

Voici quelques exemples de conversion binaire décimal :

TEMPERATURE	DIGITAL OUTPUT (Binary)	DIGITAL OUTPUT (Hex)
+125°C	0000 0111 1101 0000	07D0h
+85°C*	0000 0101 0101 0000	0550h
+25.0625°C	0000 0001 1001 0001	0191h
+10.125°C	0000 0000 1010 0010	00A2h
+0.5°C	0000 0000 0000 1000	0008h
0°C	0000 0000 0000 0000	0000h
-0.5°C	1111 1111 1111 1000	FFF8h
-10.125°C	1111 1111 0101 1110	FF5Eh
-25.0625°C	1111 1110 0110 1111	FE6Fh
-55°C	1111 1100 1001 0000	FC90h

Figure 7 : Analogie température- code binaire

On voit rapidement que les 8 bits centraux codent directement la partie entière de la température.

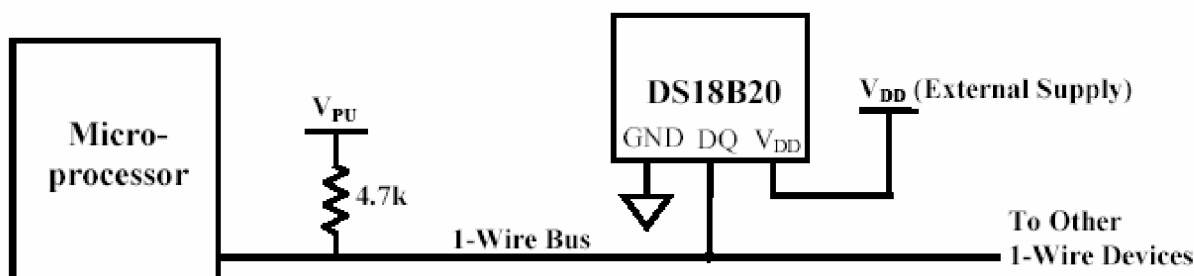


Figure 8 : Interfaçage du DS18B20

Le mode de communication 1-fil a comme son nom l'indique une unique ligne de transmission. Chaque périphérique (dans ce cas le microcontrôleur et le Dallas) communique par cette ligne via un drain ouvert ou un port 3 états (0,1, état « idle »). Cela permet à chaque périphérique de pouvoir relâcher la ligne quand celui-ci ne transmet pas de donnée et ainsi laisse le bus libre pour l'autre périphérique. On voit sur le schéma ci-dessus que l'état « idle » est à l'état haut à cause de la pull-up.

Pour commander un DS18B20, l'utilisateur doit respecter 3 étapes :

- l'Initialisation du DS
- Commande de ROM
- Commande de fonction

A noter que toutes les données et les commandes sont transmises avec le LSB en premier.

L'initialisation du DS consiste à mettre à 0 la ligne pendant 480µs et si le capteur est présent il applique à son tour l'état bas pendant une certaine durée.

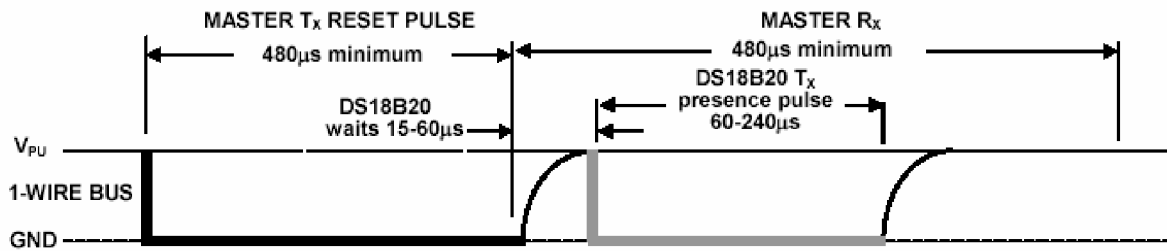


Figure 9 : Séquence d'initialisation

La commande de ROM sert à sélectionner le DS18B20 avec lequel on veut parler, dans notre cas il n'y en a qu'un c'est pourquoi nous enverrons à chaque fois « Skip ROM » 0xCC. La commande de fonction indique au capteur ce qu'il doit réaliser, comme par exemple faire une prise et une conversion de température.

