Using a MFRC522 reader to read and write MIFARE RFID cards on ARDUINO through the MFRC522 library ! COO"RO O#$

Mario Ca%urso &'$ca%urso(libero$it)

MIFARE is the N*+ ,e'iconductors-owned trade'ar. o/ a series o/ chi%s widely used in contactless s'art cards and %roOi'ity cards$ According to the %roducers1 billions o/ s'art card chi%s and 'any 'illions o/ reader 'odules ha2e been sold$ 345

#he technology is owned by N*+ ,e'iconductors &s%in o// /ro' +hili%s Electronics in 2667) with head8uarters in Eindho2en1 Netherlands1 and 'ain business sites in Ni9'egen1 Netherlands1 and :a'burg1 ;er'any$

#he MIFARE na'e co2ers %ro%rietary technologies based u%on 2arious le2els o/ the I,O<IEC 4===> #y%e A 4>$57 M:? contactless s'art card standard$
#he MIFARE na'e &deri2ed /ro' the ter' MI.ron FARE Collection ,yste') co2ers se2en di//erent .inds o/ contactless cards$

MIFARE Classic e'%loys a %ro%rietary %rotocol co'%liant to %arts &but not all) o/ I,O<IEC 4===>-> #y%e A1 with an N*+ %ro%rietary security %rotocol /or authentication and ci%hering$

#he MIFARE Classic card is /unda'entally 9ust a 'e'ory storage de2ice1 where the 'e'ory is di2ided into seg'ents and bloc.s with si'%le security 'echanis's /or access control$ #hey are A,IC-based and ha2e li'ited co'%utational %ower$ #han.s to their reliability and low cost1 those cards are widely used /or electronic wallet1 access control1 cor%orate ID cards1 trans%ortation or stadiu' tic.eting$

#he MIFARE Classic 4@ o//ers 462= bytes o/ data storage1 s%lit into 47 sectorsA each sector is %rotected by two di//erent .eys1 called A and $ Each .ey can be %rogra''ed to allow o%erations such as reading1 writing1 increasing 2aluebloc.s1 etc$ MIFARE Classic =@ o//ers =6B7 bytes s%lit into /orty sectors1 o/ which >2 are sa'e si?e as in the 4@ with eight 'ore that are 8uadru%le si?e sectors$ MIFARE Classic 'ini o//ers >26 bytes s%lit into /i2e sectors$

For each o/ these card ty%es1 47 bytes %er sector are reser2ed /or the .eys and access conditions and can not nor'ally be used /or user data$ Also1 the 2ery /irst 47 bytes contain the serial nu'ber o/ the card and certain other 'anu/acturer data and are read only$ #hat brings the net storage ca%acity o/ these cards down to C52 bytes /or MIFARE Classic 4.1 >==6 bytes /or MIFARE Classic =.1 and 22= bytes /or Mini$ It uses an N*+ %ro%rietary security %rotocol &Cry%to-4) /or authentication and ci%hering$

#he encry%tion used by the MIFARE Classic card uses a =D bit .ey$ 34D5
A %resentation by :enry. +lEt? and @arsten Nohl34B5 at the Chaos Co''unication Congress in Dece'ber 266C described a %artial re2erse-engineering o/ the algorith' used in the MIFARE Classic chi%$ Abstract and slides3265 are a2ailable online$ A %a%er that describes the %rocess o/ re2erse engineering this chi% was %ublished at the August 266D U,ENI* security con/erence$3245

In March 266D the Digital ,ecurity3225 research grou% o/ the Radboud Uni2ersity Ni9'egen 'ade %ublic that they %er/or'ed a co'%lete re2erse-engineering and were able to clone and 'ani%ulate the contents o/ an OF-Chi%.aart which is a MIFARE Classic card$32>5 For de'onstration they used the +roO'ar. de2ice1 a 425 .:? < 4>$57 M:? research instru'ent$32=5 #he sche'atics and so/tware are released under the /ree ;NU ;eneral +ublic Gicense by Honathan Iesthues in 266C$
#hey de'onstrate it is e2en %ossible to %er/or' card-only attac.s using 9ust an ordinary stoc.-co''ercial NFC reader in co'bination with the libn/c library$

In A%ril 266B new and better card-only attac. on MIFARE Classic has been /ound$ It was /irst announced at the Ru'% session o/ Eurocry%t 266B$3>55 #his attac. was

%resented at ,ECR!+# 266B$3>75 #he /ull descri%tion o/ this latest and /astest attac. to date can also be /ound in the IACR %re%rint archi2e$3>C5 #he new attac. i'%ro2es by a /actor o/ 'ore than 46 all %re2ious card-only attac.s on MIFARE Classic1 has instant running ti'e1 and it does not re8uire a costly %reco'%utation$

#he new attac. allows to reco2er the secret .ey o/ any sector o/ MIFARE Classic card 2ia wireless interaction1 within about >66 8ueries to the card$ It can then be co'bined with the nested authentication attac. in the Ni9'egen Oa.land %a%er to reco2er subse8uent .eys al'ost instantly$ oth attac.s co'bined and with the right hardware e8ui%'ent such as +ro0'ar.>1 one should be able to clone any MIFARE Classic card in not 'ore than 46 seconds$ #his is 'uch /aster than %re2iously thought$

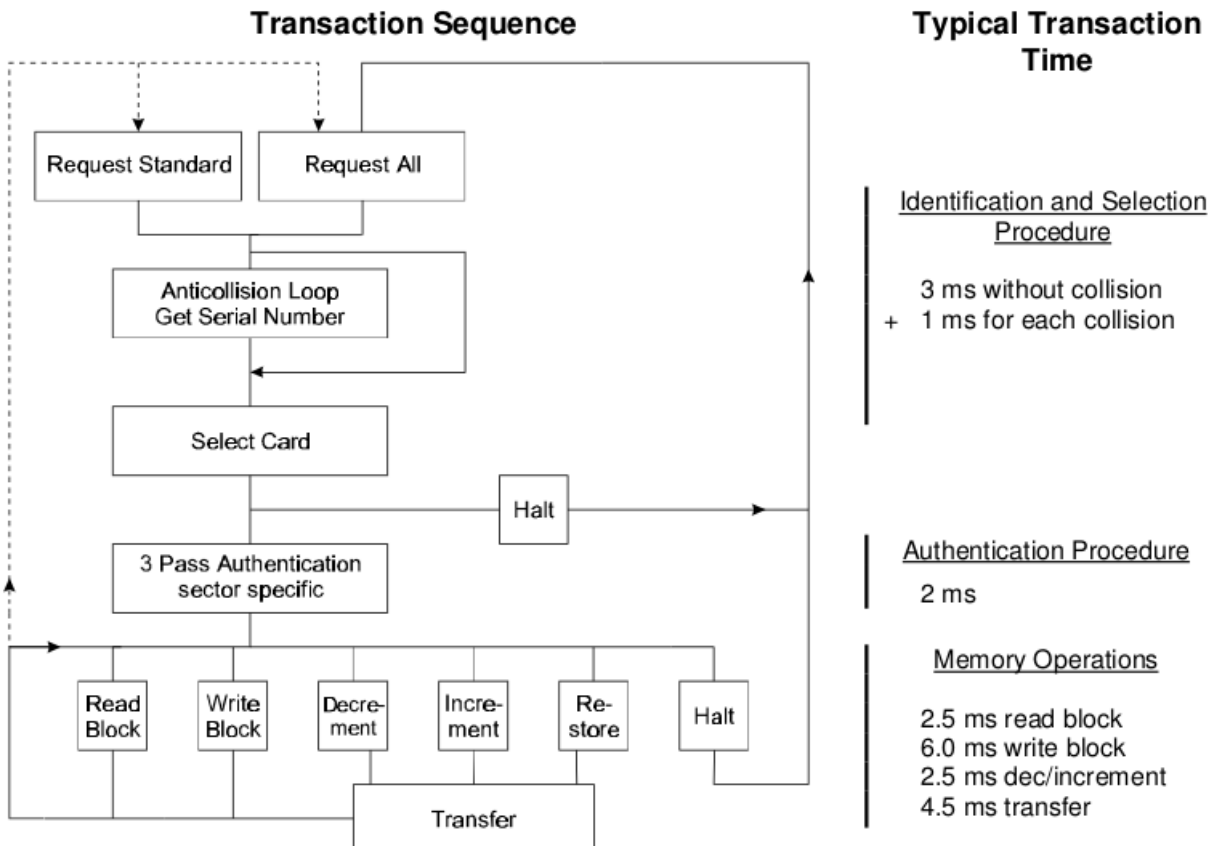
#he MFRC522 is a highly integrated reader<writer IC /or contactless co''unication at 4>$57 M:?$ #he MFRC522 reader su%%orts I,O<IEC 4===> A<MIFARE 'ode$
#he MFRC522Js internal trans'itter is able to dri2e a reader<writer antenna designed to co''unicate with I,O<IEC 4===> A<MIFARE cards and trans%onders without additional acti2e circuitry$ #he recei2er 'odule %ro2ides a robust and e//icient i'%le'entation /or de'odulating and decoding signals /ro' I,O<IEC 4===> A<MIFARE co'%atible cards and trans%onders$ #he digital 'odule 'anages the co'%lete I,O<IEC 4===> A /ra'ing and error detection &%arity and CRC) /unctionality$

