1 RFID :Identification Radio-fréquence

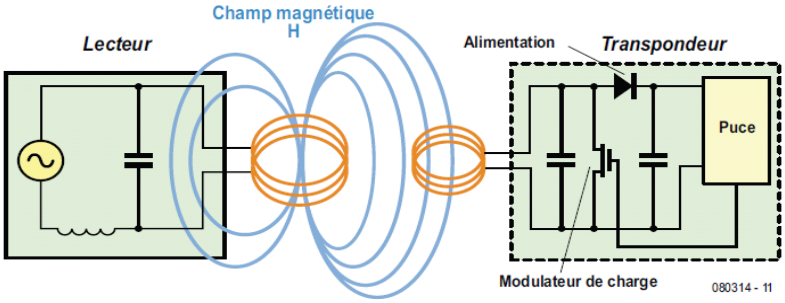
Un objet est identifié sans contact grace à un « Tag » qui est lu par une station fixe.

Le « Tag » peut être à lecture seule ou à lecture-écriture. On le nomme TRANSPONDEUR .

La technologie et les fréquences employées donnent des caractéristiques assez différentes.

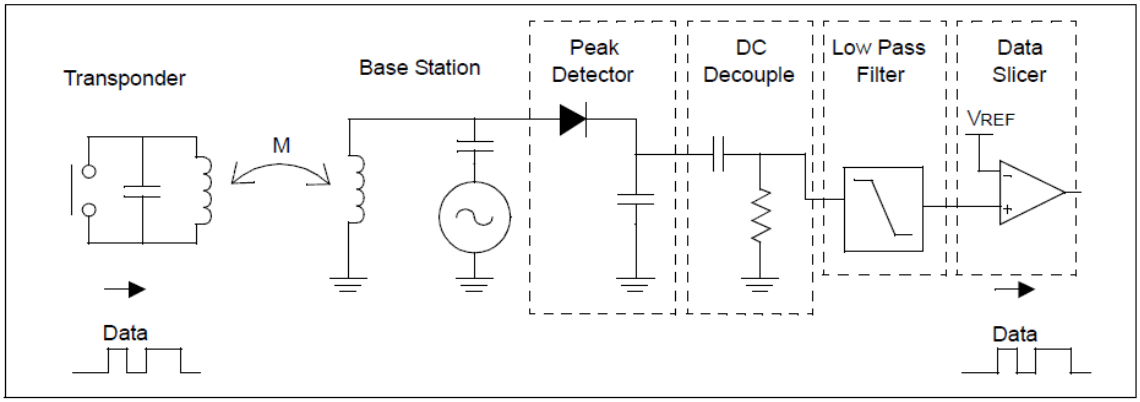
Le « Tag » est généralement passif, il reçoit son énergie par onde haute fréquence (HF) de la station de lecture.

Principe de la liaison « Tag » - lecteur (Source Elektor) :



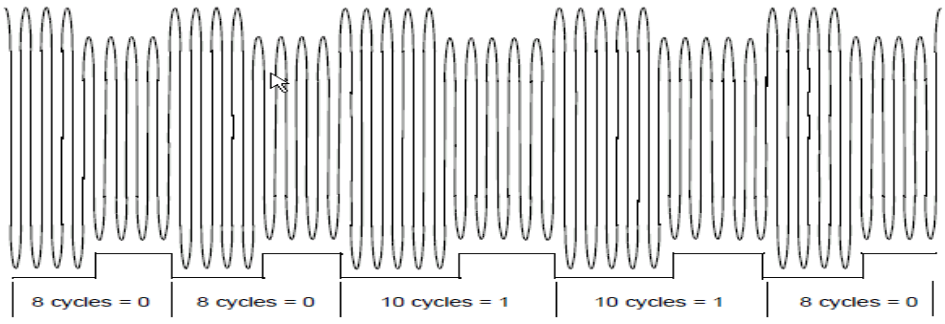
Par le biais du Modulateur de charge, le Tag fait varier la charge de l'émetteur (le Lecteur).

Une autre représentation (source Microchip) :



Celle-ci montre mieux la structure d'un lecteur analogique.

