RF-ID

Materialet er **meget foreløbigt**.

Derfor modtages rettelser og forslag meget gerne. ☺

RF ID

RF-ID står for Radio Frequency Identification.

RF-ID teknologien kendes fra fx Natløb. Her får man en chip i skoen, og når man løber over start, registreres det af noget elektronik. Det samme sker når man kommer i mål, og herefter kan en computer udregne løbetiden.

Ansatte på skolen har et adgangskort, så vi kan komme ind uden for skolens åbningstid, eller logge ind på printerne. Kortene har indbygget RF-chips, eller TAGS, som de også kaldes.

Og der er RF-ID Tags på alle bøger på biblioteket, så en læser kan identificere bogen.

Skikortet i skijakken skal blot holdes op i nærheden af en læser.

Yderligere kan nævnes Brobizzer og Rejsekort.

Og efterhånden vil vi måske se tags i alle supermarkedets varer. Herved kan kasseapparatet hurtigt skanne hvad der er i indkøbsvognen, - og udregne en pris.

Men hvordan virker teknologien? Følgende er taget fra Wiki:

Fra Wiki: http://da.wikipedia.org/wiki/RFID

RFID er en forkortelse for Radio Frequency IDentification og en fællesbetegnelse for alle de teknologier, der anvender radiobølger til at identificere mennesker og objekter. Som det fremgår af navnet, så ligger RFID-teknologiens primære styrke i evnen til at identificere forskellige objekter.

RFID kaldes derfor også populært for den elektroniske eller fremtidens stregkode, da teknologien har nogle af de samme egenskaber som den traditionelle stregkode. Det betyder dog ikke at RFID-teknologien er bedre end stregkoden. Der er nemlig tale om to forskellige teknologier der har forskellige anvendelsesmuligheder, som dog til tider overlapper hinanden.

Den største forskel på teknologierne er at stregkoden er en line-of-sight teknologi, mens RFID er en trådløs teknologi. Med stregkode-teknologi skal scanneren have visuel kontakt med stregkoden for at aflæse den, men med RFID-teknologi kan RFID-taggen aflæses uden visuel kontakt, så længe den er inden for læsbar afstand.

Den læsbare afstand varierer alt afhængig af typen af RFID-tag, hvor nogle kan læses på op til 100 meters afstand, mens andre blot har en læseafstand på få cm.

I mange store og kommercielle RFID-systemer er informationen der udveksles dog blot et unikt ID-nummer, som også kaldes et EPC-nummer (Electronic Product Code). Dette ID-nummer anvendes som reference i en online database, hvor de reelle informationer om produktet ligger gemt. Dermed kan man, via en computer med internetadgang, få adgang til, opdatere og behandle disse informationer. Dette betyder desuden at størrelsen på RFID-taggens indbyggede hukommelse ikke er vigtig, da de egentlige data om objektet ligger online i en database og dermed i princippet kan fylde flere gigabyte.

Virkemåde: Passive og Aktive og teknologier.

Der findes 2 principielt forskellige typer RF-ID. De aktive, hvor en enhed aktivt sender signal til en modtagerantenne, som det fx sker i tidtagningssystemer til motocross-konkurrencer. Sende-enhederne har selv en forsyningskilde i form af et batteri. (Baseret på samtale med formanden på Nybølbanen)

De passive typer derimod skal have tilført energi udefra. Det får de via et vekslende magnetfelt. Langt de fleste systemer er af denne type.

Her ses et får, der er chipmærket.

Køer får også RF-ID chips, så foder- og malke-robotten kan kende koen.

Anvendte frekvenser:

Fælles for systemerne er, at de udveksler data ved hjælp af vekslende magnetfelter. Der findes forskellige standarder, hvor der anvendes forskellige frekvenser.

En oversigt:

Band	Regulations	Range	Data speed	Remarks
120-150 kHz (LF)	Unregulated	10 cm	Low	Animal identification, factory data collectionW
13.56 MHz (HF)	ISM band worldwide	1 m	Low to moderate	Smart cards
433 MHZ (UHF)	Short Range Devices	1-100 m	Moderate	Defence applications, with active tags
868-870 MHz (Europe) 902-928 MHz (North America)UHF	ISM band	1-2 m	Moderate to high	EAN, various standards
2450 MHz.5800 MHz (microwave)	ISM band	1-2 m	High	802.11 WLAN, Bluetooth standards
3.1 Ghz-10 GHz (microwave)	Ultra wide band	to 200 M	High	requires semi-active or active tags