#he MFRC522 su%%orts all 2ariants o/ the MIFARE Mini1 MIFARE 4@1 MIFARE =@1 MIFARE Ultralight1 MIFARE DE,Fire EF4 and MIFARE +lus RF identi/ication %rotocols$ #o aid readability throughout this data sheet1 the MIFARE Mini1

MIFARE 4@1 MIFARE =@1 MIFARE Ultralight1 MIFARE DE,Fire EF4 and MIFARE +lus %roducts and %rotocols ha2e the generic na'e MIFARE$

#he /ollowing host inter/aces are %ro2idedK

L ,erial +eri%heral Inter/ace &,+I)
L ,erial UAR# &si'ilar to R,2>2 with 2oltage le2els de%endant on %in 2oltage su%%ly)
L I2C-bus inter/ace



Answer to Re8uest K

Iith the Answer to Re8uest se8uence the MIFAREM RID &Read Irite De2ice) re8uests all MIFAREM cards in the antenna /ield$ Ihen a card is in the o%erating range o/ a RID1 the RID continues co''unication with the a%%ro%riate %rotocol$

Anticollision loo%
In the Anticollision loo% the serial nu'ber o/ the card is read$ I/ there are se2eral cards in the o%erating range o/ a RID they can be distinguished by their di//erent serial nu'bers and one can be selected &,elect card) /or /urther transactions$ #he unselected cards return to the standby 'ode and wait /or a new Answer to Re8uest and Anticollision loo%$

,elect Card
Iith the ,elect Card co''and the RID selects one indi2idual card /or /urther authentication and 'e'ory related o%erations$ #he card returns the Answer to ,elect &A#,) code1 which deter'ines the indi2idual ty%e o/ the selected card$

Access ,%eci/ication
A/ter identi/ication and selection o/ one card the RID s%eci/ies the 'e'ory location o/ the /ollowing access$

#hree +ass Authentication
#he a%%ro%riate access .ey /or the %re2iously s%eci/ied access is used /or > +ass Authentication$ Any co''unication a/ter authentication is auto'atically encry%ted at the sender and decry%ted by the recei2er$

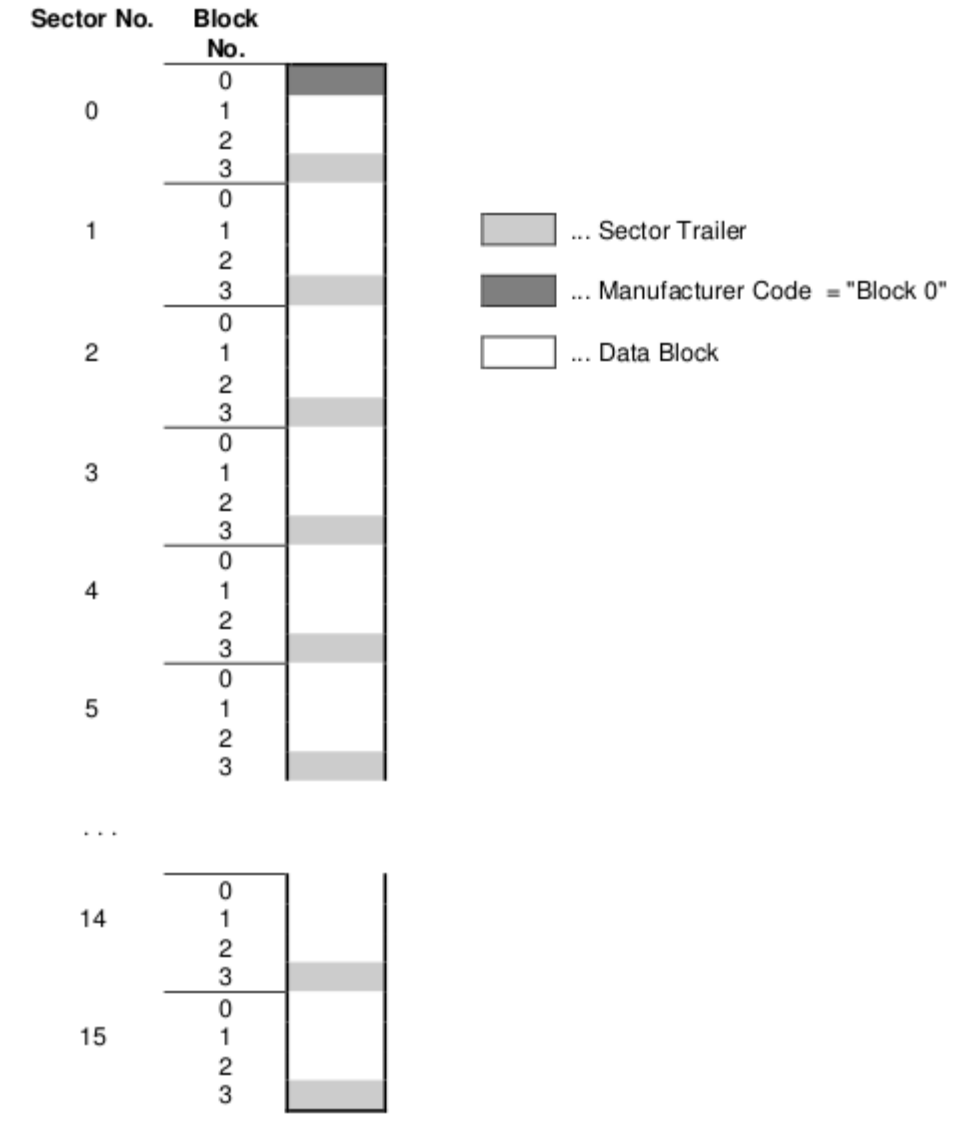
Read<Irite

A/ter authentication any o/ the /ollowing o%erations 'ay be %er/or'edK

READ	reads one bloc.
IRI#E	writes one bloc.
DECREMEN#	decre'ents the contents o/ one bloc. and stores the result in the data-register
INCREMEN#	incre'ents the contents o/ one bloc. and stores the result in the data-register
#RAN,FER	writes the contents o/ the data-register to one bloc.
RE,#ORE	stores the contents o/ one bloc. in the data-register

