Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

ib technology

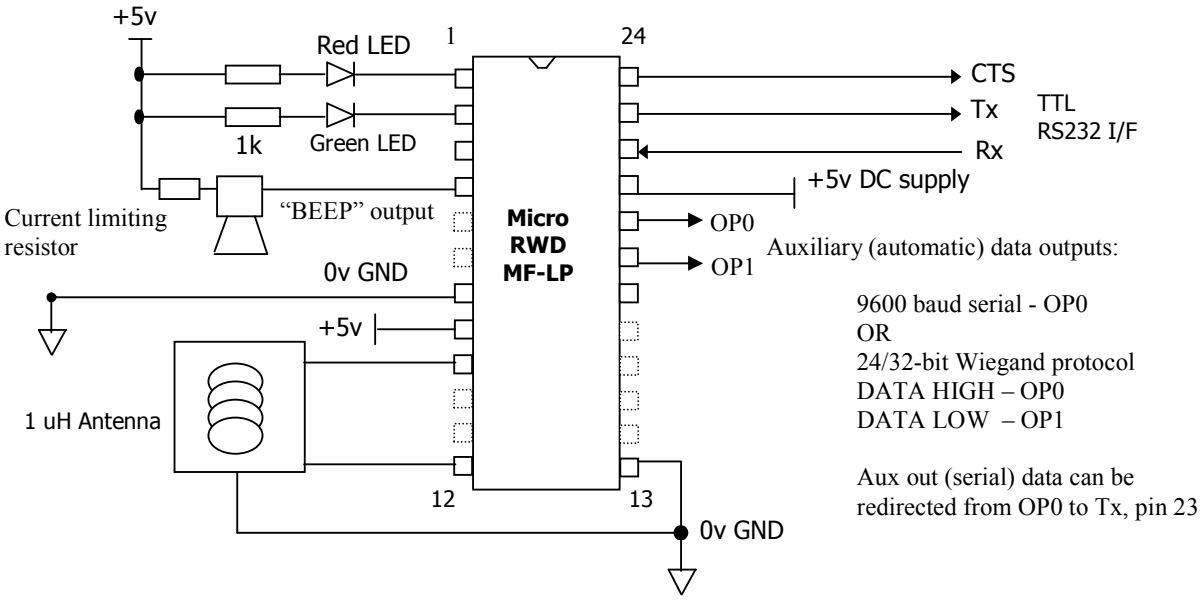
Data Sheet

MFPROT_LP.pdf
35 Pages
Last Revised 09/08/11

Micro RWD MF (Mifare) Low Power Version
(with auxiliary data outputs)

The MicroRWD MF LP (Low Power) module is a complete read and write solution for 13.56 MHz Mifare Classic cards (1k, 4k and Ultralight versions) and supports "Mifare" contactless operations to dual-interface cards such as Mifare ProX, Smart MX (JCOP) and other types. DESfire and Mifare PLUS cards are supported for serial number acquisition only. The solution is entirely housed within a 24-pin DIL package and only needs an antenna connected and a 5v DC supply to be a fully featured ISO14443A Mifare read/write system. The MicroRWD MF LP version behaves in the same manner as the standard reader except that it has an **active, average current consumption of less than 150µA (micro Amps)** with 1 second polling rate. As on other RWD modules, all commands and data response are via a simple TTL level RS232 interface. In addition, the RWD MF LP version has **auxiliary data outputs on the OP0 / OP1 pins** that can be programmed to automatically output UID (serial number) or other block data as **asynchronous 9600 baud** serial or **Wiegand protocol Data High / Data Low signals**. All these features can be configured and turned ON/OFF by setting RWD EEPROM parameters. The diagram below shows the pin out configuration for the MicroRWD MF LP module.

Micro RWD MF LP module connections



With the ultra-low-power current consumption and the additional auxiliary data output features, this one of the most compact and flexible reader systems available.

Auxiliary Data Output

The Micro RWD MF LP version uses the 4-byte UID (serial number) or the least significant (first) 4 bytes of data from a Mifare card memory block to create a 32-bit data frame. The data frame can then be output as asynchronous 9600 baud serial data on OP0 pin or as 24 / 32 bit Wiegand protocol with parity bits attached (making 26 or 34 bits of data) on OP0 / OP1 pins.

An RWD EEPROM parameter can redirect the serial auxiliary output on OP0 (pin 20) to the main TX output (pin 23). This allows both bi-directional command/data communication and the automatic auxiliary serial data output with the same 3-wire RS232 interface.
Note that when the auxiliary serial output has been redirected to TX pin, there will be NO acknowledgement or data response to commands (to avoid confusion of data).

For normal command and data response, the serial auxiliary output MUST be directed to the OP0 pin or turned OFF.

The auxiliary data outputs on OP0 / OP1 are **AUTOMATIC** and if enabled, occur when a card enters the RF field for the first time. The "BEEP" output signal delay, data source, byte order and Hex/ASCII format for the auxiliary output and the various Wiegand protocol options are all controlled by programmable RWD EEPROM parameters (see page 8). A zero data length parameter effectively turns the auxiliary outputs OFF.
In this manner the MicroRWD can be used in battery powered application (down to 150µA average current consumption) and automatically output blocks of data (such as the UID) WITHOUT any commands being sent to the module. In addition, the "Green" LED output or the BEEP output can be used as a control signal to "interrupt" the host computer or microcontroller just before the automatic data is transmitted.

NOTE that the "BEEP" output (RWD pin 4) idles in a high state and "sinks" current. External loads can be connected between the supply rail and pin 4 with a series resistor to ensure "sink" current does not exceed 25ma.
Note that setting Polling rate parameter to minimum value (0x00) means the polling rate is always as fast as possible and does not change ("SLEEP" and power-down is skipped).

MicroRWD MF (Mifare) operation

The Mifare transponder cards have significantly more memory than most other cards and the 13.56 MHz carrier frequency provides fast transaction times of 106 kbaud. The cards are available with 64 bytes (Mifare Ultralight), 1024 bytes (Mifare 1k) and 4096 bytes (Mifare 4k) of memory. For the 1k and 4k cards the memory is organised as 16 and 40 Sectors respectively, each Sector has 64 bytes arranged as 4 Blocks of memory (3 of which are available for general Read/Write use). Each Sector can be separately locked/unlocked for access using security keys. Initial communication with the cards can only proceed after mutual authentication between the RWD and the card has succeeded (as defined by ISO 14443A standard). The Mifare cards are ideally suited to Electronic-Purse applications such as ticketing and vending applications where each sector can hold entirely separate data for different applications.

The MicroRWD MF LP is a proximity system and a Read/Write range of up to 10cm can be achieved under ideal conditions using the appropriate antenna. For evaluation purposes the RWD is available on a base board with PCB antenna, LEDs, power-regulation, 9-pin RS232 and optional USB interfaces. When power is first applied to the board the red and green LEDs flash once to indicate successful power-up (both LEDs stay on if initialisation fails). The RWD can also check for antenna faults and internal error conditions, these problems are indicated by the red LED or both LEDs flashing continuously until the fault has been rectified.

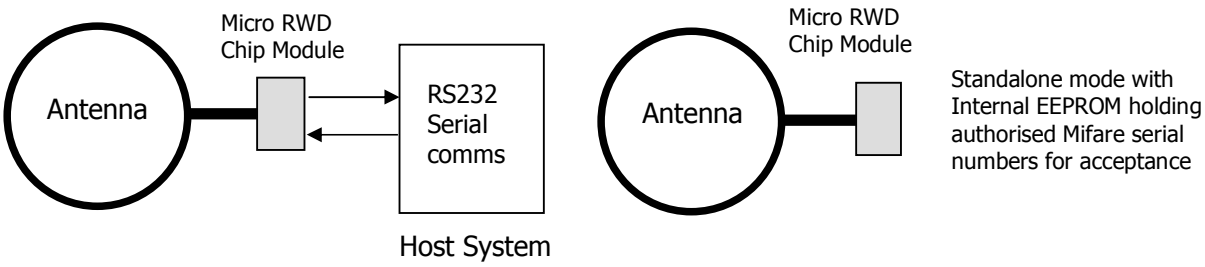
The RWD will normally have the red LED lit until a valid card is brought into the RF field. If the tag is accepted as valid then the green LED is turned ON (and red LED OFF). If the auxiliary output features are enabled then the UID (serial number) is acquired, or block data is internally read and transmitted as serial data or Wiegand protocol data on OP0/OP1 pins.
If the Beep delay is set then the "BEEP" output (pin 4) is pulsed ON/OFF. With auxiliary output features turned OFF, the RWD responds to host commands on the TTL serial interface at 9600 baud, 8 bits, 1 stop, no parity, as usual.

Note: Some ISO14443A compliant cards have a SINGLE (4 byte) UID and others have a DOUBLE (7 byte) UID. These serial numbers are acquired as part of the initial anticollision/select procedure when a card is brought into the field. This UID information can be reported using the CARD UID command. For Mifare Classic 1k and 4k cards the SINGLE UID is acquired and can be reported BUT subsequent block read/write operations will not function if the Security KEY is incorrect for the particular sector and the authentication fails.