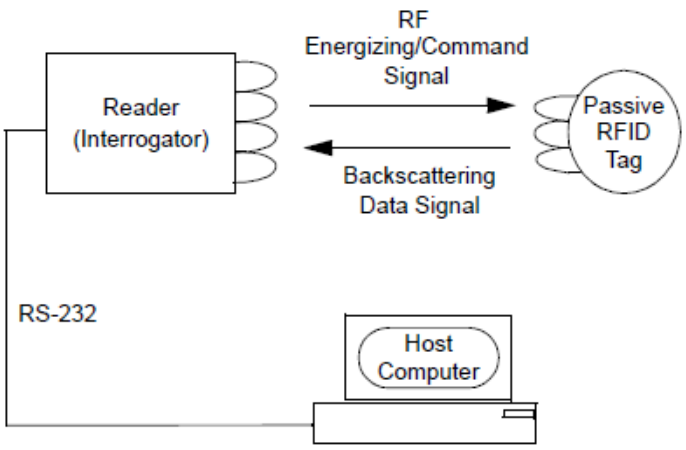
Kilde: https://en.wikipedia.org/wiki/Radio-frequency_identification

Virkemåden for Passive Tags:

Et passivt RF-ID-system er baseret på at en oscillator via en spole genererer et vekslende magnetfelt, fx på 125 KHz.

Når RF-chippen kommer ind i et passende veksel-magnetfelt, som genereres af læseren, vågner den op.

Det sker fordi der i taggen også er indbygget en spole, og heri genereres en vekslende spænding.



Den genererede veksel-spænding ensrettes, og reguleres ned til fx 5 Volt, som den ligeledes indbyggede microcontroller skal bruge.

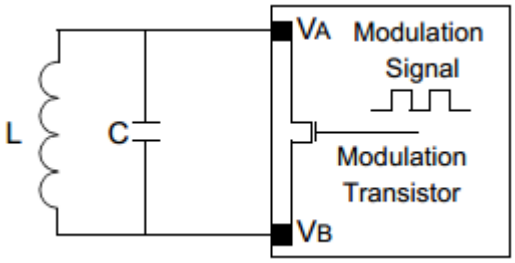
Når den indbyggede microcontroller liver op, begynder den at udføre det program, den er programmeret til at køre.

Her ses nogle eksempler på tags. Man kan se spolen, og ane microcontrolleren også.

Programmet i microcontrolleren sender et mønster af 1ére og 0ére ud på en pin. Dvs. Tag-ens kortnummer.

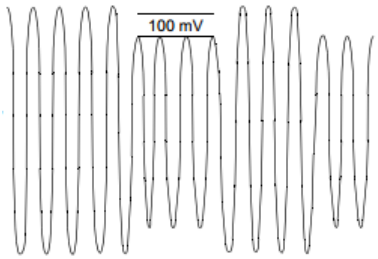
Pin-en er tilsluttet en transistor, der tændes og slukkes i mønsteret.

Transistoren tænder for strømmen i en modstand, forbundet mellem plus og nul i den genererede spænding i tag-en. Dvs. der bruges energi i et mønster. Og energien tages fra magnetfeltet, og det kan registreres i læseren, der jo genererer magnetfeltet.



Hvis transistoren er ON, dykker magnetfeltet.

Så det der sker er at Tag-en belaster magnetfeltet i et bestemt mønster, og på den måde sender en kode retur.

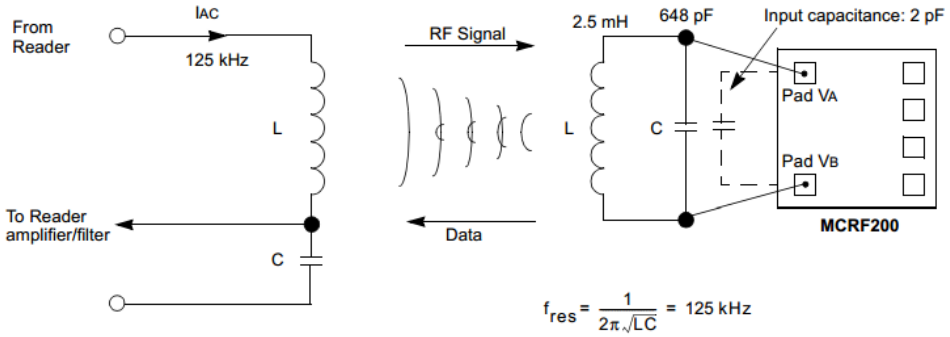


En graf af de 125 kHz i læseren, belastet i et mønster.