#he MF4IC,56 IC o/ a Mi/are Classic has integrated a D4B2 it EE+ROM which is s%lit into 47 sectors with = bloc.s$ One bloc. consists o/ 47 bytes &4 yte N D it)$

Memory Organisation:



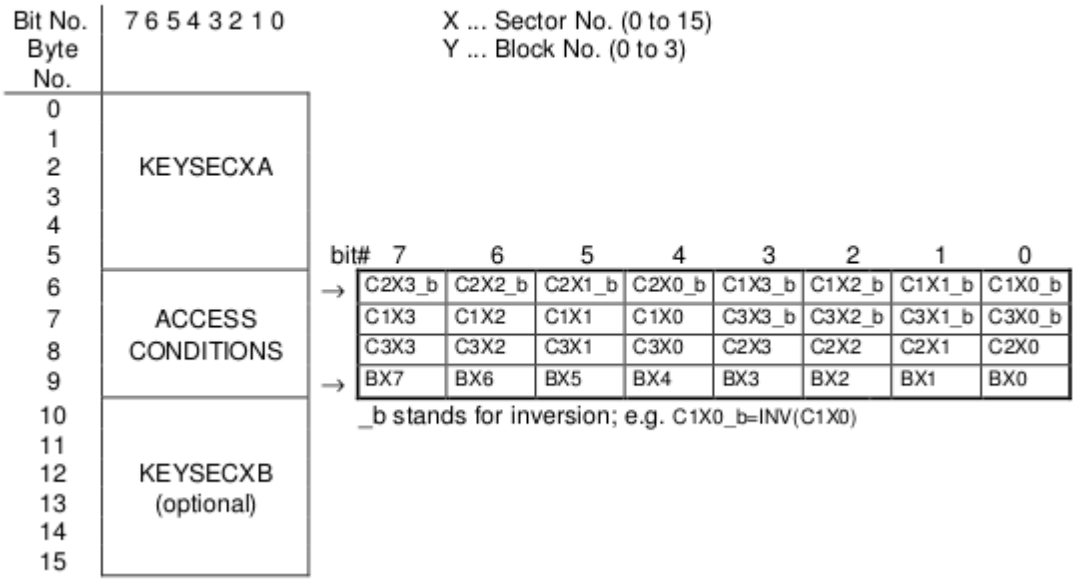
Manu/acturer Code & loc. 6 o/ ,ector 6)

#he /irst bloc. o/ the 'e'ory is reser2ed /or 'anu/acturer data li.e >2 bit serial nu'ber$
#his is a read only bloc.$ In 'any docu'ents it is na'ed 0 loc. 60$

Data loc. & loc. 6 to > e0ce%t 0 loc. 60)

Access conditions /or the Data loc.s are de/ined in the ,ector #railers$ According to these conditions data can be read1 written1 incre'ented1 decre'ented1 trans/erred or restored either with @ey A1 @ey or ne2er$

2.6.1 Sector Trailer (Block 3):



The fourth block of any sector is the Sector Trailer. The Sector Trailer contains access Key A (KEYSECXA) an optional Key B (KEYSECXB) and the access conditions for the four blocks of that sector. If Key B is not needed, the last 6 Bytes of block 3 can be used as data bytes. The corresponding access condition settings are marked grey below.
C1XY to C3XY which are stored twice for safety reasons define the access condition independently for the sector's four blocks. The last byte of the access conditions may be used to store some specific application data (e.g. location of the write backup block).

- Access condition for the Sector Trailer (Y = 3)

			KEYSECXA		ACCESS COND.		KEYSECXB	
C1X3	C2X3	C3X3	read	write	read	write	read	write
0	0	0	never	key A	key A	never	key A	key A
0	1	0	never	never	key A	never	key A	never
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
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

incr, decr, transfer, restore : never

NOTE: Key A|B means key A or key B;
If key B may be read (all grey marked lines) the memory space for Key B is used for data storage and it shall not be used for authentication because all further memory access operations will fail.

Since the transport access conditions (after chip manufacturing) equal to 001, new cards must not be authenticated with Key B !

- Access condition for Data Blocks (Y = 0 to 2)

C1XY	C2XY	C3XY	read	write	incr	decr, transfer, restore
0	0	0	keyA|B[1]	key A|B[1]	key A|B[1]	key A|B[1]
0	1	0	keyA|B[1]	never	never	never
1	0	0	keyA|B[1]	key B[1]	never	never
1	1	0	keyA|B[1]	key B[1]	key B[1]	key A|B[1]
0	0	1	keyA|B[1]	never	never	key A|B[1]
0	1	1	key B[1]	key B[1]	never	never
1	0	1	key B[1]	never	never	never
1	1	1	never	never	never	never

The process of decrement and increment of a block's data is performed and controlled by the Card-IC.

- Transport code

For transportation, KEYSECXA and the access conditions are predefined by the manufacturer as follows:

C1X0, C2X0, C3X0 = 0 0 0	block 0 (data block)
C1X1, C2X1, C3X1 = 0 0 0	block 1 (data block)
C1X2, C2X2, C3X2 = 0 0 0	block 2 (data block)
C1X3, C2X3, C3X3 = 0 0 1	block 3 (Sector Trailer)

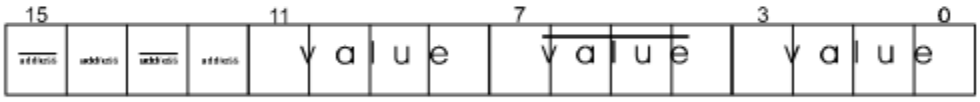
KEYSECXA .secret key, known only by
the manufacturer and system integrator

I/ @ey 'ay be read in the corres%onding ,ector #railer it cannot ser2e /or authentication &all grey 'ar.ed lines in %re2ious table)$ Conse8uencesK I/ the RID tries to authenticate any bloc. o/ a sector with .ey using grey 'ar.ed
access conditions1 the card will re/use any subse8uent 'e'ory access a/ter authentication

In the MF1ICS50 IC two types of Data Blocks are used:

a) read/write blocks
are used to read and write general 16 bytes of data.

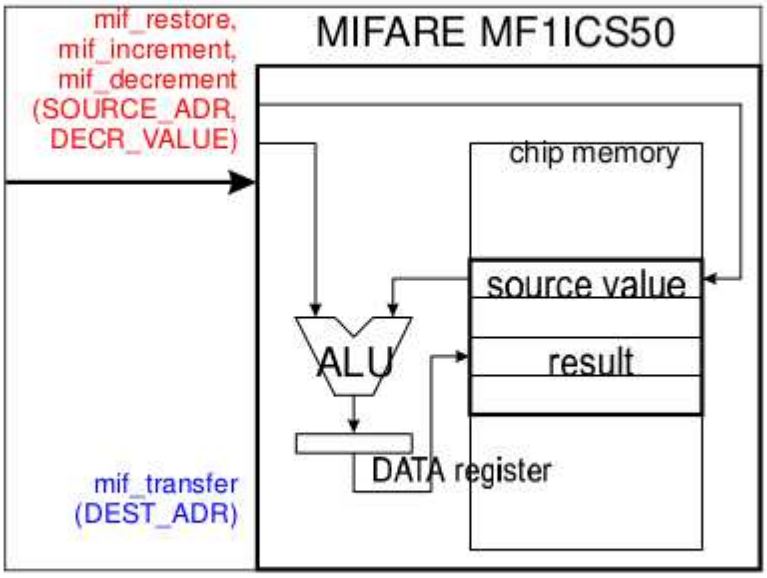
b) value blocks
are used for electronic purse functions (read, increment, decrement, transfer, restore). The maximum size of a value is 4 byte including sign bit, even when a complete 16 byte block has to be reserved. To provide error detection and correction capability, any value is stored 3 times into one value block. The remaining 4 bytes are reserved to some extent for check bits.



value: 32 bit signed 2th complement format stored 3 times
(the consistency of the 3 occurrences of the value is internally checked before the chip can perform any calculation)

address:8 bit arbitrary address byte stored 4 times
(this byte is not internally interpreted)

A value blocks is first time generated by a WRITE instruction to the desired address. The value may be used for subsequent DECREMENT / INCREMENT / RESTORE instructions.