For many applications where only the UID (serial number) is required, the Security KEY information is therefore NOT required.

MicroRWD MF modes of operation

The Micro RWD has two basic modes of operation:-



Remote mode (connected to a host computer or microcontroller) and Standalone mode.

1) Remote mode involves connecting to a host serial interface. This is where the stored list of authorised identity codes (serial numbers) can be empty, effectively authorising any Mifare card for subsequent read/write operations (depending on correct Security Key. A simple serial protocol allows a host system to communicate with the Micro RWD in order to program new authorised identity codes, change parameters, load Security Keys and perform Read/Write operations to the card itself.

2) Standalone mode is where the Mifare card identity codes (serial numbers) are checked against a stored list of authorised codes. If an identity code is matched, the output drives and Green LED are enabled. Effectively standalone mode occurs when there is no host system communicating with the Micro RWD. Up to 60 serial numbers can be stored in the authorisation list so this mode of operation can be used to create a "mini access control" system.

   The automatic auxiliary outputs can also be used to send serial number or other card memory block data to a controller with no command interaction required.

Supported transponder types

The MicroRWD MF is designed to communicate with the following passive RF transponder types. Note that Mifare cards and transponder devices are fabricated by several companies under licence from Philips/NXP Semiconductors. They are all fully Mifare compliant and only differ in having different default Security Keys loaded in the factory:-

1) Mifare standard 1k card (MF1 IC S50 transponder) and Infineon equivalent.

2) Mifare standard 4k card (MF1 IC S70 transponder)

3) Mifare Ultralight card (MF0 IC U1 transponder)

4) Mifare ProX, Smart MX (JCOP) dual-interface card types are supported to allow single or double UID to be acquired and "Mifare" operations performed across the contactless interface. DESFire, Mifare PLUS supported for serial number acquisition.

5) Any ISO 14443A compliant contactless card can be accessed for Serial Number. Full Read/Write access will only be possible if card fully supports Philips/NXP Semiconductors CRYPTO1 algorithm and encrypted data protocols.

The operation of the Micro RWD MF and the Mifare transponders is described in more detail at the end of this document.

The identification codes described in this text are regarded as the four-byte Mifare SINGLE UID (Unique Identifier/serial number) or the least significant four bytes of the seven-byte Ultralight DOUBLE UID.

Serial Interface

This is a basic implementation of RS232. The Micro RWD does not support buffered interrupt driven input so it must control a BUSY (CTS) line to inhibit communications from the host when it is fully occupied with card communication. It is assumed that the host (such as a PC) can buffer received data. This CTS signal must be connected to the host computer communication port to allow "hardware handshaking" or the host driver software must check the CTS signal and only send commands/data when it is in a LOW state. The CTS signal is pulsed LOW for a 6ms period each polling cycle. The host computer must wait for this LOW signal and then send the command and data.

The CTS line remains in a LOW state while the command and data bytes are being received. After the last byte of data the CTS signal "times out" for 6ms and returns HIGH.
This 6ms "window" every polling cycle allows the host computer to send a single command and associated data to the RWD. Please note that only one command and it's corresponding data bytes can be sent during a CTS LOW period, the command and data bytes must be sent with no gaps between, if there is a pause of more than 6ms between bytes then "time out" occurs, the CTS line returns high and the command fails (flagged as RS232 error). The CTS signal idles in this HIGH state (to inhibit host communication) until the next polling cycle begins.

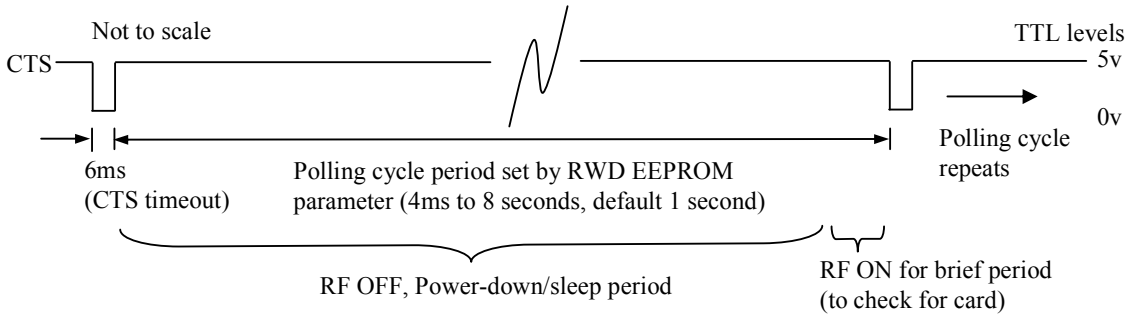
The communication baud rate is 9600 baud, 8 bits, 1 stop, no parity. The RWD Tx, Rx and CTS signals are all TTL level and can be converted to +/-10v RS232 levels using a level converter device such as the MAX202 (note the inversion of the TTL levels).

The Micro RWD MF LP (low-power) version has been specifically designed to operate with very low average power consumption but still remain responsive to cards entering and leaving the field and be able to read large amounts of data as quickly as possible.

THE RWD HAS THREE POLLING STATES:

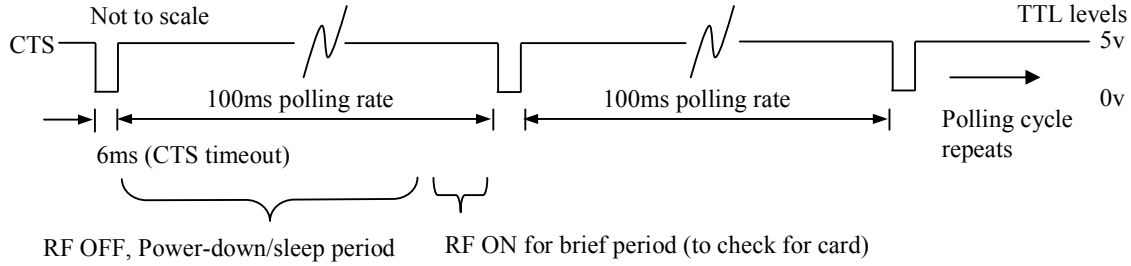
1) **NO card present and NO host commands received.**

   Polling cycle rate (time between subsequent CTS low periods) is determined by the "polling rate" parameter stored in the RWD EEPROM memory. This is typically set to a long period (4ms to 8 seconds, default setting 260mS) and is the primary means to reduce average power consumption. This is because most of the polling cycle period is spent in a power-down/sleep mode.



2) **Mifare card in field, NO host commands received**.

   When a card is detected in the field the polling rate changes to approximately 100ms (between CTS low periods). This is to ensure that the RWD can respond quickly to the card leaving the field and a new card being presented.



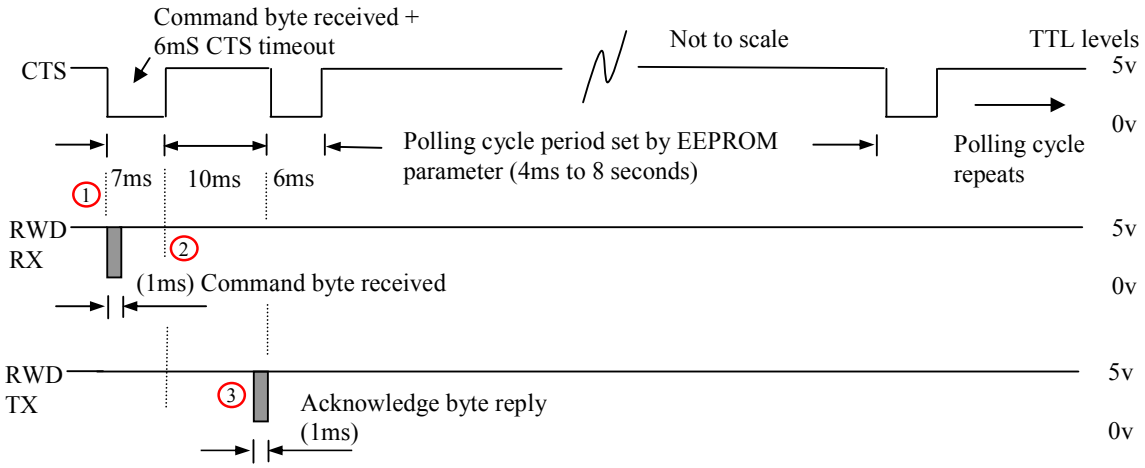
3) **Host commands received and processed.**

When the RWD receives commands from the host computer, the polling rate increases to allow a quick response to the command. This means that commands such as READ or WRITE BLOCK can be repeated quickly and the large amounts of data read from, or written to the card as fast as possible.

The polling cycle delay in this case is effectively the minimum, so the RWD responds to the host command immediately after the RF communication is complete.