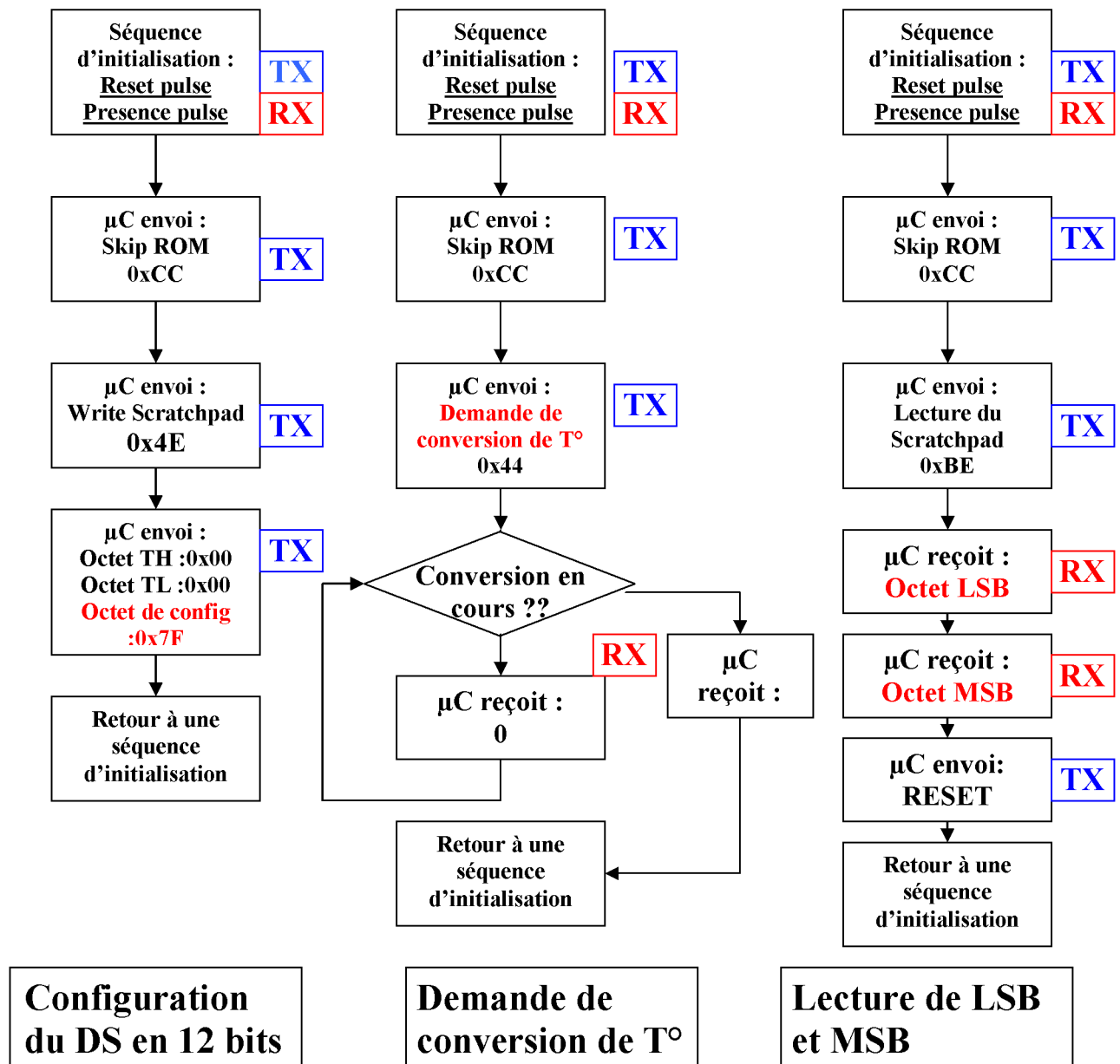


Figure 10 : Organigramme de configuration du DS18B20 et de prise de mesure

- Lecture et écriture de donnée avec le protocole 1-fil de dallas

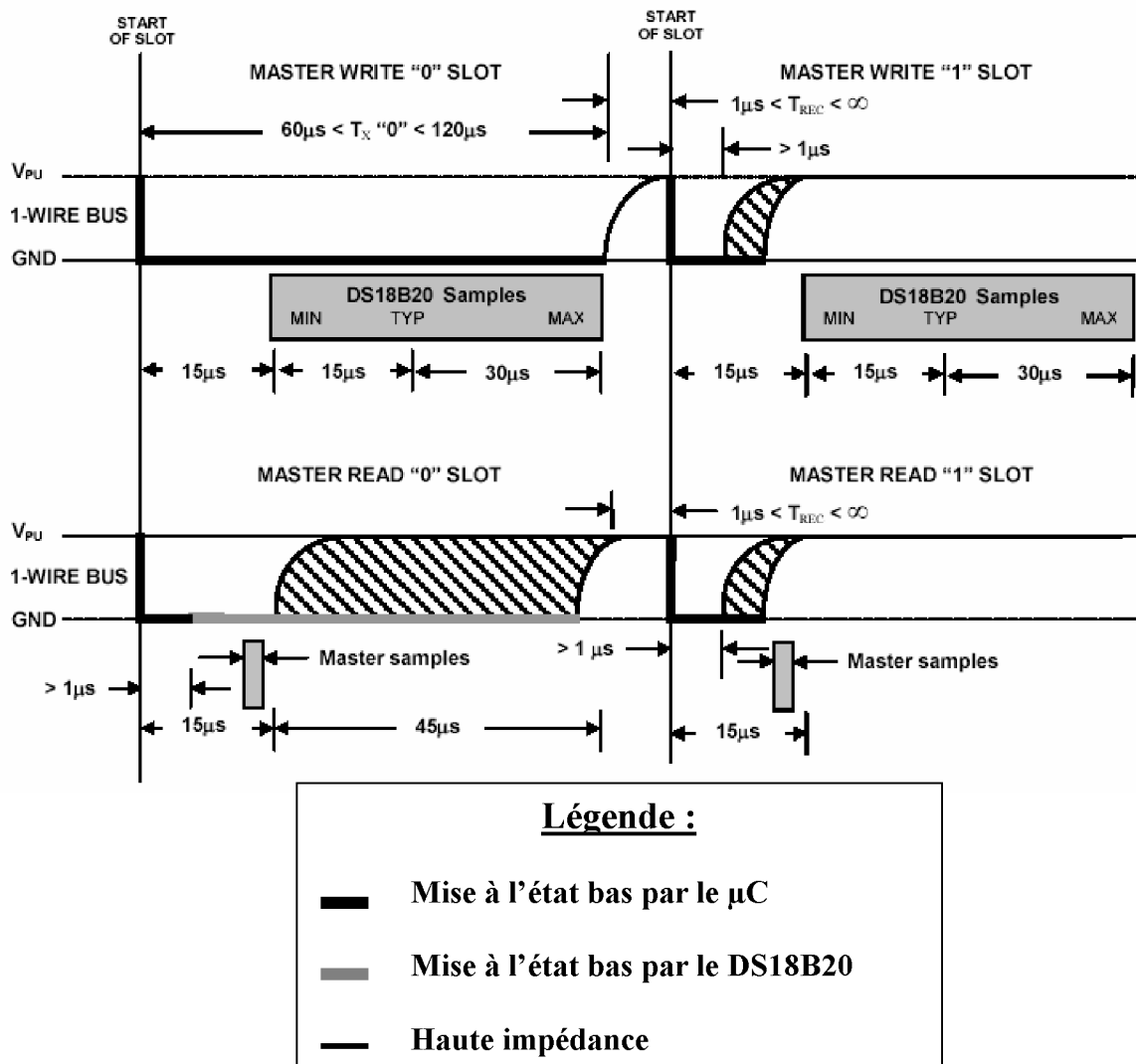
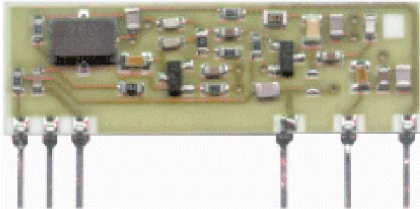


Figure 11 : Oscillogramme représentant la manière d'écrire ou de lire des données bit à bit

L'oscillogramme est explicite mais détaillons quand même par exemple comment lire un bit. Un niveau bas est appliqué pendant 1 à 2 µs, le µcontrôleur met sa ligne en entrée et ensuite au bout de 15µs la lecture de la ligne va indiquer la valeur du bit à lire. Si le DS veut indiquer 0, il maintient sa ligne à 0 et s'il veut indiquer 1 il relâche la ligne.

b) L'émetteur TELECONTROLLI RT8-868

L'émetteur qu'on a utilisé est le TELECONTROLLI RT8-868. C'est un émetteur numérique.