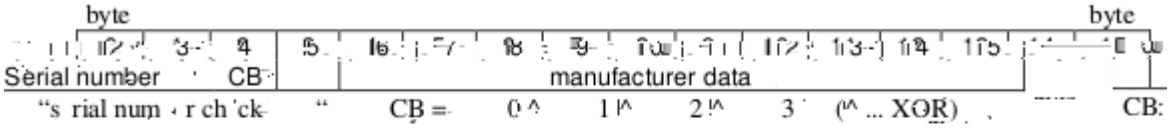
The result of a calculation instruction is temporally stored in a buffer register. For updating the memory with the calculation result the TRANSFER instruction has to be issued. The chip refuses calculations if any error in the block format could be detected.

#he described 'e'ory organi?ation 'a.es it %ossible to a%%oint di//erent sectors to di//erent a%%lications and to %re2ent data corru%tion by using a%%lication s%eci/ic secret .eys$ @eys can only be altered by a RID which has stored the actual @ey A or @ey i/ this is allowed according to access conditions$ Otherwise the actual .ey cannot be changed any'ore$

e/ore the eOecution o/ a co''and the correct /or'at o/ the Access Conditions is chec.ed by the Card-IC$ #hus1 when %rogra''ing the ,ector #railer the card needs to be /iOed within the o%erating range o/ a RIDPs antenna to %re2ent interru%tion o/ the write o%eration because any unsuccess/ul write o%eration 'ay lead to bloc.ing the whole sector$

2.7 Memory contents after IC test

2.7.1 Block 0 (manufacturer block):



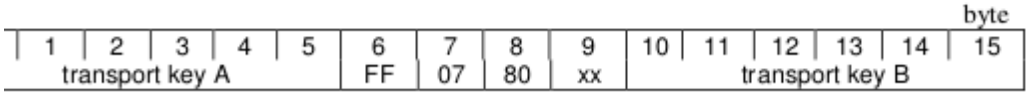
.2 Data Blocks:

a blocks: contain variable data.
cks 1,2 / 4,5,6 / 8,9,10 / 12,13,14 / 16,17,18 / 20,21,22 / 24,25,26 / 28,29,30 / 32,33,34 /
7,38 / 40,41,42 / 44,45,46 / 48,49,50 / 52,53,54 / 56,57,58 / 60,61,62)

.3 Sector Trailers:

e: The initial state of sector trailers after IC test can be modified depending on the personalisation done e.g. at the card manufacturer.

ult coding:



cks 3 / 7 / 11 / 15 / 19 / 23 / 27 / 31 / 35 / 39 / 43 / 47 / 51 / 55 / 59 / 63)
e 9 of all sector trailers is not defined. Its memory contents after IC test can vary.

MFRC522$h Q A Gibrary to use ARDUINO RFID MODUGE @I# 4>$57 M:R ! COO"RO O#$

#here are three hardware co'%onents in2ol2edK

4) #he 'icro controllerK An Arduino
2) #he +CD &+roOi'ity Cou%ling De2ice)K N*+ MFRC522 Contactless Reader IC
>) #he +ICC &short /or +roOi'ity Integrated Circuit Card)K A card or tag using the I,O 4===>A inter/ace1 eg Mi/are or N#A;26>$

MIFARE Classic 4@ &MF4,56>O)K

:as 47 sectors S = bloc.s<sector S 47 bytes<bloc. N 462= bytes$ #he bloc.s are nu'bered 6-7>$

loc. > in each sector is the ,ector #railer$
S ytes 6-5K @ey A
S ytes 7-DK Access its
S ytes BK User data
S ytes 46-45K @ey &or user data)

loc. 6 is read only 'anu/acturer data$

#o access a bloc.1 an authentication using a .ey /ro' the bloc.Ps sector 'ust be %er/or'ed /irst$

EOa'%leK #o read /ro' bloc. 461 /irst authenticate using a .ey /ro' sector > &bloc.s D-44)$

All .eys are set to FFFFFFFFFFFFh at chi% deli2ery$

IarningK +lease read section D$C OMe'ory AccessO$ It includes this teOtK i/ the +ICC detects a /or'at 2iolation the whole sector is irre2ersibly bloc.ed$

#o use a bloc. in O2alue bloc.O 'ode &/or Incre'ent<Decre'ent o%erations) you need to change the sector trailer$ Use +ICCT,etAccess its&) to calculate the bit %atterns$

Co''ands sent to the +ICC$

#he co''ands used by the +CD to 'anage co''unication with se2eral +ICCs &I,O 4===>->1 #y%e A1 section 7$=)

+ICCTCMDTRE"A N 6O271
RE"uest co''and1 #y%e A$ In2ites +ICCs in state IDGE to go to READ! and %re%are /or anticollision or selection$ C bit /ra'e$
+ICCTCMDTIU+A N 6O521
Ia.e-U+ co''and1 #y%e A$ In2ites +ICCs in state IDGE and :AG# to go to READ!&S) and %re%are /or anticollision or selection$ C bit /ra'e$
+ICCTCMDTC# N 6ODD1
Cascade #ag$ Not really a co''and1 but used during anti collision$
+ICCTCMDT,EGTCG4 N 6OB>1
Anti collision<,elect1 Cascade Ge2el 4
+ICCTCMDT,EGTCG2 N 6OB51
Anti collision<,elect1 Cascade Ge2el 2
+ICCTCMDT,EGTCG> N 6OBC1
Anti collision<,elect1 Cascade Ge2el >
+ICCTCMDT:G#A N 6O561
:aG# co''and1 #y%e A$ Instructs an AC#IFE +ICC to go to state :AG#$

#he co''ands used /or MIFARE Classic
Use +CDTMFAuthent to authenticate access to a sector1 then use these co''ands to read<write<'odi/y the bloc.s on the sector$
#he read<write co''ands can also be used /or MIFARE Ultralight$

+ICCTCMDTMFTAU#:T@E!TA   N 60761
+er/or' authentication with @ey A
+ICCTCMDTMFTAU#:T@E!T   N 60741
+er/or' authentication with @ey
+ICCTCMDTMFTREAD   N 60>61
Reads one 47 byte bloc. /ro' the authenticated sector o/ the +ICC$ Also used /or MIFARE Ultralight$
+ICCTCMDTMFTIRI#E   N 60A61
Irites one 47 byte bloc. to the authenticated sector o/ the +ICC$ Called OCOM+A#I IGI#! IRI#EO /or MIFARE Ultralight$
+ICCTCMDTMFTDECREMEN#   N 60C61
Decre'ents the contents o/ a bloc. and stores the result in the internal data register$
+ICCTCMDTMFTINCREMEN#   N 60C41
Incre'ents the contents o/ a bloc. and stores the result in the internal data register$
+ICCTCMDTMFTRE,#ORE   N 60C21
Reads the contents o/ a bloc. into the internal data register$
+ICCTCMDTMFT#RAN,FER   N 60 61
Irites the contents o/ the internal data register to a bloc.$

