSEMAT.

THE MOUSEMAT RESPONDS WITH A 06 HEX INDICATING IT RECEIVED THE 40 CHARACTERS OF DATA.

THE MOUSEMAT IMMEDIATELY ATTEMPTS TO WRITE THE TAG

A RESPONSE CODE WILL BE RECEIVED AFTER EACH BLOCK WRITE

MOUSEMAT NOW GOES TO **IDLE LOOP 1** AND WAITS FOR THE NEXT COMMAND FROM THE APPLICATION.

\*\*\*\*\*

### TAGIT WRITE (HEX)

**NOTE: THE WRITE PROTOCOL FOR THIS TAG IS SET UP TO WRITE 8 BLOCKS OF DATA TOTALING 64 CHARACTERS.**

**IN THIS MODE, AN ASCII TO HEX CONVERSION TAKES PLACE. 2 ASCII BYTES REPRESENT THE UPPER AND LOWER NIBBLE OF THE HEX BYTE. YOU MUST ENTER ONLY CHARACTERS 0-9 AND A-F FOR THE HEX CHARACTERS.**

**EXAMPLE: "A1" = A1 HEX**

**EACH BLOCK OF DATA SENT TO THE MOUSEMAT MUST BE STRUCTURED AS 10 BYTES. THE 2 COMMAND CHARACTERS FOLLOWED BY 8 DATA BYTES.**

**NO WRITE TO BLOCK = "0, 0, 0, 0, 0, 0, 0, 0, 0, 0"**  
**WRITE BLOCK = "F 0", (8 ASCII CHARS)**  
**WRITE BLOCK AND LOCK = "F1", (8 ASCII CHARS)**

**(8 BLOCKS) X (10 BYTES) = 80 CHARS**

Document Number DI-044	Page 18 of 24	Revision D
------------------------------	------------------	---------------

**2 BYTES WILL BE RETURNED AS A RESPONSE CODE AFTER EACH BLOCK HAS BEEN WRITTEN. THE BREAKDOWN OF THE RESPONSE CODES IS AS FOLLOWS.**

**THE RESPONSE CODES DEPEND ON A TAG PRESENT IN THE RF FIELD. THE APPLICATION MUST WAIT UP TO 8 SECONDS FOR A RESPONSE CODE IF A TAG IS NOT PRESENT IN THE FIELD.**

**BLOCK #**

-----

- BLOCK 1 = 30H**
- BLOCK 2 = 31H**
- BLOCK 3 = 32H**
- BLOCK 4 = 33H**
- BLOCK 5 = 34H**
- BLOCK 6 = 35H**
- BLOCK 7 = 36H**
- BLOCK 8 = 37H**

**RESPONSE CODES**

-----

- BLOCK#, C9H= BLOCK ERROR ATTEMPTING TO RE WRITE**
- BLOCK#, C8H= ERROR NO TAG FOUND**
- BLOCK#, CAH= BLOCK NOT WRITTEN**
- CBH, CBH= PARTITION COMPLETE, SEND NEXT PARTITION DATA**
- BLOCK #, C5H = WRITE TO BLOCK COMPLETE (GOOD)**
- BLOCK #, C6H = ERROR, BLOCK ALREADY LOCKED**
- BLOCK #, C7H = BLOCK WRITTEN AND NOW LOCKED**
- CCH, CCH= WRITE TO TAG GOOD AND COMPLETE**

#####

**PROTOCOL**

AN ASCII **“OK”** IS RETURNED TO THE APPLICATION INDICATING THAT THE MOUSEMAT HAS ACKNOWLEDGED THE COMMAND.

THE APPLICATION SENDS **80** CHARACTERS TO THE MOUSEMAT AND WAITS FOR A RESPONSE FROM THE MOUSEMAT.

THE MOUSEMAT RESPONDS WITH A **06 HEX** INDICATING IT RECEIVED THE **80** CHARACTERS OF DATA.

Document Number DI-044	Page 19 of 24	Revision D
------------------------------	------------------	---------------

THE MOUSEMAT IMMEDIATELY ATTEMPTS TO WRITE THE TAG

A RESPONSE CODE WILL BE RECEIVED AFTER EACH BLOCK WRITE

THE MOUSEMAT NOW GOES TO **IDLE LOOP 1** AND WAITS FOR THE NEXT COMMAND FROM THE APPLICATION.