Description générale

Le "RT8-868" est un émetteur stabilisé par résonateur à onde de surface (SAW) bénéficiant d'une technologie sur substrat céramique à double étage transistorisé. Sa sortie pour antenne extérieure, son filtre atténuant l'émission d'harmonique et son implantation de type "SIL" le prête à de multiples cas d'applications dans le cadre de réalisation de télécommandes ou de mini émetteurs divers.

Caractéristiques électriques

Désignations	Min.	Typ.	Max.	Unité
Tension d'alimentation	4,5	-	13	Volt
Débit maximum	-	-	9600	Bit/s
Fréquence d'émission	-	868,35	-	Mhz
Puissance émise dans 50 Ohm (Vcc=5V)	-	7	10	dBm

Figure 12 : caractéristiques électriques

Description des broches				Dimensions du module	
2	Entrée	11	GND		
3	AUX	13	Sortie Antenne		
4	GND	15	VCC		

Figure 13 : Description des broches

Ce module va transmettre les différentes mesures de température par liaison HF sur une distance de quelques dizaines de mètre au moins.

La puissance émise est de 7 dBm à Vcc =5V avec une antenne adaptée de 50 Ohm. $P(\text{dBm})=10*\log(P(\text{mW})) \rightarrow P=5 \text{ mW}$.

Cependant notre antenne ne sera pas adaptée mais l'alimentation sous 9 V compensera un peu cette perte de puissance. Le débit de donnée maximum est de 9600 Bits/s ce qui est largement suffisant dans notre cas.

Son utilisation est très simple bien que la conception d'un tel circuit soit difficile. En effet, lorsque le système le gérant veut émettre une donnée, il transmet sa séquence de bits à travers la patte 2, et l'émetteur se charge de la modulation discrète et de l'envoi de l'information par transmission hertzienne au travers de l'antenne qui est connecté à la patte 13 et cela à la fréquence de 868 MHz.

Cet émetteur est doté d'une mise en veille automatique, lorsque le signal d'entrée ne varie plus, le circuit ne consomme plus. Il n'y a donc pas besoin de commuter l'alimentation à l'aide d'un transistor.

c) le microcontrôleur PIC 16F870

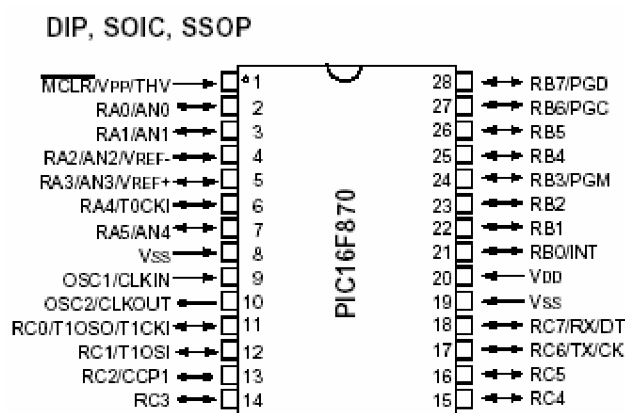


Figure 14 : brochage du PIC16F870

Le 16F870 de Microchip est un microcontrôleur 8 bits CMOS FLASH d'architecture RISC ne disposant « que » de 35 instructions. Sa prise en main est donc relativement rapide.

Il contient :

- 2 K * 14 bits de mémoire programme FLASH.
 - 128 octets de mémoire de donnée (RAM).
 - 64 octets de mémoire de donnée (EEPROM).
 - 5 convertisseurs A/N 10 bits.
 - 3 ports d'E/S.
 - Entrée Sortie USART.
 - 3 Timers 8/16 bits (Timer0, Timer1 et Timer2)
 - 10 sources d'interruptions (matérielles et logicielles).
- mode SLEEP
- Précédemment, nous avons émis l'information que notre PIC utiliserait le mode SLEEP ou « power down » afin de consommer le moins possible pendant les instants pendant lequel il est « inactif », c'est à dire entre 2 traitements de mesure :
- prélèvement de la mesure de température.
 - stockage dans sa mémoire interne EEPROM.
 - transmission hertzienne via le RT8.

Donc une fois dans ce mode, le PIC est placé en sommeil et cesse d'exécuter son programme. Mais le passage en mode SLEEP n'a de réel intérêt que s'il est possible d'en sortir. Le PIC 16F870 ne réagit dans ce mode qu'aux évènements suivants :

- application d'un niveau 0 sur le MCLR, provoquant ainsi un reset.
- écoulement du temps ou débordement du TIMER1.
- apparition d'une interruption sur la patte RB0/INT du port B.