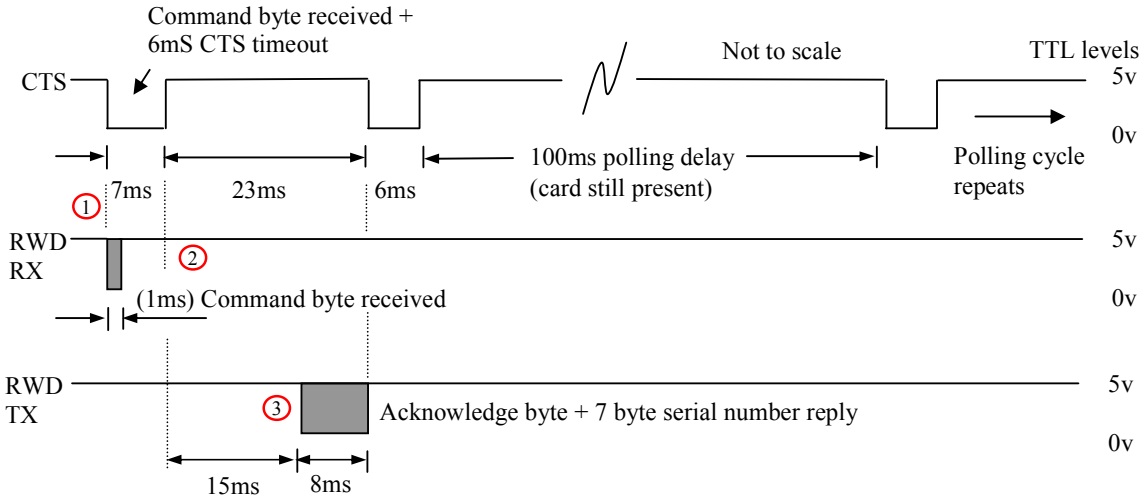
Example a) NO card present, single CARD UID (0x55) command received.

Note: at 9600 baud serial communication rate, a single byte is received or transmitted in approximately 1mS (104µS per bit). If no commands follow then the polling rate reverts back to the stored parameter value as in (1).

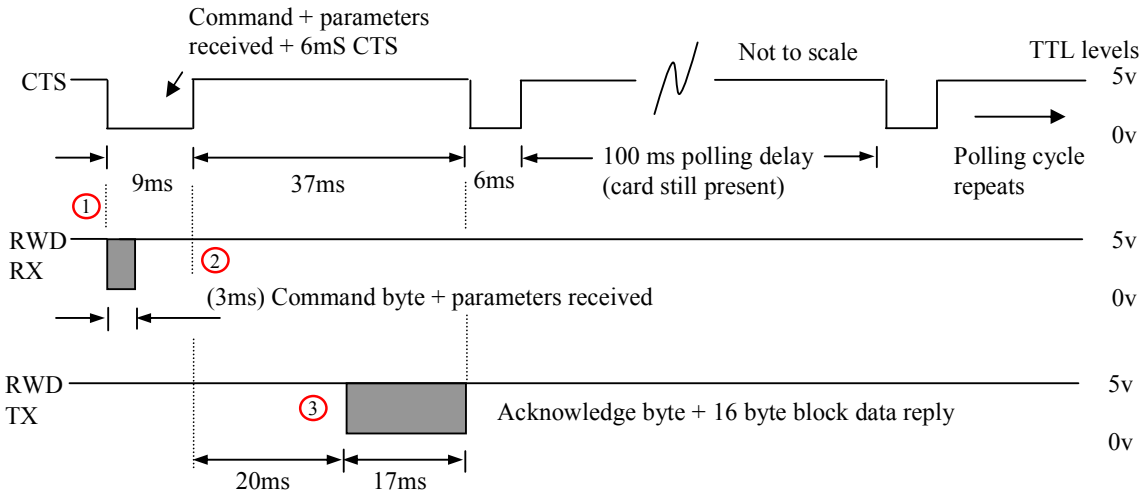


1. Host waits for CTS falling edge then sends command byte.
2. RWD processes command, RF turned ON for brief period to check if card present.
3. RWD then replies with acknowledge byte (+ data).

Example b) Mifare card in field, single CARD UID (0x55) command received.

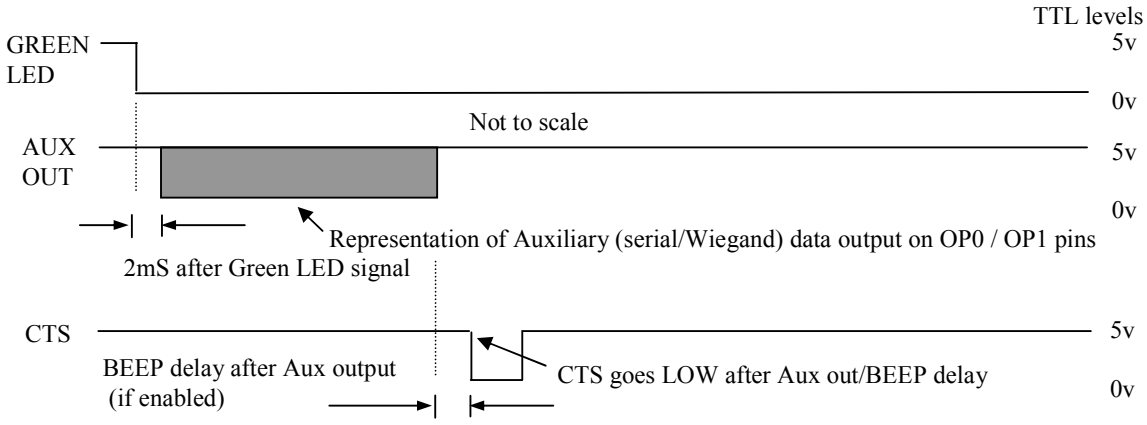


Example c) Mifare card in field, valid READ BLOCK command received (Read cmd (0x52) + Keycode number + Block number)



Auxiliary output and BEEP delay timing (if options are enabled)

Mifare card in field for first time, Auxiliary output enabled and BEEP delay set. Green LED signal can be used as an interrupt signal to the host to indicate that auxiliary data will follow.



Summary of Polling rates and command timing

Three polling rates:

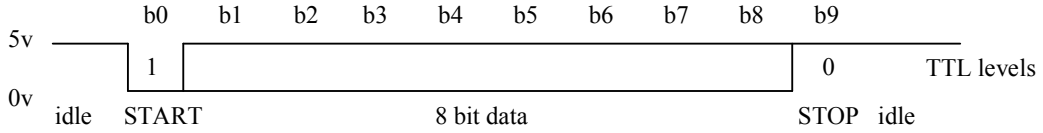
1) NO card and NO commands: Polling rate determined by Polling rate parameter in RWD EEPROM (4mS to 8 seconds, default setting 260mS)
2) Card present but NO commands: 100ms polling delay between CTS pulses.
3) Command (and parameters) received: 10ms polling delay to next CTS pulse.

For lowest power consumption, the Polling rate parameter in EEPROM is typically set to a long period (> 1 second). Auxiliary output (if enabled) occurs after Green LED signal and before CTS.

Host communication software must be able to handle the three polling rates.

Note that Auxiliary outputs (and "BEEP" output) should be turned OFF if standard RS232 command interface is being used to ensure minimum power consumption and no additional delays occur in the polling loop.

Transmitted or Received data byte, 9600 baud, 8 bit, 1 stop, No parity (104 µS per bit)



Host Driver software

Communication with the MicroRWD module is via the TTL level RS232 interface (9600 baud, 8 bit, 1 stop bit, no parity) and uses the CTS line for hardware handshaking. The Windows applications (supplied with the Evaluation kit) can be used to communicate with the module or the user can write their own application on a PC or a microcontroller. Please note that the host software must be able to handle the three distinct polling rates (different periods between CTS pulses). The following basic communication algorithm can be used:-

Typical host computer "pseudo" driver code

```
if (Green LED ON (pin 2 = 0))                  // Optional check for valid tag in field
{
   if (CTS = 0)          // Wait for CTS = 0 (RWD ready to receive command / data)
   {
            // CTS times out after 6ms so command and all parameters must be sent with no-
            // gaps otherwise CTS times out and goes HIGH.
            // For example, send READ BLOCK 1 using KEY 0 as KEYA (0x52 0x01 0x00)

            SEND_CMD( );              // Sent command + parameters to RWD

            // RWD sets CTS = 1 after last parameter received. RWD module processes
            // command, turns on RF for short period, then sends reply.

            GET_REPLY( );             // Get Acknowledge byte + data
            // Response to READ command is 0x80 (no tag) or 0x86 + sixteen bytes of DATA.
   }
}
```

Command Protocol

The following commands are supported. The corresponding acknowledge code should be read back by the host and decoded to confirm that the command was received and handled correctly. The serial bit protocol is 9600 baud, 8 bits, 1 stop, no parity (lsb transmitted first).

