

Wstęp

Pomysł opracowania i budowy beaconu MGM zaszczerpił nam (tzn ekipie SP5PIP) Zdzisław SP6LB podczas spotkania na Stogu Izerskim w sierpniu 2004. Etap konstrukcji poprzedziły dyskusje w gronie klubowym na temat filozofii pracy beaconu oraz koncepcji wykonania, która pozwalałaby uzyskać elastyczną konstrukcję unikając jednocześnie zbyt skomplikowanych i drogich elementów, które mogłyby uczynić beacon podatnym na kradzież lub uszkodzenie.

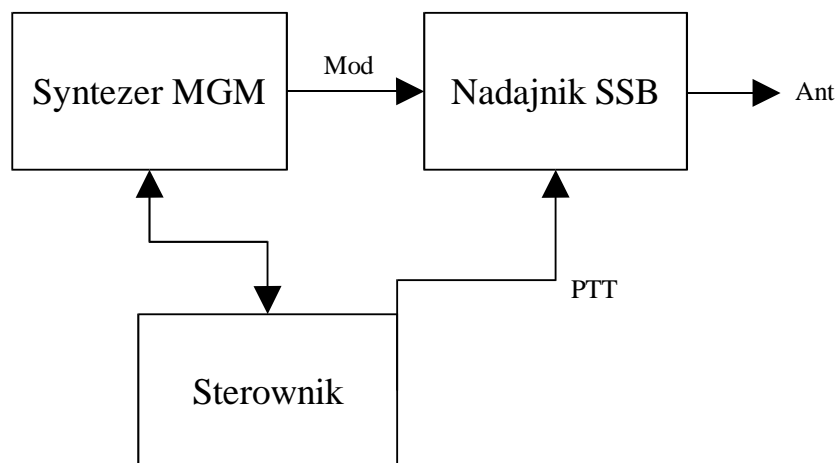
Przedstawione poniżej rozwiązanie należy traktować jako prototyp powstały w celu weryfikacji pewnej koncepcji rozwiązania problemu technicznego, jakim jest synteza sygnału MGM poza komputerem. Jest to zatem punkt wyjściowy do opracowania w pełni funkcjonalnego układu beaconu posiadającego możliwość kontroli czasu transmisji wraz z synchronizacją zewnętrznym wzorcem czasu. Rozpatrując różne dostępne wzorce czasu odrzuciliśmy ideę synchronizacji beaconów poprzez internet. Takie rozwiązanie wymagałoby dostępu do sieci we wszystkich punktach instalacji beaconów. Dodatkowo wymagałoby implementacji protokołów sieciowych w mikrokontrolerze beacona, co znacznie podniosłoby komplikację konstrukcji. Z dwóch powszechnie dostępnych źródeł wzorca czasu tj., DCF77 i GPS zdecydowaliśmy się skupić na tym pierwszym. Decyzja została podyktowana względami ekonomicznymi. Odbiornik DCF77 jest o rząd wielkości tańszy niż najprostsze urządzenia GPS.

W przedstawionym poniżej rozwiązaniu nie zaimplementowano jeszcze funkcji synchronizacji zewnętrznym wzorcem czasu. Ten problem pozostawiona na następny etap rozwijania konstrukcji beaconu MGM.

Koncepcja

W przypadku beacona MGM można wyróżnić trzy funkcjonalne elementy:

1. Sterownik
2. Źródło sygnału modulującego – syntezer MGM
3. Analogowy tor nadajnika radiowego.



Funkcja sterownika polega na wybieraniu poszczególnych komunikatów MGM, sterowaniu nadajnika (PTT) oraz synchronizacji okresów nadawania w przypadku pracy beaconu w sieci.

Rozważając możliwe rozwiązania sterownika oraz syntezeru MGM odrzucono koncepcję zastosowania do tego celu komputera ze względu na koszty oraz groźbę kradzieży po zainstalowaniu beacona w docelowym miejscu.

