

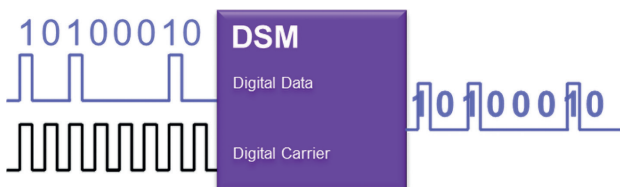


zapisanej w rejestrze ADRPT, to wartość z rejestru ADFLT staje się średnią wartością sygnału wejściowego za ADCNT próbek. Średnia wartość jest poprawnie obliczana dla ADRPT równe  $ADRPT=2^{ADCNT}$ .

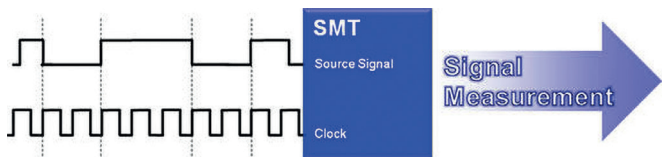
- **FILTROWANIE DOLNOPRZEPUSTOWE.** Charakterystyka tłumienia filtra zależy oczywiście od częstotliwości próbkowania. Maksymalna częstotliwość sygnału próbkowanego powinna być równa połowie częstotliwości próbkowania (zgodnie z twierdzeniem Nyquista). Rząd filtru jest programowany w zakresie od 1 do 6 (rejestr ADCRS). Na **rysunku 2** pokazano przykład charakterystyk filtrów różnych rzędów dla próbkowania z częstotliwością 900 Hz.

Oprócz przetwornika zabudowano bardzo ciekawe, ale bardziej już znane interfejsy typu Core Independent:

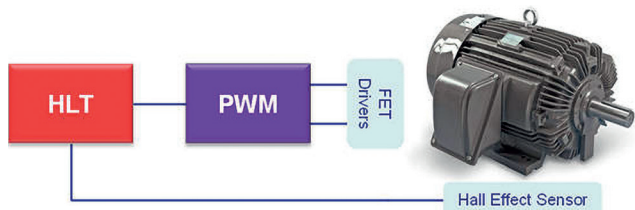
- **MODUŁ PWM** o rozdzielczości 10 bitów z wyrównaniem do zbocza (edge aligned output).
- **GENERATOR SYGNAŁU KOMPLEMENTARNEGO CWIG** z funkcją dodawania czasów *rising edge deadband* i *falling edge deadband* (**rysunek 4**). Źródłem sygnału jest wyjście generatora PWM.
- Cyfrowo sterowany **OSCYLATOR NCO** – precyzyjny oscylator z dokładną regulacją częstotliwości.
- **MODUŁ DSM** (Digital Signal Modulator) modulujący sygnał nośnej (carrier) sygnałem danych – przykład działania DSM pokazano na **rysunku 6**.
- **MODUŁ SIGNAL MEASUREMENT TIMER** (**rysunek 7**) to 24-bitowy licznik z zaawansowaną logiką taktowania i bramkowania sygnału mierzonego. Może być skonfigurowany do mierzenia różnych sygnałów cyfrowych, takich jak długość impulsów, częstotliwość i wypełnienie PWM, czas pomiędzy zboczami dwóch sygnałów.
- **MODUŁ HARDWARE LIMIT TIMER HLT** (**rysunek 8**) jest przeznaczony do wykrywania periodyczności zdarzeń na przykład z czujnika wału obrotów wału silnika. Może wykrywać przeciążenia (spadek obrotów wału poniżej określonej wartości)



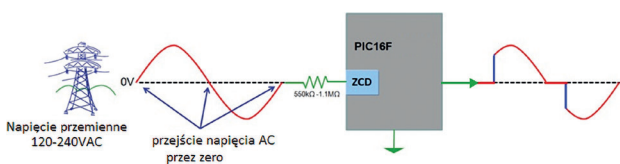
Rysunek 6. Moduł DSM



Rysunek 7. Moduł SMT



Rysunek 8. Przykład wykorzystania modułu HLT



Rysunek 9. Przykład zastosowania modułu Zero Cross Detect

lub zatrzymanie wału silnika. Wykrycie stanów awaryjnych może być sygnałem dla generatora PWM, który zatrzyma sterowanie silnikiem.