Gist o/ the /unctions in the library

Functions /or setting u% the Arduino

MFRC522&byte chi%,elect+in1 byte reset+owerDown+in)A
2oid set,+ICon/ig&)A

 asic inter/ace /unctions /or co''unicating with the MFRC522

2oid +CDTIriteRegister&byte reg1 byte 2alue)A
2oid +CDTIriteRegister&byte reg1 byte count1 byte S2alues)A
byte +CDTReadRegister&byte reg)A
2oid +CDTReadRegister&byte reg1 byte count1 byte S2alues1 byte rOAlign N 6)A
2oid set itMas.&unsigned char reg1 unsigned char 'as.)A
2oid +CDT,etRegister itMas.&byte reg1 byte 'as.)A
2oid +CDTClearRegister itMas.&byte reg1 byte 'as.)A
byte +CDTCalculateCRC&byte Sdata1 byte length1 byte Sresult)A

Functions /or 'ani%ulating the MFRC522

2oid +CDTInit&)A
2oid +CDTReset&)A
2oid +CDTAntennaOn&)A

Functions /or co''unicating with +ICCs

byte +CDT#ranscei2eData&byte SsendData1 byte sendGen1 byte Sbac.Data1 byte Sbac.Gen1 byte S2alid its N NUGG1 byte rOAlign N 61 bool chec.CRC N /alse)A

byte +CDTCo''unicatelith+ICC&byte co''and1 byte waitIR81 byte SsendData1 byte sendGen1 byte Sbac.Data N NUGG1 byte Sbac.Gen N NUGG1 byte S2alid its N NUGG1 byte rOAlign N 61 bool chec.CRC N /alse)A

```
byte +ICCTRe8uestA&byte Sbu//erA#"A1 byte Sbu//er,i?e)A
byte +ICCTIa.eu%A&byte Sbu//erA#"A1 byte Sbu//er,i?e)A
byte +ICCTRE"ATorTIU+A&   byte co''and1 byte Sbu//erA#"A1 byte Sbu//er,i?e)A
byte +ICCT,elect&Uid Suid1 byte 2alid its N 6)A
byte +ICCT:altA&)A

Functions /or co''unicating with MIFARE +ICCs

byte +CDTAuthenticate&byte co''and1 byte bloc.Addr1 MIFARET@ey S.ey1 Uid Suid)A
2oid +CDT,to%Cry%to4&)A
byte MIFARETRead&byte bloc.Addr1 byte Sbu//er1 byte Sbu//er,i?e)A
byte MIFARETIrite&byte bloc.Addr1 byte Sbu//er1 byte bu//er,i?e)A
byte MIFARETDecre'ent&byte bloc.Addr1 long delta)A
byte MIFARETIncre'ent&byte bloc.Addr1 long delta)A
byte MIFARETRestore&byte bloc.Addr)A
byte MIFARET#rans/er&byte bloc.Addr)A
byte MIFARETUltralightTIrite&byte %age1 byte Sbu//er1 byte bu//er,i?e)A

,u%%ort /unctions

byte +CDTMIFARET#ranscei2e&     byte SsendData1 byte sendGen1 bool acce%t#i'eout N /alse)A
const char S;et,tatusCodeNa'e&byte code)A
byte +ICCT;et#y%e&byte sa.)A
const char S+ICCT;et#y%eNa'e&byte ty%e)A
2oid +ICCTDu'%#o,erial&Uid Suid)A
2oid +ICCTDu'%Mi/areClassic#o,erial&Uid Suid1 byte %icc#y%e1 MIFARET@ey S.ey)A
2oid +ICCTDu'%Mi/areClassic,ector#o,erial&Uid Suid1 MIFARET@ey S.ey1 byte sector)A
2oid +ICCTDu'%Mi/areUltralight#o,erial&)A
2oid MIFARET,etAccess its&byte Saccess it u//er1 byte g61 byte g41 byte g21 byte g>)A


Con2enience /unctions - does not add eOtra /unctionality

bool +ICCTIsNewCard+resent&)A
bool +ICCTReadCard,erial&)A


Detailed docu'entation Q enu' and structures

+ICC ty%es we can detect$ Re'e'ber to u%date +ICCT;et#y%eNa'e&) i/ you add 'ore$

enu' +ICCT#y%e

Return codes /ro' the /unctions in this class$ Re'e'ber to u%date ;et,tatusCodeNa'e&) i/ you
add 'ore$

enu' ,tatusCode

A struct used /or %assing the UID o/ a +ICC$

ty%ede/ struct U
      byte         si?eA        Nu'ber o/ bytes in the UID$ =1 C or 46$
      byte         uid yte3465A
      byte         sa.A         #he ,A@ &,elect ac.nowledge) byte returned
                                /ro' the +ICC a/ter success/ul selection$
      V Uid

A struct used /or %assing a MIFARE Cry%to4 .ey

ty%ede/ struct U
      byte         .ey yte3MFT@E!T,IRE5A
      V MIFARET@eyA
```

```
EOa'%leK
<< +re%are .ey - all .eys are set to FFFFFFFFFFFFh at chi% deli2ery /ro' the /actory$
MFRC522KKMIFARET@ey .eyA
/or &byte i N 6A i W 7A iXX) .ey$.ey yte3i5 N 6OFFA

Me'ber 2ariables

      Uid uidA           Used by +ICCTReadCard,erial&)$

Detailed docu'entation Q /unctions

XCreate ob9ect instance
MFRC522&
byte chi%,elect+in1       Arduino %in used /or ,+I chi% select
byte reset+owerDown+in    Arduino %in used /or ,+I reset
)A
EOa'%leK
Yinclude W,+I$hZ
Yinclude WMFRC522$hZ
Yde/ine ,,T+IN 46    <<Arduino Uno
Yde/ine R,#T+IN B
MFRC522 '/rc522&,,T+IN1 R,#T+IN)A        << Create MFRC522 instance$

XInitiali?es the MFRC522 chi%$
2oid MFRC522KK+CDTInit&)
EOa'%leK
2oid setu%&) U
        ,erial$begin&B766)A        << Init serial co''unications with +C
        ,+I$begin&)A               << Init ,+I bus
        '/rc522$+CDTInit&)A        << Init MFRC522 card
V

X,et ,+I bus to wor. with MFRC522 chi%$

+lease call this /unction i/ you ha2e changed the ,+I con/ig since the MFRC522 constructor
was run$

2oid MFRC522KKset,+ICon/ig&)

X+er/or's a so/t reset on the MFRC522 chi% and waits /or it to be ready again$
2oid MFRC522KK+CDTReset&)

X#urns the antenna on by enabling %ins #*4 and #*2$
A/ter a reset these %ins are disabled$
2oid MFRC522KK+CDTAntennaOn&)

X#rans'its a RE"uest co''and1 #y%e A$ In2ites +ICCs in state IDGE to go to READ! and %re%are
/or anticollision or selection$ C bit /ra'e$
 ewareK Ihen two +ICCs are in the /ield at the sa'e ti'e I o/ten get ,#A#U,T#IMEOU# -
%robably due do bad antenna design$
return ,#A#U,TO@ on success1 ,#A#U,T[[[ otherwise$
byte MFRC522KK+ICCTRe8uestA&
byte Sbu//erA#"A1  #he bu//er to store the A#"A &Answer to re8uest) in
byte Sbu//er,i?e    u//er si?e1 at least two bytes$ Also nu'ber o/ bytes returned i/
,#A#U,TO@$


X#rans'its a la.e-U+ co''and1 #y%e A$ In2ites +ICCs in state IDGE and :AG# to go to READ!&S)
and %re%are /or anticollision or selection$ C bit /ra'e$
 ewareK Ihen two +ICCs are in the /ield at the sa'e ti'e I o/ten get ,#A#U,T#IMEOU# -
%robably due do bad antenna design$
```

return ,#A#U,TO@ on success1 ,#A#U,T[[[otherwise$
byte MFRC522KK+ICCTIa.eu%A&
byte Sbu//erA#"A1 #he bu//er to store the A#"A &Answer to re8uest) in
byte Sbu//er,i?e   u//er si?e1 at least two bytes$ Also nu'ber o/ bytes returned i/
,#A#U,TO@$