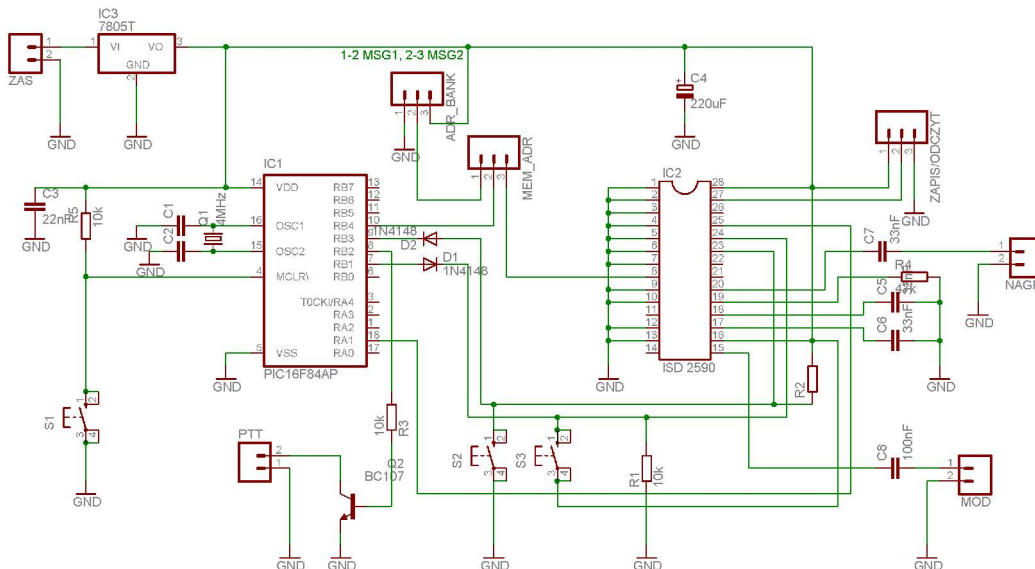
Konstrukcję syntezy postanowiono oprzeć na układzie ISD2590 pozwalającym na zapis i odtwarzanie sygnałów akustycznych o czasie trwania do 90s.

W celu zbadania przydatności układu ISD2590 do tego zadania zbudowano prototyp beacons wykonując sterownik w oparciu o popularny mikrokontroler PIC16F84 oraz TRX IC-290.

W docelowych układzie planuje się zastąpienie TRX'a dedykowanym nadajnikiem SSB oraz ewentualne wzbogacenie funkcji sterownika o możliwość synchronizacji w oparciu o sygnał wzorca czasu DCF.

Opis układu

Na rysunku przedstawiono schemat ideowy sterownika i syntezy sygnału MGM.



Układ PIC16F84 steruje pracą beacons w następującej sekwencji:

1. Wybiera adres wiadomości zapisanej w układzie ISD2590 (8 nóżka ISD)
2. Resetuje układ ISD2590.
3. Włącza układ sterowania PTT (tranzystor BC107)
4. Inicjuje odtwarzanie zapisanego sygnału
5. Po otrzymaniu sygnału END_MSG (25 nóżka ISD) wyłącza PTT i zaczyna odliczać czas przerwy.
6. Powtarza cały cykl przełączając adres wiadomości.

W układzie umieszczono dodatkowo przełączniki S1, S2, S3 pozwalające na ingerencje w pracę beacons niezależnie od mikrokontrolera oraz ułatwiające programowanie sygnałów nadawanych przez beacons.

Przełącznik S1 resetuje układ mikrokontrolera restartując równocześnie program sterujący beacons.

Przełącznik S3 resetuje układ ISD2590 natomiast przełącznik S2 inicjuje odtworzenie sygnału w stanie **Odtwarzanie** lub nagrywanie w stanie **Nagrywanie**. Tryby **Odtwarzanie (1-2)** i **Nagrywanie (2-3)** przełączane są zworą ZAPIS/ODCZYT.