Læseren kan derfor tolke, hvilket nummer, taggen "sender retur" og sende det videre til en computer. Herefter kan computeren, hvis den er koblet til et databasesystem, vide noget om den vare eller person, der bærer tag-en.

Fx kan databasen indeholde oplysninger om et beløb, der er indbetalt, - og så trække prisen for frokost fra kontoen, der tilhører bæreren af RF-ID-kortet, sådan som det sker på skolen!

Det mere tekniske:



Her ses senderspolen og spolen og kondensatoren i en TAG.

En spole og en kondensator i parallel udgør en svingningskreds.

Kilde: http://ww1.microchip.com/downloads/en/devicedoc/51115f.pdf'

Med en svingningskreds skal forstås, at energien svinger mellem at være opmagasineret i magnetflet i spolen og som elektrisk felt i kondensatoren.

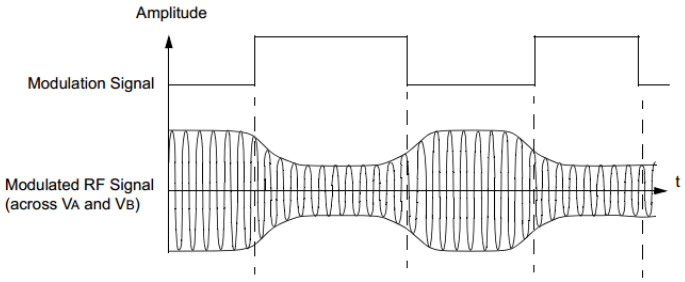
Frekvensen kan beregnes som $f = \frac{1}{2\pi\sqrt{LC}}$

Hvor L udtrykker spolens "størrelse" i Henry,
Og C er kondensatorens størrelse i Farad.

Svingningskredsen er afstemt til netop den frekvens, der arbejdes med.

Lidt ligesom en pendul i en bornholmerur er afstemt til en bestemt frekvens, som skal til for at uret går korrekt. Her svinger energi mellem potentiel energi når pendulet er længst ude, og kinetisk energi i bunden.

Her ses et eksempel på mønsteret fra taggens microcontroller, der udsender et modulationssignal, og det tilhørende magnetfelt.



Kilde: http://ww1.microchip.com/downloads/en/devicedoc/51115f.pdf'

RFID læser RDM630, købt hos Let Elektronik:

Her ses en samling af Tags.

Når de nærmer sig læseren, vågner de op, og "sender" deres kode.

Disse virker sammen med RDM630 fra Let-Elektronik

RF-ID læseren.

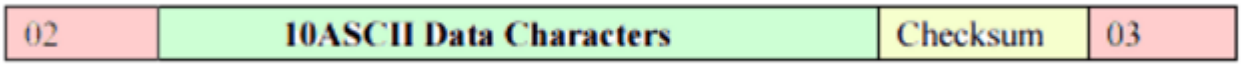
Når RF-ID læseren registrerer en tag i dens vekslende magnetfelt, og har læst dens kode, sender den koden via dens serielle port, dens UART, til en tilsluttet microcontroller, fx i en Arduino Uno.

Signalets Baudrate er 9600.
(9600bps,N,8,1)

På nogle ? af RF-ID-kortene - eller Tags, er der skrevet et flercifret decimaltal, et "Card-ID".
Hvis tallet omregnes til Hex, svarer det til det, der sendes fra læserens UART til en tilsluttet microcontroller.
Min dog ikke så ligetil !!

Når en tag kommer i nærheden af læseren, sender læseren 14 byte, dvs. 14 digits via dens UART.

Først sendes en start-karakter, som er 02h. Dernæst sendes 10 digit TAG-nummer, 2 byte med checksum og til slut en slut-karakter på 03h. Dvs. i alt sendes 14 byte.



Header 10 byte data 2 Byte Checksum Slutbyte

Eksempel:

På en af de gule Tags er der stemplet nummeret 757383. Det er et decimalt nummer, og hvis det omregnes til Hex fås værdien 000B8E87.

For konvertering fra decimal til Hex se fx:

http://www.binaryhexconverter.com/decimal-to-hex-converter

Eksempel 1:
På kortet læses bortset fra Header og footer de 10 byte som 01000B8E87. De er gemt som Hex-numre kodet i ASCII. (hvorfor lige 01 foran ???)

Dvs. det, der sendes fra læseren er først en header med værdien 02h.

Dernæst sendes et 0-tal som ASCII, dvs. 30h, så 31h, osv.