The status flags returned in the Acknowledge byte are as follows:

```
 b7  b6  b5  b4  b3 b2 b1 b0
 1   1   1   1   1  1  1  1
     |   |   |   |  |  |   EEPROM error (Internal EEPROM write error)
     |   |   |   |  |  Card OK (Card serial number matched to identity code list)
     |   |   |   |  Rx OK  (Card  communication and acknowledgement OK)
     |   |   |   RS232  error (Host serial communication error)
     |   |   MF type (0 = MF 1k byte card, 1 = MF 4k byte card)
     |   UL type (0 = MF standard 1k/4k card, SINGLE UID), 1 = MF Ultralight card, DOUBLE UID)
     MFRC error (Internal or antenna fault)
```

Note that bit 7 is fixed so that using a Mifare 1k card, the RWD acknowledge response to a valid host command would generally be 86 (Hex), indicating that a matched (or authorised) MF 1k card is present. The MF Ultralight card has a different memory structure to the standard 1k/4k MF cards so bits 4 and 5 have to be checked to determine which card type is present. Note also that only the relevant flags are set after each command as indicated in the following specification.

Card UID

Command to return card status and UID (Unique Identifier or Serial number).
The acknowledge byte flags indicate general Mifare card status.

	B7 B6 B5 B4 B3 B2 B1 B0	
Command:	0 1 0 1 0 1 0 1	(Ascii "U", 0x55)
Acknowledge:	1 F F F F F F X	(F = Status flags)

Data only follows if card was selected OK with no errors detected.

Reply1:	D D D D D D D D	(D = LS Byte of UID/Serial number from card)
Reply2:	D D D D D D D D	
Reply3:	D D D D D D D D	
Reply4:	D D D D D D D D	
Reply5:	D D D D D D D D	Dummy bytes (0x00) for Mifare 1k/4k card types
Reply6:	D D D D D D D D	
Reply7:	D D D D D D D D	

Note that Mifare 1k and 4k cards have a four-byte serial number but Mifare Ultralight cards have a seven-byte serial number. To accommodate all card types, the Card UID command returns a seven-byte field with the last three bytes padded out with 0x00 dummy bytes in the case of Mifare 1k/4k cards.

Card STATUS

Command to return card status.
The acknowledge byte flags indicate general Mifare card status.

	B7							B0	
Command:	0	1	0	1	0	0	1	1	(Ascii "S", 0x53)
Acknowledge:	1	F	F	F	F	F	F	X	(F = Status flags)

Program EEPROM

The Micro RWD has some internal EEPROM for storing system parameters such as polling rate and authorised identity codes (serial numbers). This command sequence allows individual bytes of the EEPROM to be programmed with new data. The data is internally read back after programming to verify successful operation. Note that due to the fundamental nature of these system parameters, incorrect data may render the system temporarily inoperable.

	B7							B0	
Command:	0	1	0	1	0	0	0	0	(Ascii "P", 0x50)
Argument1:	N	N	N	N	N	N	N	N	(N = EEPROM memory location 0 - 255)
Argument2:	D	D	D	D	D	D	D	D	(D = data to write to EEPROM)
Acknowledge:	1	X	X	X	F	X	X	F	(F = Status flags)

Internal EEPROM memory map

Polling delay parameter values (EEPROM location 0):

Parameter 0 value	Polling Delay SLEEP Period	
0x00	0 mS	(SLEEP and power-down is skipped)
0x10	8 mS	
0x20	16 mS	
0x30	32 mS	
0x40	65 mS	
0x50	132 mS	
0x60	**262 mS**	
0x70	524 mS	
0x80	1 second	
0x90	2 seconds	
0xA0	4 seconds	
0xB0	8 seconds	

Polling delay can be set from 0 to 8 seconds to give complete control over current consumption and battery life. Note that setting Polling delay = 0x00 skips the SLEEP and power-down operation so polling is as fast as possible (and current consumption is highest).

Byte 0: Polling Delay (SLEEP / Power down) period (default = 0x60 = approx 260 milliseconds)

Byte 1: Aux data output: 0x00 = OFF (NO output from OP0 / OP1),
0x01 = 24 (26) bit, Wiegand on OP0 / OP1.
0x02 = 32 (34) bit, Wiegand on OP0 / OP1.
0x03 = 9600 baud serial from OP0 (default)

Byte 2: Reserved (Checksum)

Byte 3: Mifare/ICODE option byte, MIFARE mode = 0x00 (default)
ICODE mode = 0x01 (NOT SUPPORTED ON THIS VERSION)

Byte 4: Wiegand parity option, NO Parity = 0x00 (default)
0x01 = Even/Odd parity added

Byte 5: Aux block address on card (Mifare card block address 0 – 255), default 0x01, block 1
(only used if parameter byte 8 is set to 0x01 for internal Block Read)

Byte 6: Key number / type used for internal Block Read of Aux data:
(TxxKKKKK), (T = Key type, 0 = KeyA, 1 = KeyB)
(K = Key code number, 0 - 31), default = key 0x00 used as typeA
(only used if parameter byte 8 is set to 0x01 for internal Block Read)

Byte 7: "Beep" delay parameter (x 40 mS) default = 0x00 (OFF)

Byte 8: Aux output source data selection.
0x00 = use UID / serial number (default)
0x01 = perform Block Read

Byte 9: Aux out (serial data) redirection (OP0 - pin 20 or Tx – pin 23)
0x00 = Serial aux output from OP0 pin (default)
0x01 = Serial aux output from main Tx pin

Byte 10: Aux output serial format (Hex or ASCII), HEX output = 0x00 (default)
ASCII output = 0x01

Byte 11: Aux output byte order option, plain data as read from card = 0x00 (default)
Byte order reversed = 0x01

Start of authorised card codes. List is terminated with FF FF FF FF sequence.
List is regarded as empty (all identity codes valid) if first code sequence in list is (FF FF FF FF).
List can hold up to 60 identity codes (serial numbers)

Byte 12: 0xFF Empty list
Byte 13: 0xFF
Byte 14: 0xFF
Byte 15: 0xFF

Byte 16: (MSB) Tag identity code
Byte 17:
Byte 18:
Byte 19: (LSB)
-
-
-
-
Byte 255: Last Internal EEPROM location

Note that the polling delay parameter must be a valid value (as shown in the table above), other values will give undefined results.

Default RWD EEPROM parameter settings:

Byte 0:	0x60,	260mS Polling delay / SLEEP period
Byte 1:	0x03,	Aux data output as 9600 baud serial on OP0
Byte 2:	Reserved	
Byte 3:	0x00	MIFARE mode (ICODE mode not supported on this version)
Byte 4:	0x00	Wiegand NO parity option (only used if Byte 1 = 0x01 / 02)
Byte 5:	0x01	Aux block address on card (only used if Byte 8 = 0x01)
Byte 6:	0x00	Key number / type used for internal Block Read of Aux data (Use Key Code 0 as Key Type A, only used if Byte 8 = 0x01)
Byte 7:	0x00	"Beep" output delay OFF
Byte 8:	0x00	Aux output source data is UID (serial number).
Byte 9:	0x00	Aux output (serial data) directed to OP0 pin.
Byte 10:	0x00	Aux output serial format, HEX byte format
Byte 11:	0x00	Aux data byte order, plain as read from card

Store Keys

The Micro RWD has additional internal storage for 32 Security KEYs. Six byte Key codes are required to access individual card sectors for any Read or Write operations. This command sequence allows 6 byte Key codes to be stored at any one of the 32 key code locations. Factory defaults are Infineon/Philips specified transport key code pairs (Hex FF FF FF FF FF FF / Hex FF FF FF FF FF FF)) and (Hex A0 A1 A2 A3 A4 A5 / Hex B0 B1 B2 B3 B4 B5) and these are stored in the RWD non-volatile memory during manufacture. Note that due to the fundamental nature of these Key codes, incorrect values may render the system inoperable. Only one or two Security key codes are required to unlock a card sector so the provision of 32 storage locations allows for many possible applications and card uses.
IT IS STRONGLY ADVISED THAT THE KEY CODES IN THE RWD AND STORED ON THE MIFARE CARD ARE NOT CHANGED UNTIL THE OPERATION OF THE MIFARE CARD SECURITY IS FULLY UNDERSTOOD.