Exemple de chronogramme de la tension à l'entrée du détecteur :

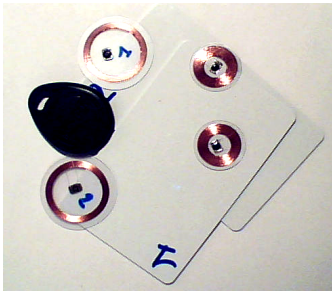


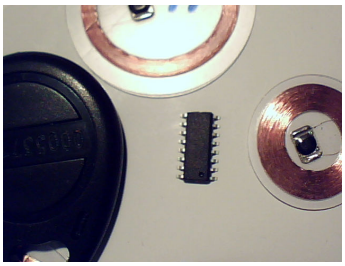
Le signal « porteur » a une fréquence fixe bien définie : celle qui est fournie par l'émetteur, comme pour la radiophonie.

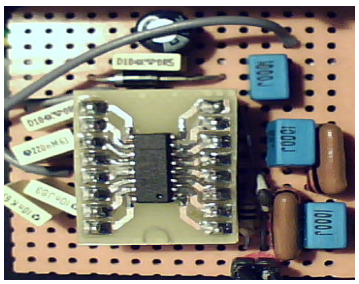
Les variations d'amplitude permettent de délimiter des durées : un « 0 » logique dure 8 périodes et le « 1 » dure 10 périodes.

Notre information est donc portée par une modulation de fréquence (FSK).

2 Le circuit utilisé : EM4095







Parmi les fréquences utilisées par la RFID, on a choisi la plus basse (125KHz) car c'est celle qui est accessible à nos appareils de mesure.

Les « Tags » utilisés sont ceux qui sont utilisés depuis longtemps pour les contrôles d'accès.

Leur fréquence basse impose des bobines importantes et leur portée est limitée. On leur préfère aujourd'hui des fréquences plus élevées : 13,56MHz, 865MHz, 2,45GHz ou 5,8GHz.

Les tags utilisés utilisent le protocole EM4100.

Un circuit intégré EM4095 permet de confectionner un lecteur-enregistreur assez simplement. Seul défaut : sa taille, en boitier « SOIC 18 », les broches sont espacées de 0,87mm …

Ci – contre le circuit seul, au milieu des Tags.

Ci dessous le circuit sur un adaptateur qui a permis les essais sur une platine d'espérimentation.

Puis réalisation d'un prototype. Comme on n 'envisage pas de farication en série, il n'y a pas eu de recherche de circuit imprimé.

La confection de la bobine et les valeurs des composants ont été déterminés par la note du constructeur AN404.

On peut toutefois acquérir des bobines « toute prêtes » auprès des marchands.

Typical Operating Configuration
Read Only Mode

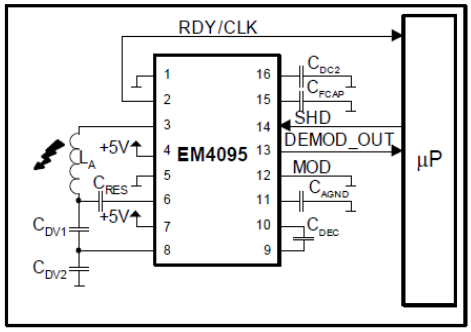


Fig. 1

Pin Assignment

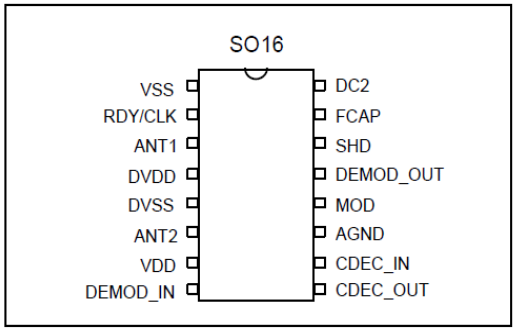
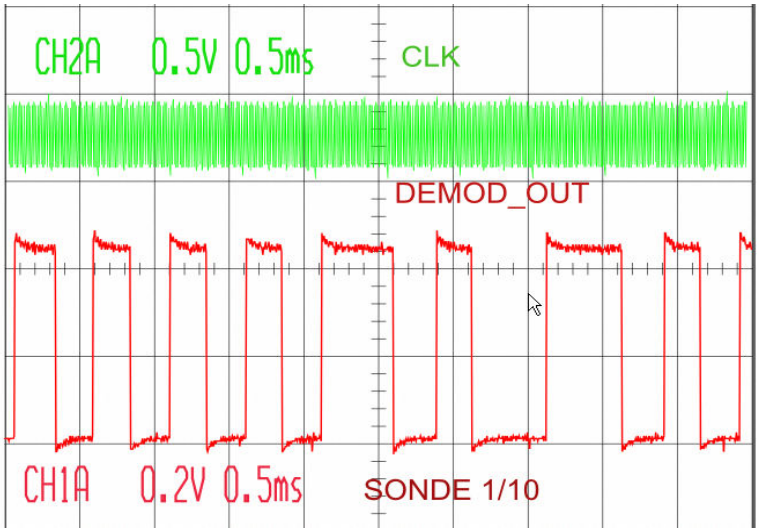


