

Mifare Classic, Daten und Keys schreiben

1 Speicher-Layout Mifare Classic

Mikron Fare Collection System (Mikron Fahrgeld-System)

1.1 Speicheraufbau

1 kByte Speichergröße: 16 Sektoren mit jeweils 4 Blöcken à 16 Bytes

2 kByte Speichergröße: 32 Sektoren mit jeweils 4 Blöcken à 16 Bytes

4 kByte Speichergröße: 32 Sektoren mit jeweils 4 Blöcken à 16 Bytes + Sektoren mit jeweils 16 Blöcken à 16 Bytes

Der erste Block im ersten Sektor enthält die UID, Konfigurationsbits und interne Nummern.

Sektorinhalt (Standard)

Block#	Beschreibung			
0	Speicherblock mit 16 Bytes, kann nur im Ganzen gelesen oder geschrieben werden.			
1	Speicherblock mit 16 Bytes, kann nur im Ganzen gelesen oder geschrieben werden.			
2	Speicherblock mit 16 Bytes, kann nur im Ganzen gelesen oder geschrieben werden.			
3	Key A (6 Bytes)	Zugriffsbits (3 Byte)	1 Byte	Key B (6 Bytes)

Den letzten Block bezeichnet man auch als „Sector Trailer“.

Beispiel Sektor 30

Sector #30	Block #120	1920	1921	1922	1923	1924	1925	1926	1927	1928	1929	1930	1931	1932	1933	1934	1935
	Block #121	1936	1937	1938	1939	1940	1941	1942	1943	1944	1945	1946	1947	1948	1949	1950	1951
	Block #122	1952	1953	1954	1955	1956	1957	1958	1959	1960	1961	1962	1963	1964	1965	1966	1967
	Block #123	1968	1969	1970	1971	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981	1982	1983

Beispiel Sektor 37

Sector #37	Block #208	3328	3329	3330	3331	3332	3333	3334	3335	3336	3337	3338	3339	3340	3341	3342	3343
	Block #209	3344	3345	3346	3347	3348	3349	3350	3351	3352	3353	3354	3355	3356	3357	3358	3359
	Block #210	3360	3361	3362	3363	3364	3365	3366	3367	3368	3369	3370	3371	3372	3373	3374	3375
...																	
	Block #222	3552	3553	3554	3555	3556	3557	3558	3559	3560	3561	3562	3563	3564	3565	3566	3567
	Block #223	3568	3569	3570	3571	3572	3573	3574	3575	3576	3577	3578	3579	3580	3581	3582	3583

1.2 Passwörter (Key A/B)

Es gibt kein spezielles Sperrkommando, sondern es wird einfach in einem Speichersektor der 4. Block, der Key A (6 Bytes), die Zugriffs-Steuerungs-Bits (4 Bytes) und Key B (6 Bytes) enthält, überschrieben. Beim nächsten Reset (Stromlos) des Mifare-Chips sind dann diese Einstellungen aktiv.

Die Philosophie, dass Key A geringere Rechte haben kann als Key B ist korrekt. So kann Key A nur das Recht haben einen Speicherblock auszulesen, während Key B diesen Speicherblock auch beschreiben darf.

Da sie immer nur ganze Speicherblöcke à 16 Bytes schreiben, beschreiben sie immer den gesamten 4. Block. Daher ist es wurscht, ob sie Key A verändern oder im Werkswert lassen.

<https://www.mifare.net/support/forum/topic/what-is-mifare-classic-1k-access-bits-means-how-to-calculate-and-use-it/>

Da das Thema beliebig kompliziert und von Hand unüberschaubar ist, können Sie diese dynamische Ansicht verwenden um Möglichkeiten durchzuspielen:

<http://calc.gmss.ru/Mifare1k/>

In dem angehängten Beispiel darf Key A den 1. Block im Sektor auslesen, aber nicht verändern. Das darf nur Key B

Im 2. Block darf nur Key B lesen oder schreiben. Key A darf gar nichts.

Im 3. Block darf Key B lediglich lesen. Key A darf gar nichts.

Im 4. Block darf Key B überschreiben, aber nicht auslesen. Nur die Access-Bytes können von beiden Keys ausgelesen, aber nicht mehr verändert werden.

Das heißt aber auch, dass sie zunächst alle gewünschten Daten in die Blöcke 1 bis 3 eines Sektors einschreiben und erst danach mit dem Beschreiben des 4. Blocks die Passwörter und Zugriffsrechte ein einem Aufwasch setzen.

...

Beispiel

Key A: A1 A2 A3 A4 A5 A6

Access bits: D2 D9 62 00

Key B: B1 B2 B3 B4 B5 B6

Der 4. Block des 2. Sektors wird hexadezimal mit 0x07 adressiert.

Zunächst einmal müssen wir uns an dem Sektor authentifizieren:

50 00 0C 16 60 04 1D B7 60 57 FF FF FF FF FF FF B3

50 = Start

00 0C = 12 Bytes Nutzlast

16 = Kommando Authentifizieren

60 = Verwende Key A

04 = Es wird am Speicherblock authentifiziert, diese Authentifizierung ist aber gültig für Sektor 0x01.