	B7							B0	
Command:	0	1	0	0	1	0	1	1	(Ascii "K", 0x4B)
Argument1:	x	x	x	K	K	K	K	K	(K = Key code number, 0 - 31)
Argument2:	D	D	D	D	D	D	D	D	(D = data to write to EEPROM, LS byte)
Argument3:	D	D	D	D	D	D	D	D	
Argument4:	D	D	D	D	D	D	D	D	
Argument5:	D	D	D	D	D	D	D	D	
Argument6:	D	D	D	D	D	D	D	D	
Argument7:	D	D	D	D	D	D	D	D	(D = data to write to EEPROM, MS byte)
Acknowledge:	1	X	X	X	F	X	X	F	(F = Status flags)

Internal Key Storage memory map (default settings)

Location 0 (0x00): Key code 0 (Default 0xFF FF FF FF FF FF)
Location 1 (0x01): Key code 1 (Default 0xFF FF FF FF FF FF)

Location 2 (0x02): Key code 2 (Default 0xA0 A1 A2 A3 A4 A5)
Location 3 (0x03): Key code 3 (Default 0xB0 B1 B2 B3 B4 B5)
-
-
-
Location 28 (0x1C): Key code 28 (Default 0xFF FF FF FF FF FF)
Location 29 (0x1D): Key code 29 (Default 0xFF FF FF FF FF FF)

Location 30 (0x1E): Key code 30 (Default 0xA0 A1 A2 A3 A4 A5)
Location 31 (0x1F): Key code 31 (Default 0xB0 B1 B2 B3 B4 B5)

Note that Mifare cards manufactured by Infineon have default transport key codes of (0xFF FF FF FF FF FF) and Philips cards have (0xA0 A1 A2 A3 A4 A5 / 0xB0 B1 B2 B3 B4 B5) default transport keys. The MicroRWD MF has both pairs stored as default settings to allow ease of use when the system is first used. (More information on the Mifare card memory maps, Security Keys and the KeyA and KeyB types can be found at the end of this document).

Write Card Block

Command to write 16 bytes of data to specified Mifare block. A Block is made up of 16 bytes and there are four blocks in each card sector (sixteen blocks per sector in upper half of Mifare 4k card). Note that blocks 3, 7, 11, 15 etc are sector trailer blocks that contain Security Key data and Access bits. Writing incorrect information to these blocks can permanently disable the sector concerned. The first argument is the block number to write data to, the second argument specifies which key code (0 - 31 from the internal storage area) to use for sector authentication/unlocking and if the Security Key is to be used as a KeyA or KeyB type code. If the write was unsuccessful (invalid card, authentication failed or card out of field) then Status flags in acknowledge byte indicate error.

	B7 B0	
Command:	0 1 0 1 0 1 1 1	(Ascii "W", 0x57)
Argument1:	N N N N N N N N	(N = MF Card Block Address 0 – 255)
Argument2:	T x x K K K K K	(T = Key Type, 0 = KeyA, 1= KeyB) (K = Key code number, 0 – 31)
Argument3:	D D D D D D D D	(D = LS Byte of data to write to card)
Argument4:	D D D D D D D D	
Argument5:	D D D D D D D D	
Argument6:	D D D D D D D D	
↓		16 Bytes of data
Argument15:	D D D D D D D D	
Argument16:	D D D D D D D D	
Argument17:	D D D D D D D D	
Argument18:	D D D D D D D D	(D = MS Byte of data to write to card)
Acknowledge:	1 F F F F F F X	(F = Status flags)

Note that Mifare Ultralight cards DO NOT USE Security Keys or CRYPTO Authentication and the memory is organised differently as groups of 4 bytes (Pages). Only one Page of 4 bytes can be written at a time so to maintain compatibility and a simple RWD host command set, the same command as above is used to write data to Ultralight cards. The command and arguments have the same structure but different meanings. The "Block" address is treated as a "Page Address" and the KeyType/Key number parameter is a dummy 0x00 byte. In addition the 4 bytes of data are padded out to 16 bytes with dummy 0x00 bytes.

	B7 B0	
Command:	0 1 0 1 0 1 1 1	(Ascii "W", 0x57)
Argument1:	x x x x N N N N	(N = UL Card Page Address 0 – 15)
Argument2:	0 0 0 0 0 0 0 0	(Dummy byte, 0x00)
Argument3:	D D D D D D D D	(D = LS Byte of data to write to UL card)
Argument4:	D D D D D D D D	
Argument5:	D D D D D D D D	
Argument6:	D D D D D D D D	(D = MS Byte of data to write to UL card)
Argument7 – Argument18		
	0 0 0 0 0 0 0 0	12 Dummy padding bytes, 0x00
Acknowledge:	1 F F F F F F X	(F = Status flags)

Read Card Block

Command to read 16 bytes of data from specified Mifare block. The first argument is the block number to read data from, the second argument specifies which key code (0 - 31 from the internal storage area) to use for sector authentication/unlocking and if the Security Key is to be used as a KeyA or KeyB type code. If the read was successful, indicated by acknowledge status flags then sixteen bytes of block data follow.

	B7 B0	
Command:	0 1 0 1 0 0 1 0	(Ascii "R", 0x52)
Argument1:	N N N N N N N N	(N = MF Card Block Address 0 – 255)
Argument2:	T x x K K K K K	(T = Key Type, 0 = KeyA, 1= KeyB) (K = Key code number, 0 – 31)
Acknowledge:	1 F F F F F F X	(F = Status flags)

Data only follows if Read was successful

Reply1:	D D D D D D D D	(D = LS Byte of data Read from card)
Reply2:	D D D D D D D D	
Reply3:	D D D D D D D D	
Reply4:	D D D D D D D D	
↓		16 Bytes of data
Reply13:	D D D D D D D D	
Reply14:	D D D D D D D D	
Reply15:	D D D D D D D D	
Reply16:	D D D D D D D D	(D = MS Byte of data Read from card)

Note that as mentioned for the WRITE command, Mifare Ultralight cards DO NOT USE Security Keys or Authentication and the memory is organised differently as groups of 4 bytes (Pages). However, unlike the Write command, 16 bytes (4 pages) can be read in a single operation The same Read command as above is used except the "Block" address is treated as a "Page Address" and the KeyType/Key number parameter is a dummy 0x00 byte. For page numbers greater than 12, the card data wraps around to page 0 etc.

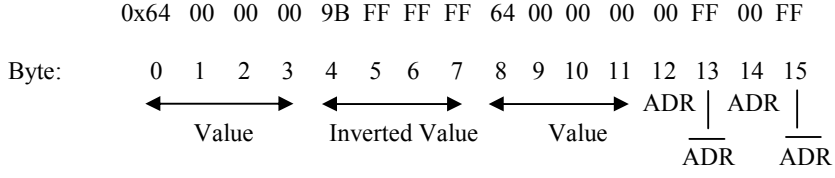
	B7 B0	
Command:	0 1 0 1 0 0 1 0	(Ascii "R", 0x52)
Argument1:	x x x x N N N N	(N = UL Card Page Address 0 – 15)
Argument2:	0 0 0 0 0 0 0 0	(Dummy byte, 0x00)
Acknowledge:	1 F F F F F F X	(F = Status flags)
Data only follows if Read was successful		
Reply1:	D D D D D D D D	(D = LS Byte of data Read from UL card)
Reply2:	D D D D D D D D	
Reply3:	D D D D D D D D	
Reply4:	D D D D D D D D	
↓		16 Bytes of data
Reply13:	D D D D D D D D	
Reply14:	D D D D D D D D	
Reply15:	D D D D D D D D	
Reply16:	D D D D D D D D	(D = MS Byte of data Read from UL card)

Inc Value (only operates on Value Data Structure)

Command to increment integer within a Value Data Structure. The command loads the value from the specified block address, adds the integer parameter and stores the result at the same or another block address. Note that the source block must have been formatted as a Value Block beforehand according to the data structure below, using the WRITE command. The INC Value command only operates on a "Value Block Structure" and will fail if the block configuration or the specified key type is incorrect.

Value Block Structure

Example format for value = 100 decimal (0x64), at block address 0.
(Value data stored LS byte first, ADR = block address, $\overline{ADR}$ = inverted block address)



The first argument is the source block address to load data from, the second argument specifies which key code and type to use for sector authentication (0-31 and if it is KeyA or KeyB type).

The third argument specifies the destination block address where the incremented data is stored. Note that source and destination blocks must be within same authenticated sector. The four byte positive integer to add follows (least significant byte first).

	B7 B0	
Command:	0 1 0 0 1 0 0 1	(Ascii "I", 0x49)
Argument1:	N N N N N N N N	(N = MF source block address 0 – 255)
Argument2:	T x x K K K K K	(T = Key Type, 0 = KeyA, 1= KeyB) (K = Key code number, 0 – 31)
Argument3:	N N N N N N N N	(N = MF destination block address 0 – 255)
Argument4:	D D D D D D D D	(D = LS byte of integer to add)
Argument5:	D D D D D D D D	4 byte integer
Argument6:	D D D D D D D D	
Argument7:	D D D D D D D D	(D = MS byte of integer to add)
Acknowledge:	1 F F F F F F X	(F = Status flags)

Dec Value (only operates on Value Data Structure)

Command to decrement integer within a Value Data Structure. The DEC Value command operates as the INC command except the integer parameter is subtracted from the loaded value. The first argument is the source block address to load data from, the second argument specifies which key code and type to use for sector authentication (0-31 and if it is KeyA or KeyB type). The third argument specifies the destination block address where the decremented data is stored. Note that source and destination blocks must be within same authenticated sector. The four byte positive integer to subtract follows (least significant byte first).