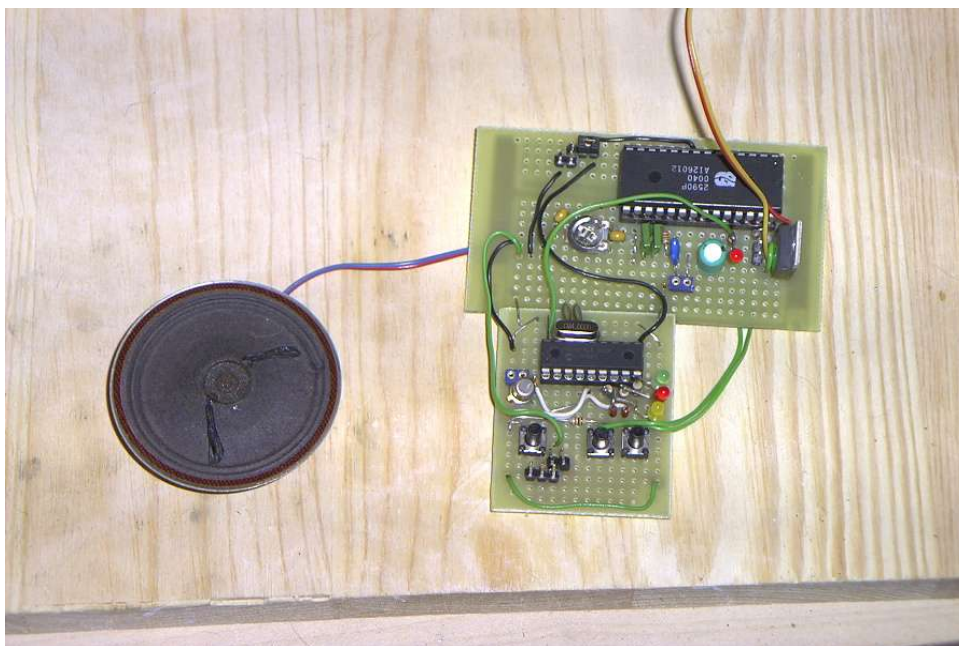
W układzie można zastosować dowolne diody przełączające. Typ diod nie jest krytyczny.

Zapisanie treści sygnałów odbywa się według następującej procedury.

1. Zablokować mikroprocesor PIC16F84. Można to zrobić wyjmując układ z podstawki lub na stałe zwierając przełącznik S1.
2. Przełączyć zworę ZAPIS/ODCZYT w pozycję Nagrywanie (2-3)
3. Przełączyć zworę MEM_ADR w pozycję 1-2
4. Zworą ADR_BANK wybrać wiadomość do zapisu
5. Do gniazda NAGR podłączyć źródło sygnału MGM (w moim wypadku jest to karta dźwiękowa komputera z uruchomionym programem odpowiednim do danej emisji)
6. Zresetować układ IDS2590 przyciskając przełącznik S3
7. Uruchomić źródło sygnału wciskając równocześnie przełącznik S2
8. Po zakończeniu sygnału zwolnić przełącznik S2, czas nagrania nie może przekraczać 45sekund
9. W przypadku nagrywania drugiego komunikatu należy przełączyć zworę ADR_BANK i powtórzyć czynności 6-8.
10. Po zakończeniu nagrywania należy przełączyć zworę ZAPIS/ODCZYT w pozycję odczyt. Ręczne odtworzenie nagranych komunikatów można uzyskać wciskając na chwilę przełączniki S3 a potem S2.
11. Przełączyć zworę MEM_ADR w pozycję 2-3 i odblokować (włożyć w podstawkę) mikroprocesor PIC16F84.

Kolejność czynności 1-5 nie jest krytyczna i może być zmieniana.

Podczas nagrywania istotną sprawą jest prawidłowy dobór poziomu sygnału źródłowego. Zbyt niski poziom spowoduje wzrost tła szumowego w sygnale beacons natomiast zbyt wysoki skutkuje zniekształceniami i wystąpieniem niepożądanych elementów w widmie sygnału wyjściowego. W oryginalnym układzie aplikacyjnym ISD2590 występuje obwód ARW. W przypadku zastosowania ISD2590 w beaconie zrezygnowano z niego jako zbędnej komplikacji układu. Zakładam, że nagrywanie komunikatów będzie zabiegiem sporadycznym, jeśli nie jednorazowym, więc specjalne wspieranie tej czynności jest zbędne.



Prototyp beacons MGM zmontowany na płytce uniwersalnej. Podczas pierwszych prób do wyjścia podłączyłem głośnik żeby móc "na ucho" ocenić jakość sygnału MGM.