X#rans'its ,EGEC#<AN#ICOGGI,ION co''ands to select a single +ICC$
 e/ore calling this /unction the +ICCs 'ust be %laced in the READ!&S) state by calling
+ICCTRe8uestA&) or +ICCTIa.eu%A&)$
On successK

-#he chosen +ICC is in state AC#IFE&S) and all other +ICCs ha2e returned to state IDGE<:AG#$
&Figure C o/ the I,O<IEC 4===>-> dra/t$)
-#he UID si?e and 2alue o/ the chosen +ICC is returned in Suid along with the ,A@$

A +ICC UID consists o/ =1 C or 46 bytes$
Only = bytes can be s%eci/ied in a ,EGEC# co''and1 so /or the longer UIDs two or three
iterations are usedK

UID si?e	Nu'ber o/ UID bytes	Cascade le2els	E0a'%le o/ +ICC
NNNNNNNN	NNNNNNNNNNNNNNNNNNNN	NNNNNNNNNNNNNN	NNNNNNNNNNNNNNN
single	=	4	MIFARE Classic
double	C	2	MIFARE Ultralight
tri%le	46	>	Not currently in use[

return ,#A#U,TO@ on success1 ,#A#U,T[[[otherwise$
byte MFRC522KK+ICCT,elect&
Uid Suid1 +ointer to Uid struct$ Nor'ally out%ut1 but can also
be used to su%%ly a .nown UID$

byte 2alid its #he nu'ber o/ .nown UID bits su%%lied in Suid$

Nor'ally 6$ I/ set you 'ust also su%%ly uid-Zsi?e$

Descri%tion o/ bu//er structureK

 yte 6K ,EG Indicates the Cascade Ge2elK +ICCTCMDT,EGTCG41 +ICCTCMDT,EGTCG2 or
+ICCTCMDT,EGTCG>
 yte 4K NF Nu'ber o/ Falid its &in co'%lete co''and1 not 9ust the UID)K :igh
nibbleK co'%lete bytes1 Gow nibbleK E0tra bits$
 yte 2K UID-data or C# ,ee e0%lanation below$ C# 'eans Cascade #ag$
 yte >K UID-data
 yte =K UID-data
 yte 5K UID-data
 yte 7K CC loc. Chec. Character - *OR o/ bytes 2-5
 yte CK CRCTA
 yte DK CRCTA

#he CC and CRCTA is only trans'itted i/ we .now all the UID bits o/ the current Cascade
Ge2el$
Descri%tion o/ bytes 2-5K &,ection 7$5$= o/ the I,O<IEC 4===>-> dra/tK UID contents and
cascade le2els)