	B7 B0	
Command:	0 1 0 0 0 1 0 0	(Ascii "D", 0x44)
Argument1:	N N N N N N N N	(N = MF source block address 0 – 255)
Argument2:	T x x K K K K K	(T = Key Type, 0 = KeyA, 1= KeyB) (K = Key code number, 0 – 31)
Argument3:	N N N N N N N N	(N = MF destination block address 0 – 255)
Argument4:	D D D D D D D D	(D = LS byte of integer to subtract)
Argument5:	D D D D D D D D	4 byte integer
Argument6:	D D D D D D D D	
Argument7:	D D D D D D D D	(D = MS byte of integer to subtract)
Acknowledge:	1 F F F F F F X	(F = Status flags)

Transfer Value (only operates on Value Data Structure)

Command to transfer (copy) Value Data Structure. The command loads the value from the specified block address and then stores the result at the same or another block address. As with INC and DEC commands the source block must have been formatted as a Value Block beforehand and the block addresses must be within same authenticated sector.

The first argument is the source block address to load data from, the second argument specifies which key code to use for sector authentication (0-31) and if it is a KeyA or KeyB code. The third argument specifies where the data is stored.

	B7							B0	
Command:	0	1	0	1	0	1	0	0	(Ascii "T", 0x54)
Argument1:	N	N	N	N	N	N	N	N	(N = MF source block address 0 – 255)
Argument2:	T	x	x	K	K	K	K	K	(T = Key Type, 0 = KeyA, 1= KeyB) (K = Key code number, 0 – 31)
Argument3:	N	N	N	N	N	N	N	N	(N = MF destination block address 0 – 255)
Acknowledge:	1	F	F	F	F	F	F	X	(F = Status flags)

If the Inc, Dec or Transfer function was unsuccessful (invalid card, card out of field, authentication failed or data structures are incorrect) then Status flags in acknowledge byte indicate error. Note that the value manipulation commands operate internally on the Mifare card and no data is transferred back to the MicroRWD. Note also that Ultralight cards do not support Value Data Structures or the Inc, Dec, Transfer commands.

Type Identification

Command to return the **ATQA** (Answer to Request, Type A) two-byte codes and the **SAK** (Select Acknowledge) single-byte code after the complete UID has been acquired. As part of the initial communication with the Mifare card (as defined by ISO 14443A specification), the Mifare transponder responds to REQA (Request Command, Type A) with ATQA. The two-byte ATQA contains information that allows particular transponder types to be indentified. Following on from this the Mifare transponder responds to the SELECT (Select Command, Type A) with SAK (Select Acknowledge, Type A). The SAK code is a single byte value that contains further information about the type of transponder and the length of the UID. The SAK value reported is the final value after all "cascade levels" and the complete UID has been acquired.

NOTE THAT THIS COMMAND IS INCLUDED FOR DIAGNOSTIC PURPOSES TO ALLOW THE USER TO DETERMINE THE EXACT TYPE OF MIFARE CARD PRESENT IN THE FIELD, IF REQUIRED.

	B7							B0	
Command:	0	1	1	1	1	0	0	0	(Ascii "x", 0x78)
Acknowledge:	1	F	F	F	F	F	F	X	(F = Status flags)

Data only follows if card was selected OK with no errors detected.

Reply1:	D	D	D	D	D	D	D	D	ATQA - MSB
Reply2:	D	D	D	D	D	D	D	D	ATQA - LSB
Reply3:	D	D	D	D	D	D	D	D	SAK

	MF UL	MF 1K	MF 4K	MF DESFire	MF Prox	MF Prox	MF Prox	MF Prox	MF Prox	MF Prox
ATQA - MSB	0x00	0x00	0x00	0x03	0xXX	0xXX	0xXX	0xXX	0xXX	0xXX
ATQA - LSB	0x44	0x04	0x02	0x44	0x08	0x04	0x02	0x48	0x44	0x42

	Smart MX	Smart MX	Smart MX	Smart MX	Smart MX	Smart MX
ATQA - MSB	0xXX	0xXX	0xXX	0xXX	0xXX	0xXX
ATQA - LSB	0x08	0x04	0x02	0x48	0x44	0x42

	MF UL	MF 1K	INFINEON 1K	MF 4K	MF DESFire	MF ProX	MF ProX	MF ProX	MF ProX	MF ProX	MF ProX
SAK	0x00	0x08	0x88	0x18	0x20	0x20	0x08	0x28	0x00	0x20	0x08

	MF ProX	MF ProX	MF ProX	Smart MX	Smart MX	Smart MX	Smart MX	Smart MX	Smart MX
SAK	0x28	0x18	0x38	0x00	0x20	0x08	0x28	0x18	0x38

Note that many of the "extended" Mifare types are complex dual interface cards with embedded microcontrollers running "chip and pin" applications. Depending on the card type the memory map and protocol of the "extended" Mifare card may be different to the standard card types. In these cases the MicroRWD MF will report the UID using the Card UID command but read/write operation MAY NOT be fully supported.

Message

Command to return product and firmware identifier string to host.

	B7							B0	
Command:	0	1	1	1	1	0	1	0	(Ascii "z", 0x7A)

Reply: "m IDE RWD Mifare ICODE (SECMFI_CWAX_LP V1.xx DD/MM/YY) copyright: IB Technology (Eccel Technology Ltd)" 0x00

Returned string identifies product descriptor, project name, firmware version no. and date of last software change together with IB Technology disclaimer message. Note that the string is always NULL terminated. The string begins with a unique lower case character that can be used to identify a particular version of Micro RWD.

Factory Reset

Command to restore Factory default EEPROM values and Stored Keys and perform hardware Reset operation. The 0x55 0xAA parameters protect against accidental operation.
After Reset, the Red LED will flash 5 times indicating the successful loading of the Factory default values.

	B7							B0	
Command:	0	1	0	0	0	1	1	0	(Ascii "F", 0x46)
Argument1:	0	1	0	1	0	1	0	1	0x55
Argument1:	1	0	1	0	1	0	1	0	0xAA

Reset occurs after the command is processed so there is no Acknowledge byte reply.

Addition Notes for Commands

NOTE also that for the "Read Card Block" or "Card UID" command, if an error flag has been set in the Acknowledge code then there will be NO following data.

NOTE that the serial communication uses hardware handshaking to inhibit the host from sending the Micro RWD commands while Card communication is in progress. The serial communication system and protocol allows for a 6ms 'window' every Card polling cycle indicated by the CTS/BUSY line being low. During this 'window' the host must assert the first start bit and start transmitting data. The CTS/BUSY goes high again 6ms after the last stop bit is received. NOTE that only one command sequence is handled at a time. The period between the CTS pulses (polling delay) can have three rates depending on whether a card is present or not and if commands are being received by the RWD

NOTE that the commands and parameters must be sent to the RWD with no gaps otherwise communication timeout occurs and the RWD enters the polling delay period (the command string would then be incomplete and an RS232 error is flagged).

NOTE that the MicroRWD MF LP version performs fast polling cycles as long as there are commands to be processed. As soon as the commands stop being sent or the gap between sending commands is too long then timeout occurs and the polling delay increases to 100ms (if the card is still present) or the typically longer period (as set by EEPROM parameter) if there is no card. This is to allow repeated commands to be handled quickly such as for a "complete card read" where repeated Read Block commands are sent to the RWD.

Commands sent infrequently will have the full polling delay between each CTS/BUSY period.

Basic RWD Communication

For basic operation of the MicroRWD connected to a host computer, the RWD can either be polled or communication can be triggered by an interrupt signal (Green LED output). In either case the user would initially send the "STORE KEYS" command to load a custom security key into the RWD memory or simply use the pre-loaded default key values.

For a polling technique, the host computer would then keep sending the “STATUS” or “CARD UID” command and would monitor the acknowledgement code until a valid Mifare card was detected.

For the interrupt technique, the Green LED output can be used as an interrupt signal connected to the host computer. The Green LED output is normally high and goes low only when a valid card has been detected. This falling-edge signal can trigger a host interrupt to then send the “STATUS” or “CARD UID” command to determine the card type and serial number.

In both cases once a valid card has been detected a “READ BLOCK” or “WRITE BLOCK” command can be sent and the acknowledge code monitored to establish that the operation was successful.