De 10 cifre er altså: 30h, 31h, 30h, 30h, 30h, 42h, 38h, 45h, 38h, 37h

Så Checksum, der her kan udregnes som 01h XOR 00h XOR 0Bh XOR 8Eh XOR 87h = 03h.

03 sendes også som ASCII, dvs. 30h og 33h

Og endelig sendes slutkoden 03h.

Det er vist her:

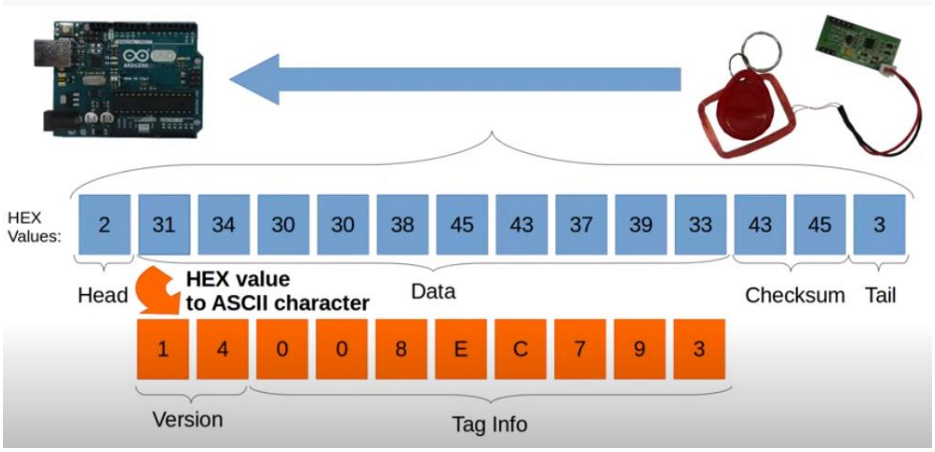
Læst kode, i Binær, Hex og decimal.
Fra Arduinos Debugvindue !!

Udsnit af ASCII tabellen.

```
2
110000  30  48
110001  31  49
110000  30  48
110000  30  48
110000  30  48
1000010  42  66
111000  38  56
1000101  45  69
111000  38  56
110111  37  55
110000  30  48
110011  33  51
11  3  3
TAG code is: 01000B8E8703
```

Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
32	20	040		Space	64	40	100	@	@
33	21	041	!	!	65	41	101	A	A
34	22	042	"	"	66	42	102	B	B
35	23	043	#	#	67	43	103	C	C
36	24	044	$	$	68	44	104	D	D
37	25	045	%	%	69	45	105	E	E
38	26	046	&	&	70	46	106	F	F
39	27	047	'	'	71	47	107	G	G
40	28	050	(	(	72	48	110	H	H
41	29	051	)	)	73	49	111	I	I
42	2A	052	*	*	74	4A	112	J	J
43	2B	053	+	+	75	4B	113	K	K
44	2C	054	,	,	76	4C	114	L	L
45	2D	055	-	-	77	4D	115	M	M
46	2E	056	.	.	78	4E	116	N	N
47	2F	057	/	/	79	4F	117	O	O
48	30	060	0	0	80	50	120	P	P
49	31	061	1	1	81	51	121	Q	Q
50	32	062	2	2	82	52	122	R	R
51	33	063	3	3	83	53	123	S	S
52	34	064	4	4	84	54	124	T	T
53	35	065	5	5	85	55	125	U	U
54	36	066	6	6	86	56	126	V	V
55	37	067	7	7	87	57	127	W	W
56	38	070	8	8	88	58	130	X	X
57	39	071	9	9	89	59	131	Y	Y
58	3A	072	:	:	90	5A	132	Z	Z
59	3B	073	;	;	91	5B	133	[	[
60	3C	074	<	<	92	5C	134	\	\
61	3D	075	=	=	93	5D	135	]	]
62	3E	076	>	>	94	5E	136	^	^
63	3F	077	?	?	95	5F	137	_	_

Her et andet eksempel:



Fra: https://www.youtube.com/watch?v=l8RDbHd1cak

Eksempel på, at udregne checksum:

Example: card number: 62E3086CED

- Header: 02h
- Output data:36h、32h、45h、33h、30h、38h、36h、43h、45h、44h
- CHECKSUM: 30h、38h
- Footer: 03h

Tjecksummen er udregnet ud fra kortets Card nummer og kan bruges som sikkerhedskontrol at kortet er læst rigtigt !!