\*\*\*\*\*

### ICODE WRITE (ASCII)

**NOTE: THE WRITE PROTOCOL FOR THIS TAG IS SET UP TO WRITE 12 BLOCKS TOTALING 48 CHARACTERS.**

**EACH BLOCK OF DATA SENT TO THE MOUSEMAT MUST BE STRUCTURED AS 5 BYTES. THE COMMAND BYTE FOLLOWED BY 4 DATA BYTES.**

**NO WRITE TO BLOCK       = 00H, 00H, 00H, 00H, 00H**  
**WRITE BLOCK             = F0H, BYTE, BYTE, BYTE, BYTE**  
**WRITE BLOCK AND LOCK = F1H, BYTE, BYTE, BYTE, BYTE**

**(12 BLOCKS) X (5 BYTES) = 60 CHARS**

**2 BYTES WILL BE RETURNED AS A RESPONSE CODE AFTER EACH BLOCK HAS BEEN WRITTEN. THE BREAKDOWN OF THE RESPONSE CODES IS AS FOLLOWS.**

**THE RESPONSE CODES DEPEND ON A TAG PRESENT IN THE RF FIELD. THE APPLICATION MUST WAIT UP TO 8 SECONDS FOR A RESPONSE CODE IF A TAG IS NOT PRESENT IN THE FIELD.**

### BLOCK #

-----  
**BLOCK 1 = 30H**  
**BLOCK 2 = 31H**  
**BLOCK 3 = 32H**  
**BLOCK 4 = 33H**  
**BLOCK 5 = 34H**  
**BLOCK 6 = 35H**  
**BLOCK 7 = 36H**  
**BLOCK 8 = 37H**

Document Number DI-044	Page 20 of 24	Revision D
------------------------------	------------------	---------------

**BLOCK 9 = 38H**  
**BLOCK 10 = 39H**  
**BLOCK 11 = 3AH**  
**BLOCK 12 = 3BH**

**RESPONSE CODES**

-----  
**BLOCK#, C9H= BLOCK ERROR ATTEMPTING TO RE WRITE**  
**BLOCK#, C8H= ERROR NO TAG FOUND**  
**BLOCK#, CAH= BLOCK NOT WRITTEN**  
**CBH, CBH= PARTITION COMPLETE, SEND NEXT PARTITION DATA**  
**BLOCK #, C5H = WRITE TO BLOCK COMPLETE (GOOD)**  
**BLOCK #, C6H = ERROR, BLOCK ALREADY LOCKED**  
**BLOCK #, C7H = BLOCK WRITTEN AND NOW LOCKED**  
**CCH, CCH= WRITE TO TAG GOOD AND COMPLETE**

#####

**PROTOCOL**

AN ASCII “**OK**” IS RETURNED TO THE APPLICATION INDICATING THAT THE MOUSEMAT HAS ACKNOWLEDGED THE COMMAND.

THE APPLICATION SENDS **60** CHARACTERS TO THE MOUSEMAT AND WAITS FOR A RESPONSE FROM THE MOUSEMAT.

THE MOUSEMAT RESPONDS WITH A **06 HEX** INDICATING IT RECEIVED THE **60** CHARACTERS OF DATA.

THE MOUSEMAT IMMEDIATELY ATTEMPTS TO WRITE THE TAG

A RESPONSE CODE WILL BE RECEIVED AFTER EACH BLOCK WRITE

THE MOUSEMAT NOW GOES TO **IDLE LOOP 1** AND WAITS FOR THE NEXT COMMAND FROM THE APPLICATION.

\*\*\*\*\*

**ICODE WRITE (HEX)**

**NOTE: THE WRITE PROTOCOL FOR THIS TAG IS SET UP TO WRITE 8 BLOCKS OF DATA TOTALING 96 CHARACTERS.**

Document Number DI-044	Page 21 of 24	Revision D
------------------------------	------------------	---------------

**IN THIS MODE, AN ASCII TO HEX CONVERSION TAKES PLACE. 2 ASCII BYTES REPRESENT THE UPPER AND LOWER NIBBLE OF THE HEX BYTE. YOU MUST ENTER ONLY CHARACTERS 0-9 AND A-F FOR THE HEX CHARACTERS.**

**EXAMPLE: "A1" = A1 HEX**

**EACH BLOCK OF DATA SENT TO THE MOUSEMAT MUST BE STRUCTURED AS 10 BYTES. THE 2 COMMAND CHARACTERS FOLLOWED BY 8 DATA BYTES.**

**NO WRITE TO BLOCK = "0, 0, 0, 0, 0, 0, 0, 0, 0, 0"  
WRITE BLOCK = "F 0", (8 ASCII CHARS)  
WRITE BLOCK AND LOCK = "F1", (8 ASCII CHARS)**

**(12 BLOCKS) X (10 BYTES) = 120 CHARS**

**2 BYTES WILL BE RETURNED AS A RESPONSE CODE AFTER EACH BLOCK IN HAS BEEN WRITTEN. THE BREAKDOWN OF THE RESPONSE CODES IS AS FOLLOWS.**