UID si?e	Cascade le2el	yte2	yte>	yte=	yte5
NNNNNNNN	NNNNNNNNNNNNNN	NNNNN	NNNNN	NNNNN	NNNNN
= bytes	4	uid6	uid4	uid2	uid>
C bytes	4	C#	uid6	uid4	uid2
	2	uid>	uid=	uid5	uid7
46 bytes	4	C#	uid6	uid4	uid2
	2	C#	uid>	uid=	uid5

```
XInstructs a +ICC in state AC#IFE&S) to go to state :AG#$
return ,#A#U,TO@ on success1 ,#A#U,T[[[ otherwise$
byte +ICCT:altA&)A
EOa'%leK
<< :alt +ICC
'/rc522$+ICCT:altA&)A


XEOecutes the MFRC522 MFAuthent co''and$
#his co''and 'anages MIFARE authentication to enable a secure co''unication to any MIFARE
Mini1 MIFARE 4@ and MIFARE =@ card$
#he authentication is described in the MFRC522 datasheet section 46$>$4$B and
htt%K<<www$nO%$co'<docu'ents<dataTsheet<MF4,56>0$%d/ section 46$4$ /or use with MIFARE
Classic +ICCs$
#he +ICC 'ust be selected - ie in state AC#IFE&S) - be/ore calling this /unction$
Re'e'ber to call +CDT,to%Cry%to4&) at the end o/ co''unication  with the authenticated +ICC
- otherwise no new co''unications can start$
All .eys are set to FFFFFFFFFFFFh at chi% deli2ery$
return ,#A#U,TO@ on success1 ,#A#U,T[[[ otherwise$ +robably ,#A#U,T#IMEOU# i/ you su%%ly the
wrong .ey$
byte MFRC522KK+CDTAuthenticatel&
byte co''and1             +ICCTCMDTMFTAU#:T@E!TA or +ICCTCMDTMFTAU#:T@E!T
byte bloc.Addr1           #he bloc. nu'ber$ ,ee nu'bering in the co''ents in
                          the $h /ile$
MIFARET@ey S.ey1          +ointer to the Cry%to4 .ey to use &7 bytes)
Uid Suid                  +ointer to Uid struct$ #he /irst = bytes o/ the UID
                          is used$
)
EOa'%leK
MFRC522KKMIFARET@ey .eyA
/or &byte i N 6A i W 7A iXX) .ey$.ey yte3i5 N 60FFA
byte trailer loc.   N CA
byte statusA
i/ & \ '/rc522$+ICCTReadCard,erial&)) returnA
status N '/rc522$+CDTAuthenticate&MFRC522KK+ICCTCMDTMFTAU#:T@E!TA1 trailer loc.1 ].ey1
]&'/rc522$uid))A
i/ &status \N MFRC522KK,#A#U,TO@) U
   ,erial$%rint&O+CDTAuthenticate&) /ailedK O)A
   ,erial$%rintln&'/rc522$;et,tatusCodeNa'e&status))A
   returnA
V

XUsed to eOit the +CD /ro' its authenticated state$
Re'e'ber to call +CDT,to%Cry%to4&) at the end o/ co''unication  with the authenticated +ICC
- otherwise no new co''unications can start$
2oid MFRC522KK+CDT,to%Cry%to4&)
EOa'%leK
<< ,to% encry%tion on +CD
'/rc522$+CDT,to%Cry%to4&)A

XReads 47 bytes &X 2 bytes CRCTA) /ro' the acti2e +ICC$
For MIFARE Classic the sector containing the bloc. 'ust be authenticated be/ore calling this
/unction$
For MIFARE Ultralight only addresses 66h to 6Fh are decoded$
 S #he MF6ICU4 returns a NA@ /or higher addresses$
 S #he MF6ICU4 res%onds to the READ co''and by sending 47 bytes starting /ro'
       the %age address de/ined by the co''and argu'ent$
 S For eOa'%leA i/ bloc.Addr is 6>h then %ages 6>h1 6=h1 65h1 67h are returned$
 S A roll-bac. is i'%le'entedK I/ bloc.Addr is 6Eh1 then the contents o/ %ages 6Eh1 6Fh1 66h
```

```
and 64h are returned$
#he bu//er 'ust be at least 4D bytes because a CRCTA is also returned$
Chec.s the CRCTA be/ore returning ,#A#U,TO@$
return ,#A#U,TO@ on success1 ,#A#U,T[[[ otherwise$
byte MFRC522KKMIFARETRead&
byte bloc.Addr1     MIFARE ClassicK #he bloc. &6-60//) nu'ber$
                    MIFARE UltralightK #he /irst %age to return data /ro'$
byte Sbu//er1       #he bu//er to store the data in

byte Sbu//er,i?e     u//er si?e1 at least 4D bytes$ Also nu'ber o/ bytes returned
                    i/ ,#A#U,TO@$
E0a'%leK
byte 2alue loc.A    N =A byte bu//er34D5A byte si?e N si?eo/&bu//er)A
byte status N '/rc522$MIFARETRead&2alue loc.A1 bu//er1 ]si?e)A

XIrites 47 bytes to the acti2e +ICC$
For MIFARE Classic the sector containing the bloc. 'ust be authenticated be/ore calling this
/unction$
For MIFARE Ultralight the o%eration is called OCOM+A#I IGI#! IRI#EO$
E2en though 47 bytes are trans/erred to the Ultralight +ICC1 only the least signi/icant =
bytes &bytes 6 to >) are written to the s%eci/ied address$ It is reco''ended to set the
re'aining bytes 6=h to 6Fh to all logic 6$
return ,#A#U,TO@ on success1 ,#A#U,T[[[ otherwise$
byte MFRC522KKMIFARETIrite&
byte bloc.Addr1
                    MIFARE ClassicK #he bloc. &6-60//) nu'ber$
                    MIFARE UltralightK #he %age &2-45) to write to$
byte Sbu//er1       #he 47 bytes to write to the +ICC
byte bu//er,i?e      u//er si?e1 'ust be at least 47 bytes$ E0actly 47 bytes are
                     written$
E0a'%leK
byte 2alue loc.A    N =A
byte 2alue4 loc.35 N U 4121>1=1  5171C1D1 B14612551421  4>14=145147VA
status N '/rc522$MIFARETIrite&2alue loc.A1 2alue4 loc.1 47)A
i/ &status \N MFRC522KK,#A#U,TO@) U
   ,erial$%rint&OMIFARETIrite&) /ailedK O)A
   ,erial$%rintln&'/rc522$;et,tatusCodeNa'e&status))A
V


XMIFARE Decre'ent subtracts the delta /ro' the 2alue o/ the addressed bloc.1 and stores the
result in a 2olatile 'e'ory$
For MIFARE Classic only$ #he sector containing the bloc. 'ust be authenticated be/ore
calling this /unction$
Only /or bloc.s in O2alue bloc.O 'ode1 ie with access bits 3C4 C2 C>5 N 34465 or 36645$
Use MIFARET#rans/er&) to store the result in a bloc.$
return ,#A#U,TO@ on success1 ,#A#U,T[[[ otherwise$
byte MFRC522KKMIFARETDecre'ent&
byte bloc.Addr1     #he bloc. &6-60//) nu'ber$
long delta          #his nu'ber is subtracted /ro' the 2alue o/ bloc. bloc.Addr$


XMIFARE Incre'ent adds the delta to the 2alue o/ the addressed bloc.1 and stores the result
in a 2olatile 'e'ory$
For MIFARE Classic only$ #he sector containing the bloc. 'ust be authenticated be/ore
calling this /unction$
Only /or bloc.s in O2alue bloc.O 'ode1 ie with access bits 3C4 C2 C>5 N 34465 or 36645$
Use MIFARET#rans/er&) to store the result in a bloc.$
return ,#A#U,TO@ on success1 ,#A#U,T[[[ otherwise$
byte MFRC522KKMIFARETIncre'ent&
byte bloc.Addr1     #he bloc. &6-60//) nu'ber$
long delta          #his nu'ber is added to the 2alue o/ bloc. bloc.Addr$
```

```
E0a'%leK
<< Add 4 to the 2alue o/ 2alue loc.A and store the result in 2alue loc.A$
byte 2alue loc.A    N 5A
,erial$%rint&OAdding 4 to 2alue o/ bloc. 0)A ,erial$%rintln&2alue loc.A)A
byte status N '/rc522$MIFARETIncre'ent&2alue loc.A1 4)A
i/ &status \N MFRC522KK,#A#U,TO@) U
       ,erial$%rint&OMIFARETIncre'ent&) /ailedK 0)A
       ,erial$%rintln&'/rc522$;et,tatusCodeNa'e&status))A
       returnA
V
status N '/rc522$MIFARET#rans/er&2alue loc.A)A
i/ &status \N MFRC522KK,#A#U,TO@) U
       ,erial$%rint&OMIFARET#rans/er&) /ailedK 0)A
       ,erial$%rintln&'/rc522$;et,tatusCodeNa'e&status))A
       returnA
V


XMIFARE Restore co%ies the 2alue o/ the addressed bloc. into a 2olatile 'e'ory$
For MIFARE Classic only$ #he sector containing the bloc. 'ust be authenticated be/ore
calling this /unction$
Only /or bloc.s in O2alue bloc.O 'ode1 ie with access bits 3C4 C2 C>5 N 34465 or 36645$
Use MIFARET#rans/er&) to store the result in a bloc.$
return ,#A#U,TO@ on success1 ,#A#U,T[[[ otherwise$
byte MFRC522KKMIFARETRestore&
byte bloc.Addr      #he bloc. &6-60//) nu'ber$
#he datasheet describes Restore as a two ste% o%eration1 but does not e0%lain what data to
trans/er in ste% 2$Doing only a single ste% does not wor.1 so I chose to trans/er 6G in ste%
two$

X:el%er /unction /or the two-ste% MIFARE Classic %rotocol o%erations Decre'ent1 Incre'ent
and Restore$
return ,#A#U,TO@ on success1 ,#A#U,T[[[ otherwise$
byte MFRC522KKMIFARET#wo,te%:el%er&
byte co''and1       #he co''and to use
byte bloc.Addr1     #he bloc. &6-60//) nu'ber$