1D B7 60 57 = Es werden nur die 4 LSBytes der UID verwendet

FF FF FF FF FF FF = Key A

B3 = Prüfsumme

Dann können wir, nachdem die vorherigen Blöcke beschrieben sind, den 4. Block mit den Zugriff-Einstellungen schreiben:

50 00 11 18 07 A1 A2 A3 A4 A5 A6 D2 D9 62 00 B1 B2 B3 B4 B5 B6 37

50 Start

00 11 = 17 Bytes Nutzlast

18 = Kommando Schreiben

07 = Block-Adresse zum Schreiben

A1 A2 A3 A4 A5 A6 = Key A (6 LSB Bytes)

D2 D9 62 00 = Zugriffsbits (4 Bytes)

B1 B2 B3 B4 B5 B6 = Key B (6 MSB Bytes)

37 = Prüfsumme

MIFARE Classic 1K Access Bits Calculator

Byte Number	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Description	KEY A						Access Bits				KEY B (optional)					
							0xD2 0xD9 0x62 USER									

Access conditions for data block 0								Application
Access bits			Access condition for					
	C ₁	C ₂	C ₃	read	write	increment	decrement, transfer, restore	
	0	0	0	key A B ¹	key A B ¹	key A B ¹	key A B ¹	transport configuration
	0	1	0	key A B ¹	never	never	never	read/write block
	1	0	0	key A B ¹	key B ¹	never	never	read/write block
	1	1	0	key A B ¹	key B ¹	key B ¹	key A B1	value block
	0	0	1	key A B ¹	never	never	key A B ¹	value block
	0	1	1	key B ¹	key B ¹	never	never	read/write block
	1	0	1	key B ¹	never	never	never	read/write block
	1	1	1	never	never	never	never	read/write block

¹ if Key B may be read in the corresponding Sector Trailer it cannot serve for authentication (all grey marked lines in last table). As a consequences, if the reader authenticates any block of a sector which uses the grey marked access conditions and using key B, the card will refuse any subsequent memory access after authentication.

Access conditions for data block 1								Application
Access bits			Access condition for					
	C1	C2	C3	read	write	increment	decrement, transfer, restore	
☐	0	0	0	key A B ¹	key A B ¹	key A B ¹	key A B ¹	transport configuration
☐	0	1	0	key A B ¹	never	never	never	read/write block
☐	1	0	0	key A B ¹	key B ¹	never	never	read/write block
☐	1	1	0	key A B ¹	key B ¹	key B ¹	key A B ¹	value block
☐	0	0	1	key A B ¹	never	never	key A B ¹	value block
☒	0	1	1	key B ¹	key B ¹	never	never	read/write block
☐	1	0	1	key B ¹	never	never	never	read/write block
☐	1	1	1	never	never	never	never	read/write block

¹ if Key B may be read in the corresponding Sector Trailer it cannot serve for authentication (all grey marked lines in last table). As a consequences, if the reader authenticates any block of a sector which uses the grey marked access conditions and using key B, the card will refuse any subsequent memory access after authentication.

Access conditions for data block 2								Application
Access bits			Access condition for					
	C1 ₂	C2 ₂	C3 ₂	read	write	increment	decrement, transfer, restore	
	0	0	0	key A B ¹	key A B ¹	key A B ¹	key A B ¹	transport configuration
	0	1	0	key A B ¹	never	never	never	read/write block
	1	0	0	key A B ¹	key B ¹	never	never	read/write block
	1	1	0	key A B ¹	key B ¹	key B ¹	key A B1	value block
	0	0	1	key A B ¹	never	never	key A B ¹	value block
	0	1	1	key B ¹	key B ¹	never	never	read/write block
	1	0	1	key B ¹	never	never	never	read/write block
	1	1	1	never	never	never	never	read/write block

¹ if Key B may be read in the corresponding Sector Trailer it cannot serve for authentication (all grey marked lines in last table). As a consequences, if the reader authenticates any block of a sector which uses the grey marked access conditions and using key B, the card will refuse any subsequent memory access after authentication.

Access conditions for the sector trailer										
Access bits			Access condition for						Remark	
			KEYA		Access bits		KEYB			
C ₁	C ₂	C ₃	read	write	read	write	read	write		
0	0	0	never	key A	key A	never	key A	key A		Key B may be read ^[1]
0	1	0	never	never	key A	never	key A	never		Key B may be read ^[1]
1	0	0	never	key B	key A B	never	never	key B		
1	1	0	never	never	key A B	never	never	never		
0	0	1	never	key A	key A	key A	key A	key A		Key B may be read, transport configuration ^[1]
0	1	1	never	key B	key A B	key B	never	key B		
1	0	1	never	never	key A B	key B	never	never		
1	1	1	never	never	key A B	never	never	never		

^[1] for this access condition key B is readable and may be used for data

HTMLified by [Akafugu Corporation](#).
The information is taken from [MF1S503x](#) from NXP Semiconductors.