**THE RESPONSE CODES DEPEND ON A TAG PRESENT IN THE RF FIELD. THE APPLICATION MUST WAIT UP TO 8 SECONDS FOR A RESPONSE CODE IF A TAG IS NOT PRESENT IN THE FIELD.**

**BLOCK #**

- **BLOCK 1 = 30H  
BLOCK 2 = 31H  
BLOCK 3 = 32H  
BLOCK 4 = 33H  
BLOCK 5 = 34H  
BLOCK 6 = 35H  
BLOCK 7 = 36H  
BLOCK 8 = 37H  
BLOCK 9 = 38H  
BLOCK 10 = 39H  
BLOCK 11= 3AH  
BLOCK 12= 3BH**

**RESPONSE CODES**

- **BLOCK#, C9H= BLOCK ERROR ATTEMPTING TO RE WRITE  
BLOCK#, C8H= ERROR NO TAG FOUND**

Document Number DI-044	Page 22 of 24	Revision D
------------------------------	------------------	---------------

**BLOCK#, CAH= BLOCK NOT WRITTEN**  
**CBH, CBH= PARTITION COMPLETE, SEND NEXT PARTITION DATA**  
**BLOCK #, C5H = WRITE TO BLOCK COMPLETE (GOOD)**  
**BLOCK #, C6H = ERROR, BLOCK ALREADY LOCKED**  
**BLOCK #, C7H = BLOCK WRITTEN AND NOW LOCKED**  
**CCH, CCH= WRITE TO TAG GOOD AND COMPLETE**

#####

**PROTOCOL**

AN ASCII **“OK”** IS RETURNED TO THE APPLICATION INDICATING THAT THE MOUSEMAT HAS ACKNOWLEDGED THE COMMAND.

THE APPLICATION SENDS **120** CHARACTERS TO THE MOUSEMAT AND WAITS FOR A RESPONSE FROM THE MOUSEMAT.

THE MOUSEMAT RESPONDS WITH A **06 HEX** INDICATING IT RECEIVED THE **120** CHARACTERS OF DATA.

THE MOUSEMAT IMMEDIATELY ATTEMPTS TO WRITE THE TAG

A RESPONSE CODE WILL BE RECEIVED AFTER EACH BLOCK WRITE

THE MOUSEMAT NOW GOES TO **IDLE LOOP 1** AND WAITS FOR THE NEXT COMMAND FROM THE APPLICATION.

\*\*\*\*\*

\*\*\*\*\*

**SEND REVISION LEVEL**

3 ASCII CHARACTERS ARE RETURNED TO THE APPLICATION.  
**(EXAMPLE: 1.0 OR (31 HEX, 2E HEX, 30 HEX)**

MOUSEMAT NOW GOES TO **IDLE LOOP 1** AND WAITS FOR THE NEXT COMMAND FROM THE APPLICATION.

\*\*\*\*\*

\*\*\*\*\*

Document Number DI-044	Page 23 of 24	Revision D
------------------------------	------------------	---------------

# DISABLE BEEPER FUNCTION

AFTER THE APPLICATION SENDS AN **A1** HEX TO THE MOUSEMAT, ALL BEEPER FUNCTIONS WILL BE DISABLED UNTIL AN **ENABLE BEEPER** COMMAND IS SENT OR THE MOUSEMAT IS POWERED DOWN AND THEN REPOWERED.

THE MOUSEMAT NOW GOES TO **IDLE LOOP 1** AND WAITS FOR THE NEXT COMMAND FROM THE APPLICATION.

\*\*\*\*\*  
\*\*\*\*\*

# ENABLE BEEPER FUNCTION

AFTER THE APPLICATION SENDS AN **A2** HEX TO THE MOUSEMAT, ALL BEEPER FUNCTIONS ARE ENABLED. THIS MODE IS THE DEFAULT MODE. THE MOUSEMAT NOW GOES TO **IDLE LOOP 1** AND WAITS FOR THE NEXT COMMAND FROM THE APPLICATION.

\*\*\*\*\*  
\*\*\*\*\*

# BEEP

AFTER THE APPLICATION SENDS AN **A3** HEX TO THE MOUSEMAT, THE MOUSEMAT WILL GENERATE A BEEP SEQUENCE. THE MOUSEMAT NOW GOES TO **IDLE LOOP 1** AND WAITS FOR THE NEXT COMMAND FROM THE APPLICATION.

\*\*\*\*\*  
\*\*\*\*\*

# REBOOT

AFTER THE APPLICATION SENDS A **D0** HEX TO THE MOUSEMAT, THE MOUSEMAT WILL REBOOT. THE MOUSEMAT NOW GOES TO **IDLE LOOP 1** AND WAITS FOR THE NEXT COMMAND FROM THE APPLICATION.

Document Number DI-044	Page 24 of 24	Revision D
------------------------------	------------------	---------------