Fig. 3

Fonctionnement :



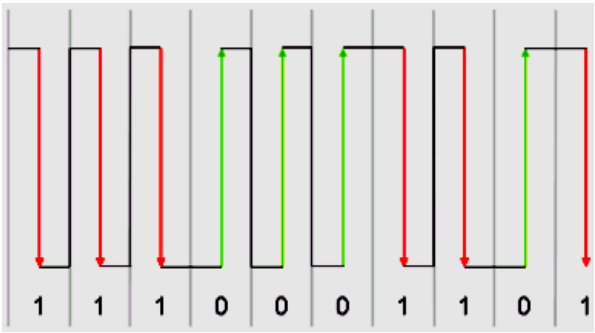
Sur le relevé oscilloscopique ci contre on voit :

Le signal CLK à 125KHz.

Le signal logique sur la sortie « DEMOD-OUT », de niveaux 0V et 5V. C'est celui-ci qui contient le code du Tag.

La durée minimale d'un état est d'environ 250µs, ce qui nous permet de dire que le décodage sera possible avec un Arduino. Le codage n 'est pas « FSK » comme dans le chronogramme de la page 1, mais « Manchester », qui est plus délicat à décoder.

3 *Le décodage :*

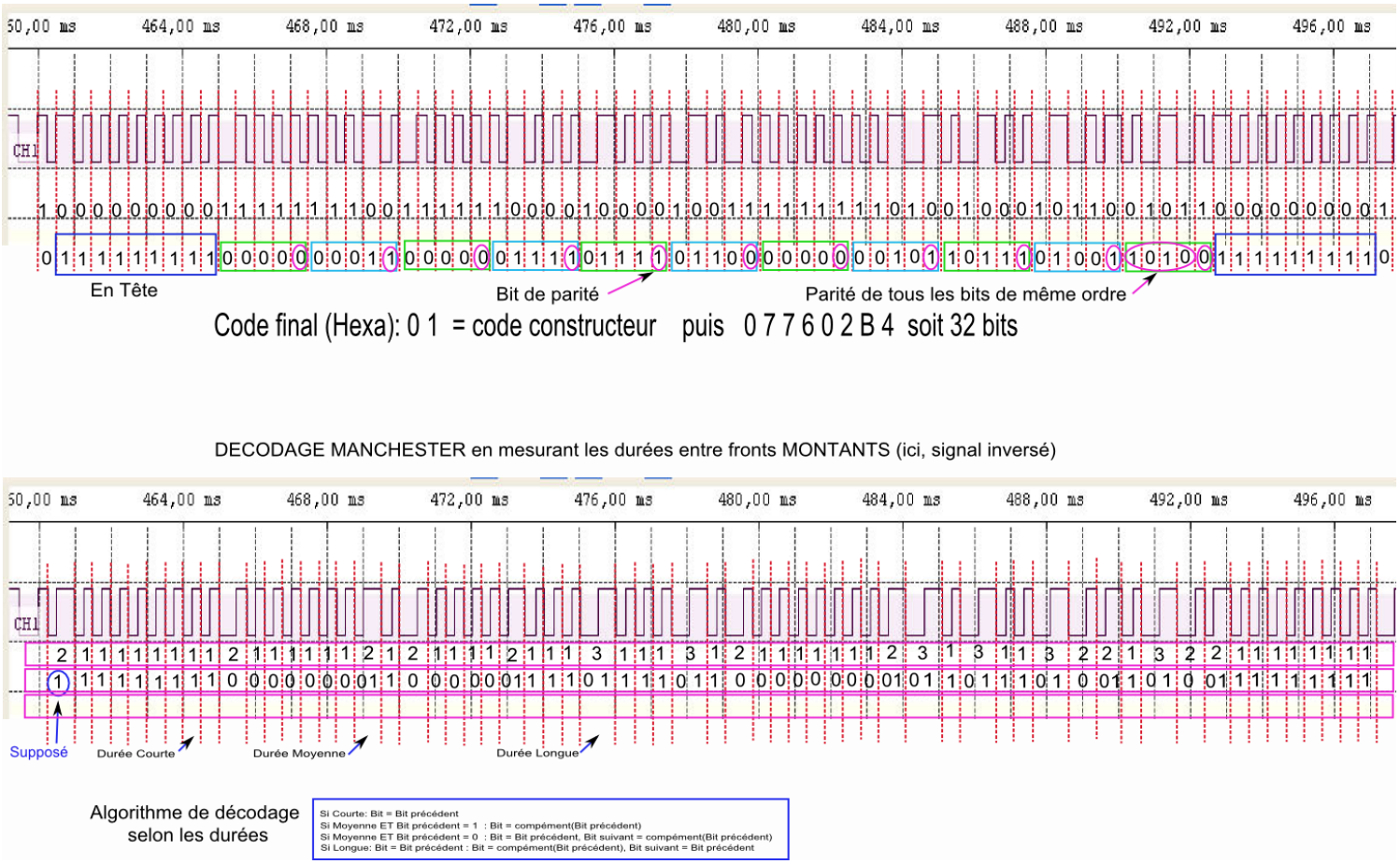


Le codage Manchester, d'après Wikipedia, est un codage synchrone.

C'est à dire que les signaux sont liés à une HORLOGE, ici représentée par les traits bleus, régulièrement espacés.

C'est le SENS du FRONT entre deux « tops » d'horloge qui donne la valeur du Bit.

Décodage pratique sur un exemple (carte N°1)



ATTENTION : pour des problèmes de constitution des circuits électriques, les signaux de la sortie DEMOD_OUT, représentés ici sont inversés : il faudrait faire une symétrie de l'image selon un axe horizontal.

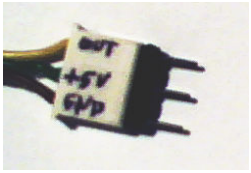
Sur le chronogramme du haut, décodage « intelligent » par analyse humaine : on retrouve la position des « tops » d'horloge par l'analyse de l'image globale.

Pour retrouver la signification des bits et le code de la carte, il faut connaître la norme EM4100.

Sur le chronogramme du bas : l'analyse que va effectuer le programme, en mesurant la durée entre deux fronts montants (descendants sur l'image …). Il devient difficile de repérer les « 0 » et les « 1 » … On utilise alors l'algorithme de Mr Vern. Noter que ce code Manchester est aussi utilisé pour les télécommandes pour le protocole « RC5 ».