Det udregnes ved at opdele kortets nummer i 5 byte, her 62h, E3h, 08h, 6Ch og EDh. De bliver så XOR-ed sammen

(62h) XOR (E3h) XOR (08h) XOR (6Ch) XOR (EDh)=08h

Udregning:

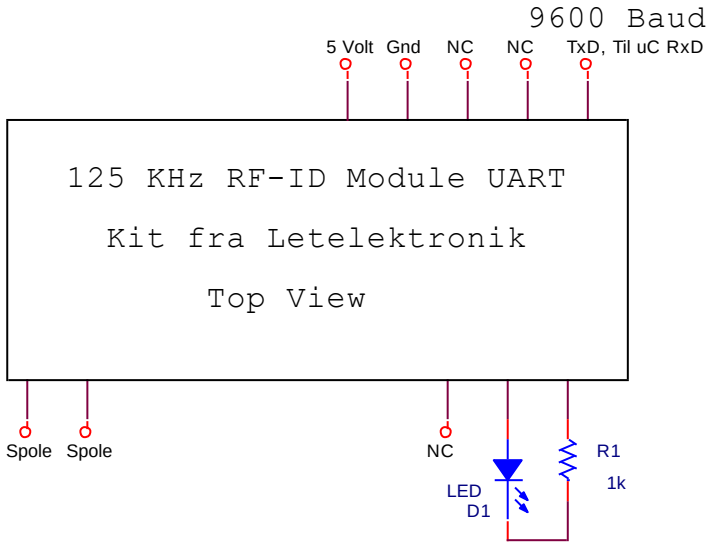
	Binær	Hex
	0110 0010	62
Exor	1110 0011	E3
=	1000 0001	
Exor	0000 1000	08
=	1000 1001	
Exor	0110 1100	6C
=	1110 0101	
Exor	1110 1101	ED
=	0000 1000 = 08h	

08h sendes nu som ASCII, dvs. 30h og 38h

Og endelig sendes en Footer med værdien 03h

Sammenhængen mellem forskellige Tag numre og deres kode:

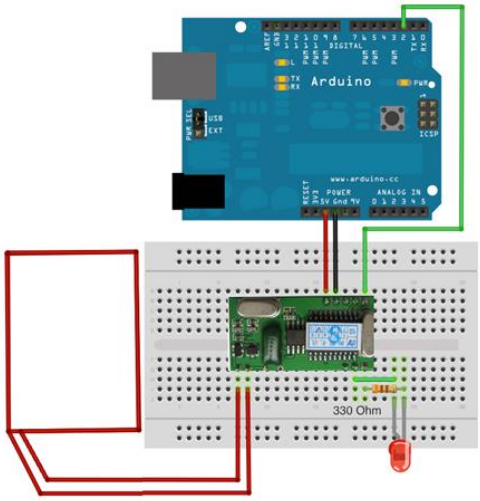
Tag:	Decimal nummer stemplet på Tag'en	Hex kode på kortet	Checksum (Hex)
Rød:	13502948	3C - 00CE09E4	1F
Gul	757383	01 - 000B8E87	03
Blå	4627040	3D - 00469A60	81
Hvid kort 33	4060429 061 - 62733	41 - 003DF50D 41 - 003D - F50D	84
Hvid kort 38	0004056634 061 - 58938	41 - 003DE63A 41 - 003D - E63A	A0



Her ses ben-forbindelserne på RF-ID læser-kittet RDM630.

Bemærk: Top View !!!

Se datablad: http://www.let-elektronik.dk/filer/produkter/tradlos_rfid_125_uart_module.pdf



Et eksempel på arduinos forbindelse til læser:

Her er der brugt pin 2 på Arduinoen !!

Den røde er 5 Volt, sort er Gnd, og grøn er signal fra læserens UART til Arduino

Der er desværre lidt forvirring omkring hvilken pin, læseren anvender som sendepin !!

Det skal lige måles med et oscilloscop når der laves testopstillinger.

OBS:

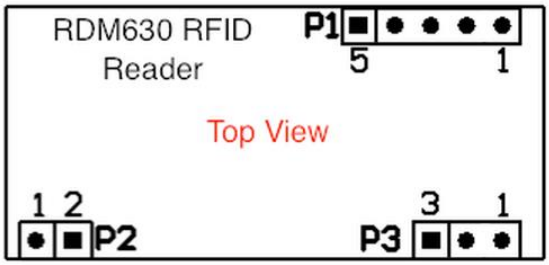
Obs: der kan være forskel på benforbindelserne på de to modeller vi har af RDM630.

Der er 2 udgaver af RDM630

Endnu nyere udgave: RDM6300. Men er en anden type !!