Listing programu dla PIC16F84

```
LIST P=16F84 ;Ustawienie typu mikroprocesora

__CONFIG 0x3FF1 ;XT, NO WDT, PWR-UP, NO CP

STATUS EQU 0x03
RA EQU 0x05
RB EQU 0x06
PC EQU 0x02
FSR EQU 0x04

W EQU 0x00
F EQU 0x01

;***** USER DEF'S *****

;COUNT EQU 0x16
;TEMP EQU 0x17
COUNT1 EQU 0x18
COUNT2 EQU 0x19
COUNT3 EQU 0x1A
COUNT4 EQU 0x1B
COUNT5 EQU 0x28
;FLAGS EQU 0x29

ORG 0x00
GOTO START

;***** DELAY LOOP *****
;
;petla opóźniająca 100ms * W
DECI MOVWF COUNT4
NXT6 CALL MS100
      DECFSZ COUNT4,1
      GOTO NXT6
      RETLW 0
;*****

;***** DELAY LOOP 100MS *****
MS100 MOVLW 0x2D ;100 MS DELAY LOOP
      MOVWF COUNT1 ;EXACT TIMING .1 SEC
L1 MOVLW 0x0C ;INCLUDING CALL AND RETURN
   MOVWF COUNT2 ;4 MHZ XTAL
L2 MOVLW 0x3C
   MOVWF COUNT3
L3 DECFSZ COUNT3,1
```

```

GOTO L3
DECFSZ    COUNT2,1
GOTO L2
DECFSZ    COUNT1,1
GOTO L1
MOVLW    0x96
MOVWF    COUNT1
LP99 DECFSZ    COUNT1,1
GOTO LP99
NOP
NOP
RETLW    0

```

```

;*****

```

```

;*****

```

```

; Resetuje Układ ISD2590 24pin - krótki impuls High

```

```

RESET_ISD  BSF  RB, 1
            CALL MS100
            BCF  RB, 1
            RETLW 0

```

```

;*****

```

```

;*****

```

```

; Startuje odtwarzanie wiadomości z ISD - krótki impuls Low

```

```

START_ISD  BCF  RB, 3
            CALL MS100
            BSF  RB, 3
            RETLW 0

```

```

;*****

```

```

;*****

```

```

;*****

```

```

;***** MAIN PROG *****

```

```

;Inicjacja portów

```

```

START BCF      3, 6
      BSF      3, 5           ;SWITCH TO F REGISTER PAGE 1
      MOVLW    0x0F
      MOVWF    RA           ;RA 0 TO 4 INPUTS,
      MOVWF    0x85
      MOVLW    0x00
      MOVWF    RB           ;RB 0 TO 7 OUTPUTS
      MOVWF    0x86

      MOVLW    0xB7
      MOVWF    0X01         ;PRESCALER /256, RB PULL UP DIS.
      BCF      3, 5         ;SWITCH TO F REGISTER PAGE 0

```

```
CLRF RA
CLRF RB
```

```
BCF RB,1 ;ustawienie 0 wejściu Reset ISD - pin 24 ISD2590
BCF RB,2 ;ustawienie 0 na wyjściu PTT
BSF RB,3 ;ustawienie 1 na wejściu Start ISD - pin 23 ISD2590
```

```
; Startujemy z główną pętlę
```

```
MAIN_LOOP BCF RB, 4 ; Ustawienie adresu pierwszej wiadomości
CALL RESET_ISD
BSF RB, 2 ;Włączenie PTT
CALL START_ISD
```

```
END_MSG1 BTFSC RA, 1
GOTO END_MSG1
```

```
BCF R B, 2 ;Wyłączenie PTT
```

```
MOVLW 0xFF ;Opóźnienie 25,5 sec.
CALL DECI
```

```
BSF RB, 4 ;Ustawienie adresu drugiej wiadomości
CALL RESET_ISD
BSF RB, 2 ;Włączenie PTT
CALL START_ISD
```

```
END_MSG2 BTFSC RA, 1
GOTO END_MSG2
```

```
BCF RB, 2 ;Wyłączenie PTT
MOVLW 0xFF ;Opóźnienie 25,5 sec.
CALL DECI
```

```
GOTO MAIN_LOOP ;
```

```
END
```