long data           #he data to trans/er in ste% 2
                                                        )

XMIFARE #rans/er writes the 2alue stored in the 2olatile 'e'ory into one MIFARE Classic
bloc.$
For MIFARE Classic only$ #he sector containing the bloc. 'ust be authenticated be/ore
calling this /unction$
Only /or bloc.s in O2alue bloc.O 'ode1 ie with access bits 3C4 C2 C>5 N 34465 or 36645$
return ,#A#U,TO@ on success1 ,#A#U,T[[[ otherwise$
byte MFRC522KKMIFARET#rans/er&
byte bloc.Addr      #he bloc. &6-60//) nu'ber$


XReturns a string %ointer to a status code na'e$
const char SMFRC522KK;et,tatusCodeNa'e&
byte code     One o/ the ,tatusCode enu's$

       code 2alues          return 2alues
         ,#A#U,TO@K               O,uccess$O
         ,#A#U,TERRORK            OError in co''unication$O
         ,#A#U,TCOGGI,IONK OCollision detected$O
         ,#A#U,T#IMEOU#K          O#i'eout in co''unication$O
         ,#A#U,TNOTROOMK          OA bu//er is not big enough$O
         ,#A#U,TIN#ERNAGTERRORK   OInternal error in the code$ ,hould not ha%%en$O
         ,#A#U,TINFAGIDK          OIn2alid argu'ent$O
         ,#A#U,TCRCTIRON;K O#he CRCTA does not 'atch$O
```

```
        ,#A#U,TMIFARETNAC@K      OA MIFARE +ICC res%onded with NA@$O
        de/aultK                 OUn.nown errorO

EOa'%leK
,erial$%rintln&'/rc522$;et,tatusCodeNa'e&status))A

X#ranslates the ,A@ &,elect Ac.nowledge) to a +ICC ty%e$
return +ICCT#y%e
byte MFRC522KK+ICCT;et#y%e&
byte sa.            #he ,A@ byte returned /ro' +ICCT,elect&)$


sa.                 return 2alue
sa. ] 606=          +ICCT#!+ETNO#TCOM+GE#E
606B                +ICCT#!+ETMIFARETMINI
606D                +ICCT#!+ETMIFARET4@
604D                +ICCT#!+ETMIFARET=@
6066                +ICCT#!+ETMIFARETUG
6046 or 6044 +ICCT#!+ETMIFARET+GU,
6064                +ICCT#!+ET#N+>***
sa. ] 6026          +ICCT#!+ETI,OT4===>T=
sa. ] 60=6          +ICCT#!+ETI,OT4D6B2
else                +ICCT#!+ETUN@NOIN

XReturns a string %ointer to the +ICC ty%e na'e$
const char SMFRC522KK+ICCT;et#y%eNa'e&
byte %icc#y%e       One o/ the +ICCT#y%e enu's$


%icc#y%e                    return 2alue
+ICCT#!+ETI,OT4===>T=       O+ICC co'%liant with I,O<IEC 4===>-=O
+ICCT#!+ETI,OT4D6B2K        O+ICC co'%liant with I,O<IEC 4D6B2 &NFC)O
+ICCT#!+ETMIFARETMINI       OMIFARE Mini1 >26 bytesO
+ICCT#!+ETMIFARET4@OMIFARE 4@ O
+ICCT#!+ETMIFARET=@OMIFARE =@ O
+ICCT#!+ETMIFARETUGOMIFARE Ultralight or Ultralight CO
+ICCT#!+ETMIFARET+GU,       OMIFARE +lusO

+ICCT#!+ET#N+>***           OMIFARE #N+>***O
+ICCT#!+ETNO#TCOM+GE#E      O,A@ indicates UID is not co'%lete$O
+ICCT#!+ETUN@NOIN           OUn.nown ty%e^
EOa'%leK
,erial$%rintln&'/rc522$+ICCT;et#y%eNa'e&%icc#y%e))A

XDu'%s debug in/o about the selected +ICC to ,erial$
On success the +ICC is halted a/ter du'%ing the data$
For MIFARE Classic the /actory de/ault .ey o/ 60FFFFFFFFFFFF is tried$
2oid MFRC522KK+ICCTDu'%#o,erial&
Uid Suid     +ointer to Uid struct returned /ro' a success/ul +ICCT,elect&)$
EOa'%leK
'/rc522$+ICCTDu'%#o,erial&]&'/rc522$uid))A

XDu'%s 'e'ory contents o/ a sector o/ a MIFARE Classic +ICC$
Uses +CDTAuthenticate&)1 MIFARETRead&) and +CDT,to%Cry%to4$
Always uses +ICCTCMDTMFTAU#:T@E!TA because only @ey A can always read the sector trailer
access bits$
2oid MFRC522KK+ICCTDu'%Mi/areClassic,ector#o,erial&
Uid Suid1           +ointer to Uid struct returned /ro' a success/ul +ICCT,elect&)
MIFARET@ey S.ey1    @ey A /or the sector$
byte sector         #he sector to du'%1 6$$>B$

XCalculates the bit %attern needed /or the s%eci/ied access bits$ In the 3C4 C2 C>5 tu%les
```

```
C4 is M,  &N=) and C> is G,  &N4)$
2oid MFRC522KKMIFARET,etAccess its&
byte Saccess it u//er1    +ointer to byte 71 C and D in the sector trailer$
                           ytes 36$$25 will be set$

byte g61                  Access bits 3C4 C2 C>5 /or bloc. 6
                          &/or sectors 6->4) or bloc.s 6-= &/or sectors >2->B)

byte g41                  Access bits C4 C2 C>5 /or bloc. 4 &/or sectors 6->4)
                          or bloc.s 5-B &/or sectors >2->B)

byte g21                  Access bits C4 C2 C>5 /or bloc. 2 &/or sectors 6->4)
                          or bloc.s 46-4= &/or sectors >2->B)

byte g>                   Access bits C4 C2 C>5 /or the sector trailer1 bloc. >
                          &/or sectors 6->4) or bloc. 45 &/or sectors >2->B)

#he access bits are stored in a %eculiar /ashion$
#here are /our grou%sK

g3>5  Access bits /or the sector trailer1 bloc. > &/or sectors 6->4) or bloc. 45 &/or
sectors >2->B)

g325  Access bits /or bloc. 2 &/or sectors 6->4) or bloc.s 46-4= &/or sectors >2->B)
g345  Access bits /or bloc. 4 &/or sectors 6->4) or bloc.s 5-B &/or sectors >2->B)
g365  Access bits /or bloc. 6 &/or sectors 6->4) or bloc.s 6-= &/or sectors >2->B)

Each grou% has access bits 3C4 C2 C>5$ In this code C4 is M,  and C> is G, $
#he /our C* bits are stored together in a nibble c0 and an in2erted nibble c0T$
E0a'%leK
<< ,ector trailer that de/ines bloc.s 5 and 7 as Falue  loc.s and enables .ey  $
byte trailer u//er35 N U2551255125512551255125516161616 1 2551255125512551255125 5VA
<< @ee% de/ault .eys$
<< g4N7&i$e$446) NZ bloc. 5 2alue bloc.$ @ey   write]incre'ent1 A or   decre'ent$
<< g2N7 NZ ,a'e thing /or bloc. 7$
<< g>N> NZ @ey   'ust be used to 'odi/y the ,ector #railer$ @ey   beco'es 2alid$
'/rc522$MIFARET,etAccess its&]trailer u//er3751 61 71 71 >)A

XChec. i/ a card is %resent
bool MFRC522KK+I(I)-25764(2)-I55764( )-4.55764( )-4N55764(%)-4.55764(r)7w55911(I)-4.55764((I)-.55911(
<< boc.&/o%r,e/'oe$ 4.5576411.64 TdR[(<)-4.55764(<)-4& TdR[(<)-4.55764(<)-4\ TdR[(<)-4.55764(<)-4'557
```

```
,erial$%rint&O+ICC ty%eK O)A
,erial$%rintln&'/rc522$+ICCT;et#y%eNa'e&%icc#y%e))A
i/ & %icc#y%e \N MFRC522KK+ICCT#!+ETMIFARETMINI
     ]]  %icc#y%e \N MFRC522KK+ICCT#!+ETMIFARET4@
     ]]        %icc#y%e \N MFRC522KK+ICCT#!+ETMIFARET=@) U
     ,erial$%rintln&O#his sa'%le only wor.s with MIFARE Classic cards$O)A
     returnA
V

:ow to con2ert a nor'al bloc. into a 2alue bloc.
Usually a nor'al bloc. has 666 access bits1 while a 2alue bloc. has 446 access bits
EOa'%leK
byte 2alue loc.A   N 5A
byte 2alue loc.    N 7A
byte trailer loc.  N CA
<< le need a sector trailer with bloc.s 5 and 7 as Falue  loc.s and enables .ey  $
byte trailer u//er35 N U 25512551255125512551255
                         6161616125512551255125512551255VA << @ee% de/ault .eys$
<< g4N7 NZ bloc. 5 as 2alue bloc.$ @ey   to write]increment1 A or   /or decrement$
<< g2N7 NZ ,a'e thing /or bloc. 7$
<< g>N> NZ @ey   'ust be used to 'odi/y the ,ector #railer$ @ey   beco'es 2alid$
'/rc522$MIFARET,etAccess its&]trailer u//er3751 61 71 71 >)A
byte status N '/rc522$MIFARETIrite&trailer loc.1 trailer u//er1 47)A
i/ &status \N MFRC522KK,#A#U,TO@) U
      ,erial$%rint&OMIFARETIrite&) /ailedK O)A
      ,erial$%rintln&'/rc522$;et,tatusCodeNa'e&status))A
      returnA
V

:ow to setu% a Falue  loc. with a 2alue set to ?ero
EOa'%leK
byte  bloc.AddrN5A
byte 2alue loc.35 N U61616161 2551255125512551 61616161
                  bloc.Addr1_bloc.Addr1bloc.Addr1_bloc.Addr VA
byte status N '/rc522$MIFARETIrite&bloc.Addr1 2alue loc.1 47)A
i/ &status \N MFRC522KK,#A#U,TO@) U
      ,erial$%rint&OMIFARETIrite&) /ailedK O)A
      ,erial$%rintln&'/rc522$;et,tatusCodeNa'e&status))A
V
```