De gamle typer sender det læste fra Tagén 1 gang, RDM6300 sender samme tekst flere gange mens tag-en er i feltet. Iflg YouTube: https://www.youtube.com/watch?v=l8RDbHd1cak

Ligesom LED-en ikke går ud for RDM6300 !! ???



P1	P2	P3
PIN1---TX	PIN1---ATN1	PIN1---LED
PIN2---RX	PIN2---ATN2	PIN2---+5v(DC)
PIN3		PIN3---GND
PIN4---GND		
PIN5---+5v(DC)		

Forbindelserne til model RDM630: New!

Der kan vist være forskel på pins imellem versionerne !!

Kodeeksempler:

Her følger nogle kodeeksempler, der kan kopieres over i Arduinos IDE. De er ikke redigeret, og ikke testet, - men med her som inspiration !!

```
/*
Kode lavet af Valle Thorø, 08-04/2014
 Virker til RDM630-læser

 Tagsene har en Header på 02hex, 10 byte data gemt som ASCII, 2 Byte Checksum og endelig en
 End of data på 03h

 Programmet skriver i Debugvinduet, hvilken Tag, der læses. Dets farve og kortnummer
 */

#include <SoftwareSerial.h>

byte  val = 0;
int tal = 0;
char code[13];
int bytesread = 0;

#define rxPin 8
#define txPin 9

SoftwareSerial RFIDSerial(rxPin,txPin);

void setup()
{
    Serial.begin(9600);  // Hardware serial for Debug 9600bps
    RFIDSerial.begin(9600);    // definer softserial
}

void loop()
{
    if(RFIDSerial.available() > 0)
    {
        if((val = RFIDSerial.read()) == B00000010) // Vent på Header
        {
            Serial.println("Hej");  // For Debug
```

```
            Serial.println(val);
            bytesread = 0;
            while(bytesread<13)
            {  // read digit code
                val = RFIDSerial.read();
                code[bytesread] = val;         // gem digit
                bytesread++;
                Serial.print(val, BIN);
                Serial.print("  ");
                Serial.print(val, HEX);
                Serial.print("  ");
                Serial.println(val, DEC);
                // ready to read next digit
            }
            //digit read is complete
            Serial.print("TAG code is: ");
            Serial.write(code);            // print TAG code
            Serial.print(" ");
            tal = (code[6]);
            Serial.print(tal);
            tal = (code[7]);
            Serial.print(tal);
            tal = (code[8]);
            Serial.print(tal);
            tal = (code[9]);
            Serial.print(tal);
            // Tjek for Kortnummer, de to sidste byte
            if((code[8]==51) && (code[9]==65)) Serial.println("   Hvid kort 38");
            if((code[8]==56) && (code[9]==55)) Serial.println("    Gul");
            if((code[8]==48) && (code[9]==68)) Serial.println("   Hvid kort 33");
            if((code[8]==54) && (code[9]==48)) Serial.println("   Blaa ");
            if((code[8]==69) && (code[9]==52)) Serial.println("   Roed");

            bytesread = 0;
            Serial.println("  ");
            Serial.println(" Naeste");
        }
    }
}
```

Eksempel, Ikke testet / Valle

```
/*********************************************************************
** Modified code to work with Arduino 1.0.1                                    /
**
** Credits to Petushka, http://www.instructables.com/member/Petushka/ **
**
                                                                        **
*********************************************************************/

#include <SoftwareSerial.h>

#define ADD_TAG_CODE "210014DFE309"  //change this ID with your own card TAG
#define DEL_TAG_CODE "210014E2BD6A"  //change this ID with your own card TAG

SoftwareSerial rfid = SoftwareSerial(5, 6);
String msg;
String ID ;  //string to store allowed cards

void setup()
{
  Serial.begin(9600);
  Serial.println("Serial Ready");

  rfid.begin(9600);
  Serial.println("RFID Ready");
}

char c;

void loop(){
```

```
  while(rfid.available()>0){
    c=rfid.read();
    msg += c;
    Serial.println(msg);
    Serial.println(msg.length());
  }
  msg=msg.substring(1,13);
  if(msg.indexOf(ADD_TAG_CODE)>=0) add();
  else if(msg.indexOf(DEL_TAG_CODE)>=0) del();
  else if(msg.length()>10) verifica();
  msg="";

}

void add(){
  Serial.print("What TAG do you wanna grant access?: ");
  msg="";
  while(msg.length()<13){
    while(rfid.available()>0){
      c=rfid.read();
      msg += c;
    }
  }
  if(ID.indexOf(msg)>=0) {
    Serial.println("\nAccess already granted for this card.");
    msg="";
  }
  else{
    Serial.print("Card: ");
    Serial.println(msg);
    ID += msg;
    ID += ",";
    //Serial.print("ID: ");
   // Serial.println(ID);
    msg="";
    Serial.println("Access granted for this card.");
  }

}

void del(){
  msg="";
  Serial.print("What TAG do you wanna deny access?: ");
  while(msg.length()<13){
    while(rfid.available()>0){
      c=rfid.read();
      msg += c;
    }
  }
  msg=msg.substring(1,13);
  if(ID.indexOf(msg)>=0){
    Serial.println(msg);
    Serial.println("TAG found. Access for this card denied.");
    //ID.replace(card,"");
    int pos=ID.indexOf(msg);
    msg="";
    msg += ID.substring(0,pos);
    msg += ID.substring(pos+15,ID.length());
    ID="";
    ID += msg;
    //Serial.print("ID: ");
    //Serial.println(ID);
  } else Serial.println("\nTAG not found or already denied");
  msg="";
}

void verifica(){
    msg=msg.substring(1,13);
    if(ID.indexOf(msg)>=0) Serial.println("Access granted.");

    else Serial.println("Access denied.");
}
```