C'est sur ces évènements que nous devons jouer pour réveiller le PIC et ainsi permettre une consommation minimale et une longue autonomie à notre circuit émetteur.

B) Description et fonctionnement du circuit émetteur

a) Schémas synoptiques détaillés

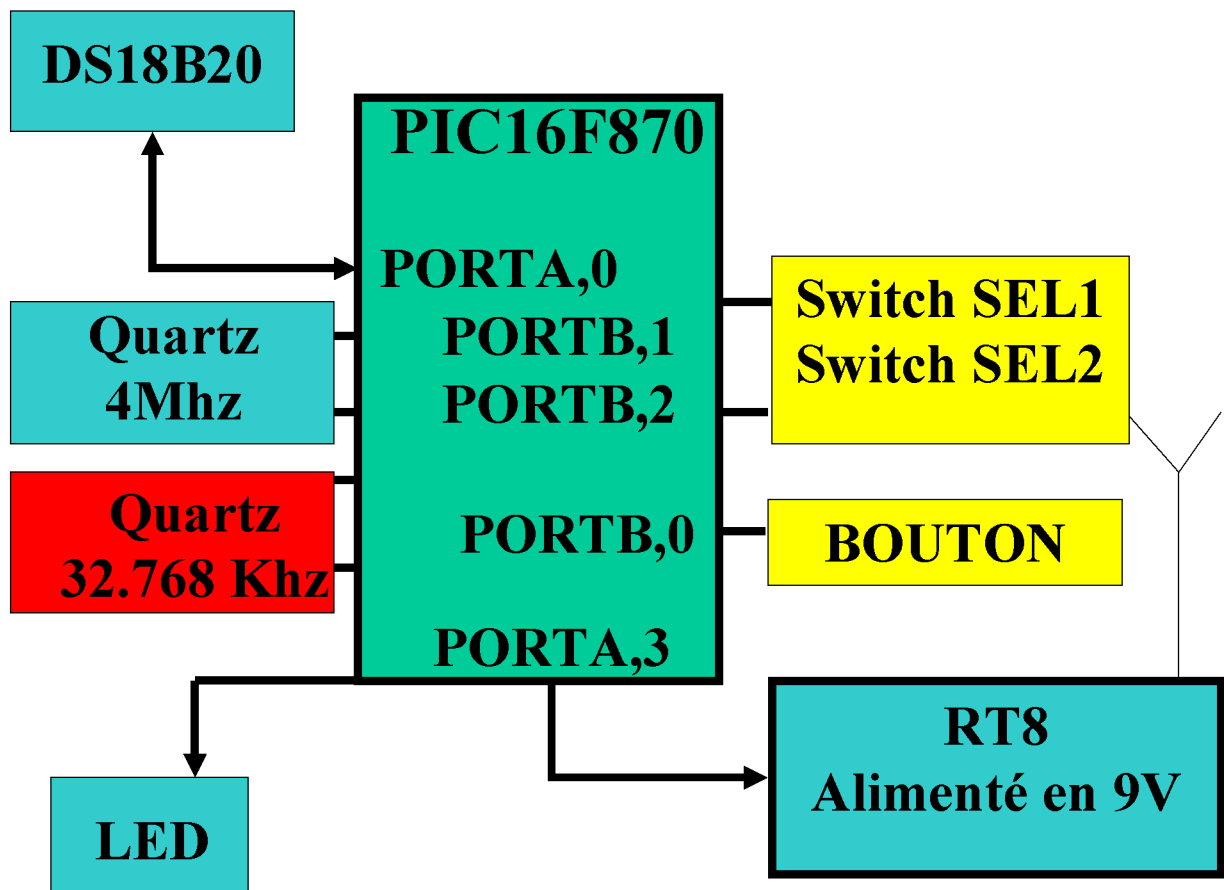


Figure 15 : Schéma synoptique du circuit émetteur

Nous voyons ici les différents composants interconnectés avec le PIC16F870. Nous remarquons la présence du quartz horloger de 32.768Khz utilisé pour faire fonctionner le Timer 1 pendant le mode SLEEP et l'arrêt de l'horloge principale.