Method of Operation

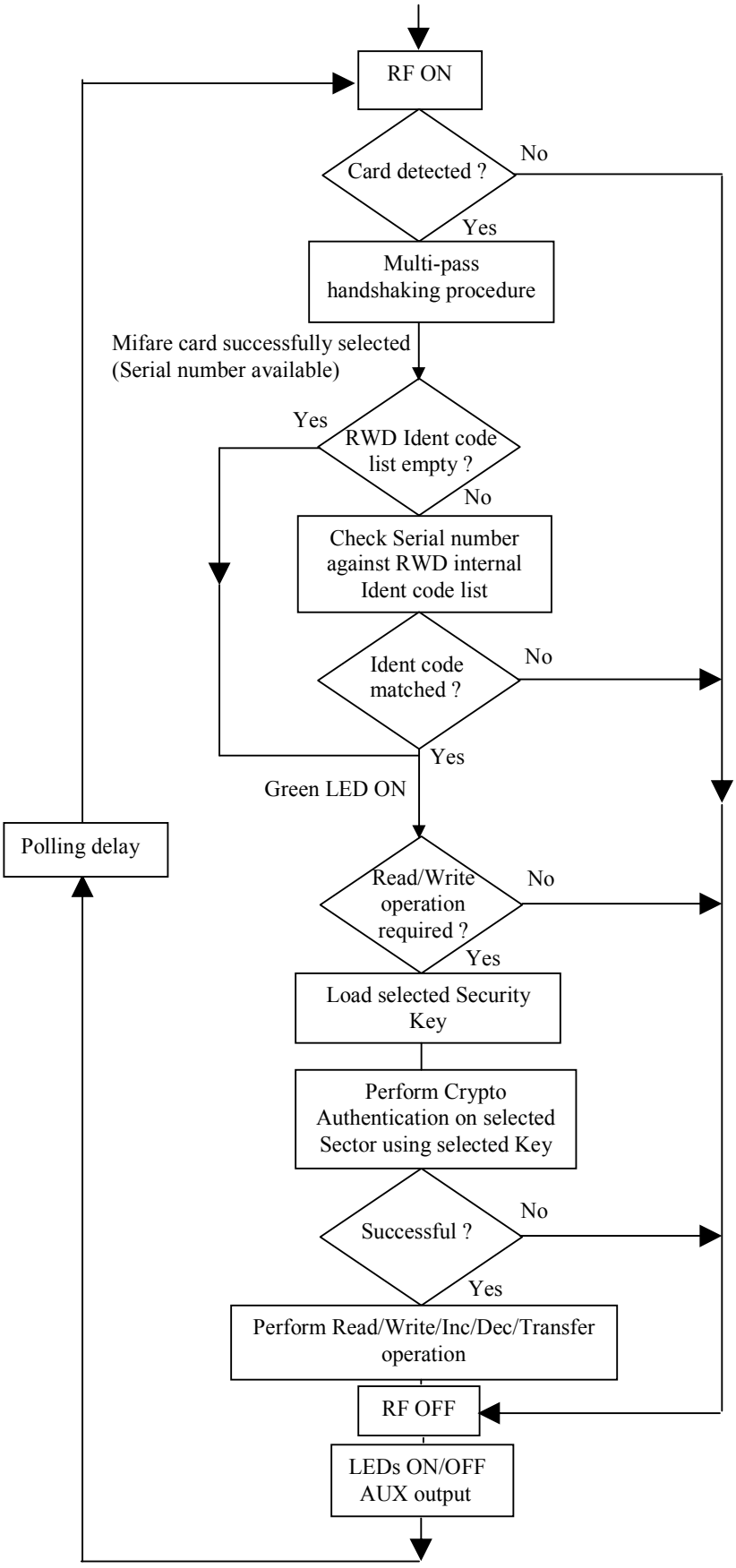
The system works on a polling principle whereby the RF field is turned on for a short period to check if a card is present. Authentication and Read/Write operations can then be performed before the RF field is turned off again and the process repeats. The polling principle where the RF is off most of the time allows for low average power consumption.

When a Mifare card is detected in the field a multi-pass handshaking procedure takes place where card information and serial number data is exchanged and checked for integrity. Once this procedure has completed successfully an individual card has been selected and is available for other operations.

The RWD itself has the additional feature of then checking the four byte Mifare UID/serial number (or least significant four bytes of Ultralight UID) against an internal authorisation list. The RWD internal EEPROM contains a list of four byte Identity codes (up to 60 of them) located from byte 12 onwards. If the list has FF FF FF FF (hex) stored at the first location (EEPROM bytes 12 - 15) then the list is treated as empty so the Identity code check is skipped.
Otherwise the card serial number is checked against all the entries in the list (until the FF FF FF FF termination code is reached) and if matched then the RWD allows the card to be accessed for other operations. If not the Red LED remains on and the card is blocked for further access. This is an additional level of security that can be used as a “mini access control” system for simple applications that only involve the serial number or where the Security Keys are not known.

Once the RWD has selected the card and has matched the serial number against it’s internal list (or the list is empty) then the Read/Write (or Inc/Dec/Transfer) operations can be performed. These require an internal high-security Authentication Crypto algorithm to take place that use the supplied Security Keys to gain access to a particular sector. If the Key selected does not match the Key stored in the Mifare card sector then the operation fails and the Red LED is turned on again.

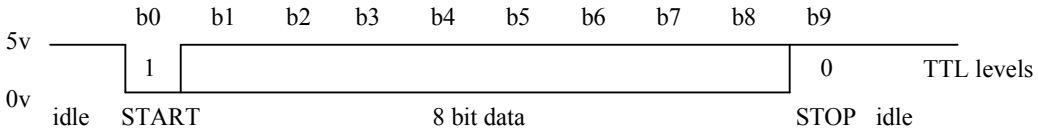
So in summary, a card can be successfully selected but can be blocked by the RWD authorisation list and fail Read/Write operations because the Keys are incorrect. Even if the Security Key is incorrect the Serial number can still be read using the “Card UID” command.


RF ON
Card detected ?
No
Yes
Multi-pass handshaking procedure
Mifare card successfully selected (Serial number available)
Yes
RWD Ident code list empty ?
No
Check Serial number against RWD internal Ident code list
Ident code matched ?
No
Yes
Green LED ON
Polling delay
Read/Write operation required ?
No
Yes
Load selected Security Key
Perform Crypto Authentication on selected Sector using selected Key
Successful ?
No
Yes
Perform Read/Write/Inc/Dec/Transfer operation
RF OFF
LEDs ON/OFF AUX output

Auxiliary Asynchronous Serial output

If selected, data can be automatically output from the **OP0 or main TX** pin as 4-bytes of data transmitted asynchronously at 9600 baud, 8-bits, 1 stop-bit, no parity. The data source can be selected as the 4-byte UID (serial number), the least significant 4-bytes of a double UID or the least significant (first) 4-bytes of a card memory block.

Data bytes transmitted at 9600 baud, 8-bits, 1-stop bit, No parity (104 µS per bit)



Auxiliary Wiegand Output Protocol

If selected, data can be automatically output from the **OP0 / OP1** pins as Data HIGH and Data LOW signals according to the Wiegand protocol.
The Wiegand protocol (24 bit data length) can be made up of a leading even parity bit (for b0 - b11), 24 bits of data (from transponder data) and a trailing odd parity bit (for b12- b23) creating a 26 bit output stream. The 32-bit mode has the same format except least significant four bytes of block data are used to form the data sequence. The parity bits are included or omitted and the byte order is reversed according to the EEPROM parameter settings.

For Example:-

Mifare block data (least significant 4 bytes): 0x04 60 22 12

(reversed byte option would use 0x12 22 60 04 as base data)

Wiegand 26 bit sequence:- E (b0 --------- b11) (b12 -------- b23) O

E (0 4 6 0 2 2) O

1 0000 0100 0110 0000 0010 0010 1

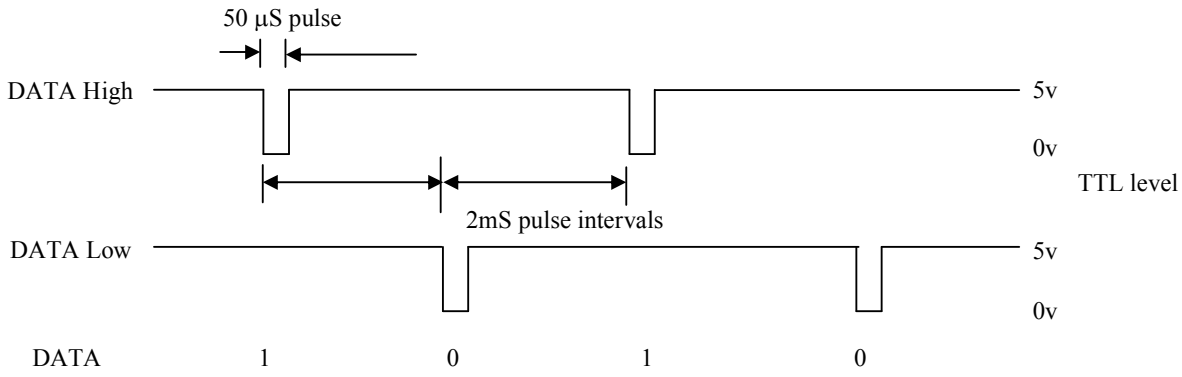
Where E is EVEN parity bit for bit 0 to 11 and O is ODD parity bit for bits 12 to 23

The base data for the Wiegand output can be the UID (least significant 4-bytes of serial number) acquired during the initial ISO14443A communication or the least significant 4-bytes of data from a block of card memory (acquired by an internal Block Read operation). Selection is by means of an RWD EEPROM parameter.
For the internal Block Read operation, the block number, key code number and type (KeyA or KeyB) are programmable EEPROM parameters. In addition, parameters control whether the base data byte order is reversed or if parity bits are added before output.