Eksempel, Ikke testet / Valle

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 10, 5, 4, 3, 2);
unsigned char card[12];
void setup()
{
Serial.begin(9600);
lcd.print("Card ID :");
}
void loop()
{
unsigned char i=0;
for(;;)
{
if (Serial.available()>0)
{
card[i]=Serial.read();
i++;
}
if (i>=11)
{
i=0;
decode();
}
}
}
void decode()
{
lcd.setCursor(0, 1);
unsigned char p;
for(p=3;p<11;p++)
{
lcd.print(card[p]);
}
}
```

Eksempel, Ikke testet / Valle

```
#include "NewSoftSerial.h"
#define stx 2
#define etx 3

NewSoftSerial mySerial(6, 7);
int counter;
byte data[14];
byte hexBlock1,hexBlock2,hexBlock3,hexBlock4,hexBlock5;
byte hexCalculatedChecksum,hexChecksum;

void setup() {
  Serial.begin(9600);
  mySerial.begin(9600);
}

void loop() {
  if (mySerial.available() > 0) {
    data[counter] = mySerial.read();
    counter++;
    if(counter > 13) {
      //we read the whole message, so reset counter
      counter = 0;
      //check if start of text and end of text is correct
      if(data[0] == stx && data[13] == etx) {
        Serial.println("Start of text and end of text correctly received.");
        Serial.print("ID: ");
        //show ID
        for(int x = 1; x < 11; x++) {
          Serial.print(data[x], BYTE);
        }
        Serial.println("");
```

```
        Serial.print("Checksum: ");
        //show checksum
        Serial.print(data[11], BYTE);
        Serial.println(data[12], BYTE);

        //Hex ID blocks. Two transmitted Bytes form one Hex ID block.
        //Hex ID blocks:      6   2  |  E   3  |  0   8  |  6   C  |  E   D
        //Transmitted Bytes: 36H 32H | 45H 33H | 30H 38H | 36H 43H | 45H 44H
        hexBlock1 = AsciiCharToNum(data[1])*16 + AsciiCharToNum(data[2]);
        hexBlock2 = AsciiCharToNum(data[3])*16 + AsciiCharToNum(data[4]);
        hexBlock3 = AsciiCharToNum(data[5])*16 + AsciiCharToNum(data[6]);
        hexBlock4 = AsciiCharToNum(data[7])*16 + AsciiCharToNum(data[8]);
        hexBlock5 = AsciiCharToNum(data[9])*16 + AsciiCharToNum(data[10]);

        //Transmitted checksum.
        hexChecksum = AsciiCharToNum(data[11])*16 + AsciiCharToNum(data[12]);

        //XOR algorithm to calculate checksum of ID blocks.
        hexCalculatedChecksum = hexBlock1 ^ hexBlock2 ^ hexBlock3 ^ hexBlock4 ^ hexBlock5;
        if ( hexCalculatedChecksum == hexChecksum )
        {
          Serial.println("Calculated checksum matched transmitted checksum.");
        }
        else {
          Serial.println("Calculated checksum didn't match transmitted checksum. Corrupt data!");
        }
      }
    }
  }
}

uint8_t AsciiCharToNum(byte data) {
  //First substract 48 to convert the char representation
  //of a number to an actual number.
  data -= '0';
  //If it is greater than 9, we have a Hex character A-F.
  //Substract 7 to get the numeral representation.
  if (data > 9)
    data -= 7;
  return data;
}
```

Eksempel, Ikke testet / Valle