Une LED indique à l'utilisateur par un allumage bref la prise et l'envoi d'une mesure.

b) Schéma électrique « Protel »

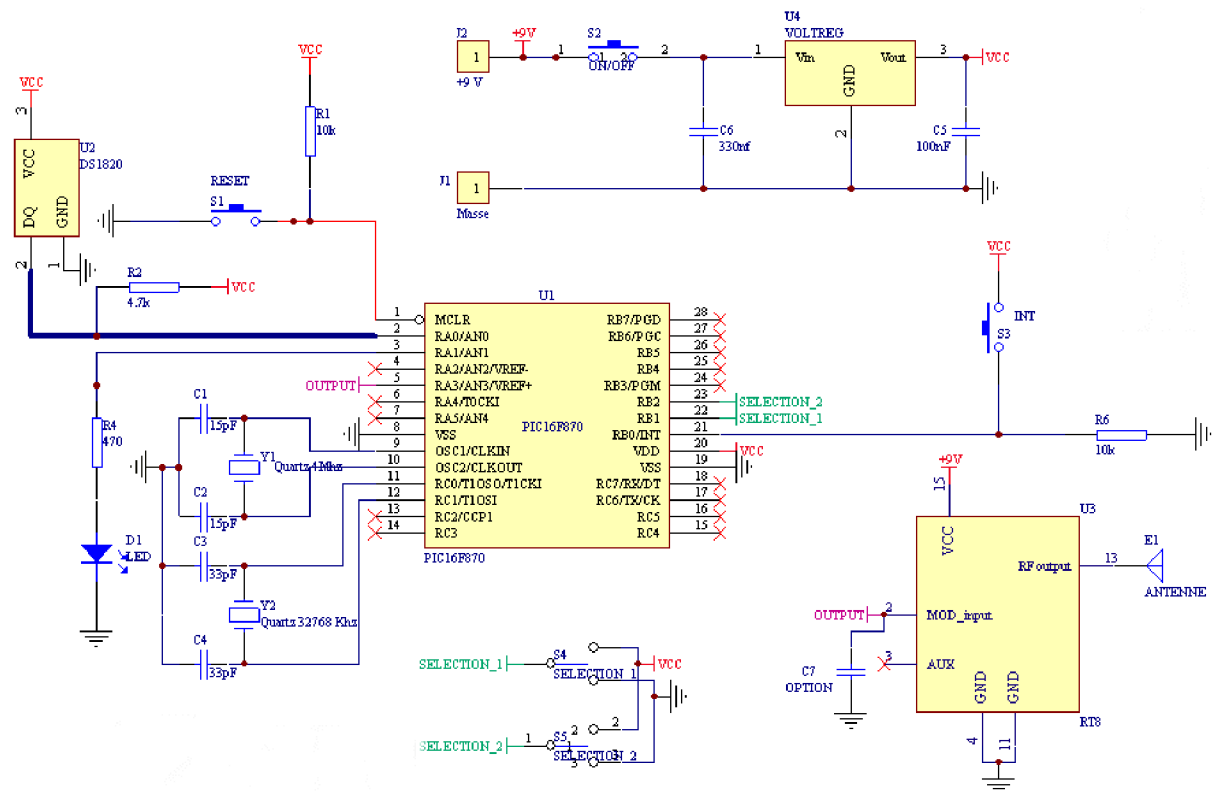


Figure 16 : Schéma Protel de l'émetteur

Nous remarquons ici la présence du régulateur MC7805, couplé à ces 2 capacités polyesters, nécessaire pour alimenter en 5 V le reste du montage.

c) Descriptions des différentes interconnexions

- Alimentation

Une pile de 9 V est utilisée pour alimenter le montage. En réalité, seul l'émetteur RT8 utilise directement cette tension pour un besoin de puissance d'émission et le reste du montage est alimenté en 5 V grâce à un régulateur de tension le MC7805.

- communication DS18B20 et PIC 16F870

La ligne de communication ne nécessite qu'une ligne comme énoncé dans la description du DS18B20 qui a 2 niveaux possibles : niveau logique 0 et état Haute Impédance. Le protocole de communication spécifique au DS18B20, nécessite une ligne bidirectionnelle (commutation écouteur1-parleur2 et parleur1-écouteur2). Il faudra pour écouter le capteur configurer la patte utilisée en entrée, ainsi le DS18B20 sera parleur. Le transfert de donnée est réalisé donc par communication série, bit par bit. Arbitrairement, nous avons pris la patte RA0.