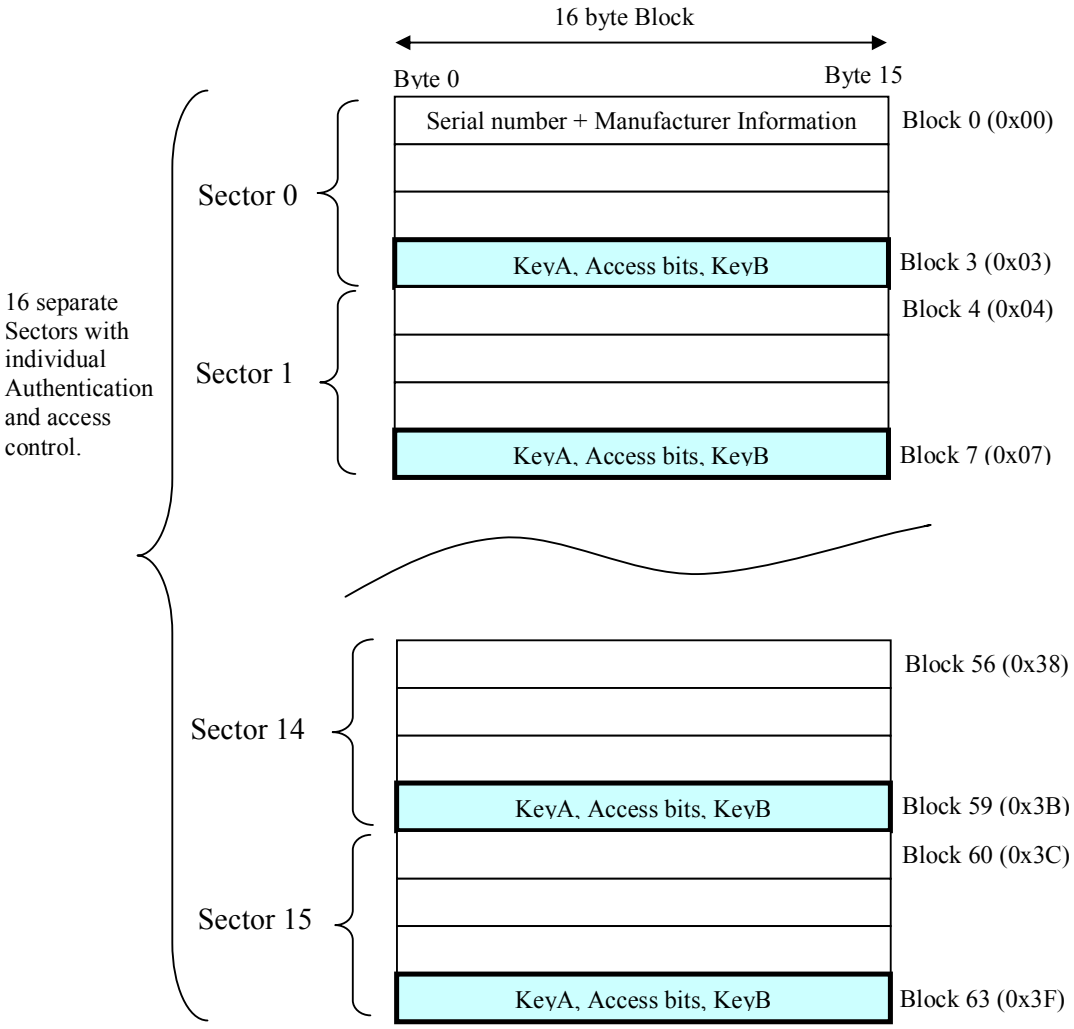
In this manner the user can select to use the UID (serial number) or user programmed information within a memory block as the base data for the Wiegand output. The complete data frame is output whenever the tag is within the RWD's antenna field and the tag has been validated. This output is independent of the normal TTL serial interface which responds to received commands and replies with the data as requested.

The physical Wiegand protocol is asynchronously transmitted as low going 50 µS pulses on the appropriate DATA low or DATA high pins. These pulses are separated by 2mS periods. The Wiegand sequence is output a single time whenever a valid tag enters the RF field for the first time. (NO Wiegand output if AUX OUTPUT parameter is ZERO/OFF).

Wiegand Protocol Timing Diagram



Mifare 1k (1024 byte) Memory Map

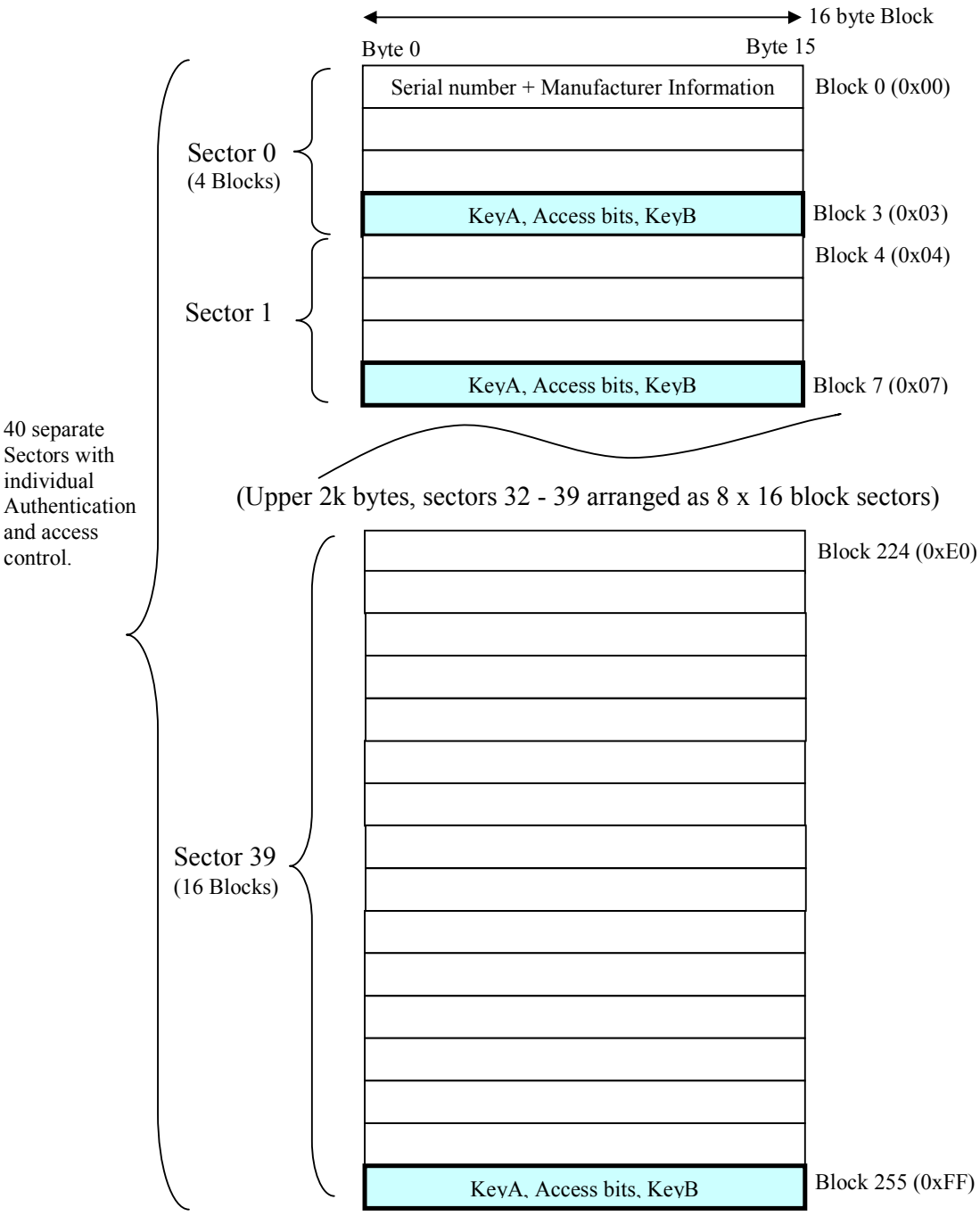


1024 byte memory is organised as sixteen sectors, each of which is made up of four blocks and each block is 16 bytes long. The first block in the memory (Block 0) is read-only and is set in the factory to contain the four-byte serial number (UID), check bytes and manufacturers data.

The last block of each sector (Blocks 3, 7, 11, 15……59, 63) is the Sector Trailer Block which contains the two security Key codes (KeyA and KeyB) and the Access bits that define how the sector can be accessed

Taking into account the Serial Number/Manufacturers Block and the Sector Trailer Blocks then there are 752 bytes of free memory for user storage. For all Read and Write operations the Mifare card memory is addressed by Block number (in hexadecimal format).

Mifare 4k (4096 byte) Memory Map



The lower 2048 bytes of the 4k card (sectors 0 – 31) are organised in the same way as the 1k card. However the upper 2048 bytes are organised as eight large sectors of 16 blocks each (sectors 32 – 39). Taking into account the Serial Number/Manufacturers Block and the Sector Trailer Blocks then there are 3440 bytes of free memory for user storage.