```
/*--------------------------------------------------------------------
This is a sample code for RDM630 RFID reader by Spekel(Spekel.se)
This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported
License.
http://creativecommons.org/licenses/by-nc-sa/3.0/
--------------------------------------------------------------------*/
#include <SoftwareSerial.h>
#define rxPin 2
#define txPin 3
char code[20];
int val = 0;
int bytesread = 0;
//------------------------------------
//create a Serial object RFID
SoftwareSerial RFID= SoftwareSerial(rxPin, txPin);

void setup()
{
  Serial.begin(9600);
  Serial.println("Serial Ready");
  RFID.begin(9600);
  Serial.println("RFID Ready");
  pinMode(rxPin, INPUT);
  pinMode(txPin, OUTPUT);
}
void loop()
{
  val = 0;
```

```
  bytesread = 0;

  while(bytesread < 12)
  {
    // read 12 digit code
    val = RFID.read();
    if(val == 3)
    { // if header or stop bytes before the 10 digit reading
      break; // stop reading
    }

    if(val != 2)
    {
      code[bytesread] = val; // add the digit
      bytesread++; // ready to read next digit
      code[bytesread] = '\0'; // add the NULL
    }
  }

  if(bytesread >= 12)
  { // if 12 digit read is complete
    Serial.print("Tag: [");
    for(int i=0; code[i]!='\0' ; i++)
    {
      Serial.print(code[i]);
    }
    Serial.println("]"); //print the whole 13 bytes
  }
}
```

Links til materiale til Arduino:

Se: http://www.seeedstudio.com/wiki/125Khz_RFID_module_-_UART

http://tronixstuff.com/2013/11/19/arduino-tutorials-chapter-15-rfid/ (God !!)

Se fx: RF-ID: Se: http://tronixstuff.wordpress.com/2010/08/18/moving-forward-with-arduino-chapter-15-rfid-introduction/

Kodeeksempler:
http://forum.arduino.cc/index.php/topic,148029.0.html
http://forum.arduino.cc/index.php?topic=113970.10;wap2
http://arbitraryuser.com/2013/04/16/rdm630-125khz-rfid-reading-with-the-arduino-mega-2560-r3/
Eks. med checksum: http://marioboehmer.blogspot.dk/2011/01/rfid-with-arduino.html

Se video om RFID generelt, 4:31: https://www.youtube.com/watch?v=gEQJxNDSKAE

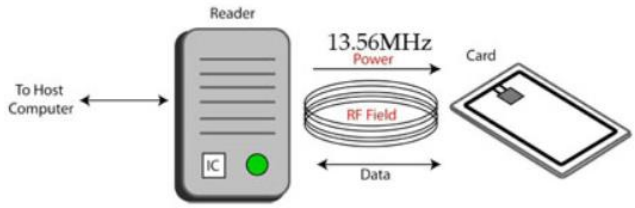
Aktive Tags.

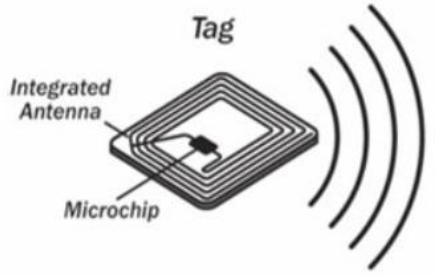
For en ordens skyld er her vist et par eksempler på aktive tags. De har batterier om bord, og sender derfor selv, uden at skulle lives op i et veksel-magnet-felt.

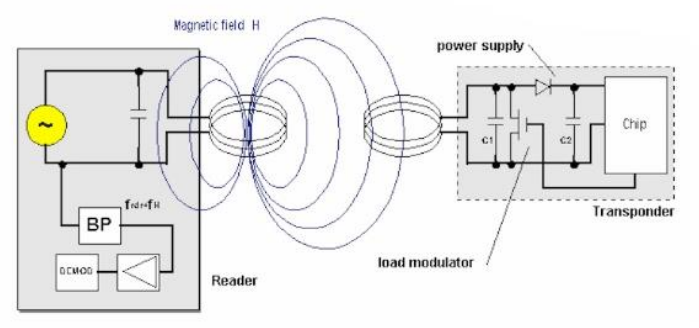
Kuffert med aktive tags, og til højre ses en placeret på en crosser.

Mere til historien / 16-11-2021

RFID, RF-ID / NFC, Near Field Communication






Magnetic field H
power supply
Chip
C1
C2
Transponder
BP
DEMOD
Reader
load modulator

"reader" **"tag"**