Nous verrons dans le programme, que la difficulté réside dans la reconnaissance de l'en-tête du code.

4 *Le décodage dans l'Arduino*



Les connexions à l'Arduino sont extrèmement réduites : +5V, GND et DEMOD_OUT connectée à Pin2 (entrée utilisable pour les interruptions).

Pour l'instant nous n'utilisons que la fonction LECTURE de l'EM4095.

L'Arduino transmet ses messages au PC par l'intermédiaire de la liaison USB (qui alimente également tous les circuits).

Listing du programme : Lect 4095.pde

à ouvrir dans l'IDE Arduino ou dans un éditeur de texte (Notepad ++ de préférence)

Le programme est une boucle infinie, composée d'une suite d'étapes :

- **Lecture de 100 durées successives** dans un tableau (réservé pour ça et nommé « Buffer »)
  Noter que la transmission complète d'un code comporte 64 bits < 100.
- **Vérification des durées** et tri entre 3 valeurs :
  Minimale , Moyenne (=1,5xminimale) et Grande (=2xMinimale)
  et … fausse … quand la liaison HF carte – lecteur est défectueuse.

S'il n'y a pas eu d'erreurs :

- **Recherche de l'en-tête** (9bits), avec au moins 64 bits suivants dans le buffer.

Si l'entête a été trouvé :

- **Décodage du code Manchester** qui donne (64-9) bits.
- **Vérification des bits** en utilisant les bits de contrôle.
  On peut trouver des erreurs ici parce que l'entête a été mal placé.

Si la vérification est positive :

- **Conversion du code de binaire vers hexadécimal.**
- **Affichage du code.**

Si on rencontre des erreurs « en route », des messages sont affichés.

5 *Un autre exemple de lecteur à base d'Arduino :*

sur le site de référence de l'Arduino : ***DIY FSK RFID Reader***

Encore plus fort : l'Arduino génère lui même le signal à 125KHz et assure la détection !