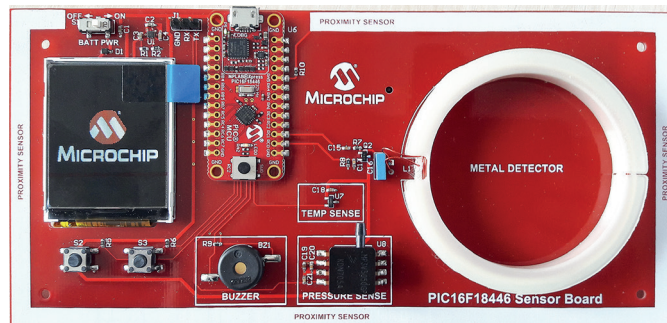
- **MODUŁ ZERO CROSS DETECT** (**rysunek 9**). Wykrywa przejście sygnału przemiennego przez zero (potencjał masy). Może być użyty do sterowania triakiem z ograniczonymi zakłóceniami EMI (sterowanie w zerze). Może również służyć do pomiaru okresu sygnału AC.

Standardowo wbudowano też mechanizm PPS (Peripheral Pin Select) pozwalający na przypisywanie sygnałów układów peryferyjnych do wybranych wyprowadzeń mikrokontrolera, rozbudowany moduł watchdoga WWD oraz tryby oszczędzania energii IDLE oraz DOZE. Ciekawe jest wprowadzenie mechanizmu MAP Memory Acces Partition zapewniającego ochronę danych użytkownika i wspierającego użycie bootloadera.

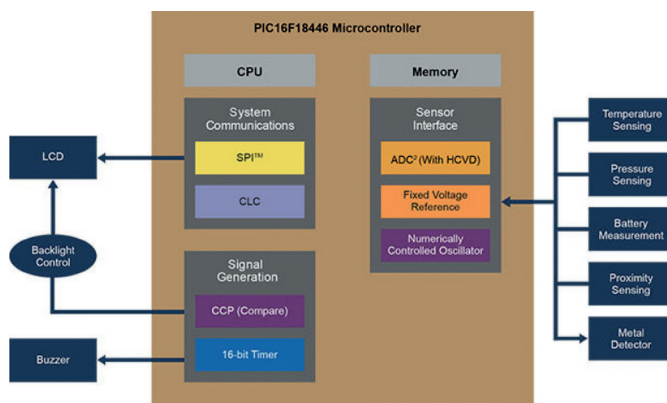
Jak widać, wyposażenie tego „małego” mikrokontrolera jest całkiem bogate. Zobaczmy teraz jak można je wykorzystać w nietypowych i standardowych zastosowaniach. Taką możliwość daje firmowy moduł ewaluacyjny PIC16F18446 Sensor Board – pokazano go na **rysunku 10**.

Mikrokontroler z układem programatora/debuggera zamontowano na niewielkiej płytce MPLAB Xpress PIC16F18446, przylutowanej do dużej płytki bazowej. Na płytce bazowej zamontowano układy: detektora metalu, detektora zbliżenia, czujnika ciśnienia atmosferycznego, czujnika temperatury otoczenia, graficzny wyświetlacz LCD ze stabilizatorem +2,8 V, dwa przyciski S2 i S3, buzzer oraz wyłącznik zasilania. Na drugiej stronie płytki zamontowano dwa gniazda dla baterii alkalicznych 1,5 V typu AA. Schemat ideowy modułu Sensor Board pokazano na **rysunku 11**.

Niespotykanym rozwiązaniem jest zastosowanie w układzie z „małym” mikrokontrolerem wyświetlacza graficznego zwykle kojarzonego z dużymi wymaganiami sprzętowymi. Wyświetlacz ma rozdzielczość 128×160 pikseli i przekątną 1,8 cala. Do komunikacji z wbudowanym sterownikiem IL9163C wykorzystuje się magistralę SPI. Napięcie zasilania może mieć wartość z zakresu 2,5...3,3 V i typowo wynosi 2,8 V. W module użyto stabilizatora +2,8 V oraz układów dopasowujących sygnały z poziomów logicznych portów mikrokontrolera do poziomów logicznych akceptowanych przez sterownik wyświetlacza zasilanego



Rysunek 10. Moduł Sensor Board

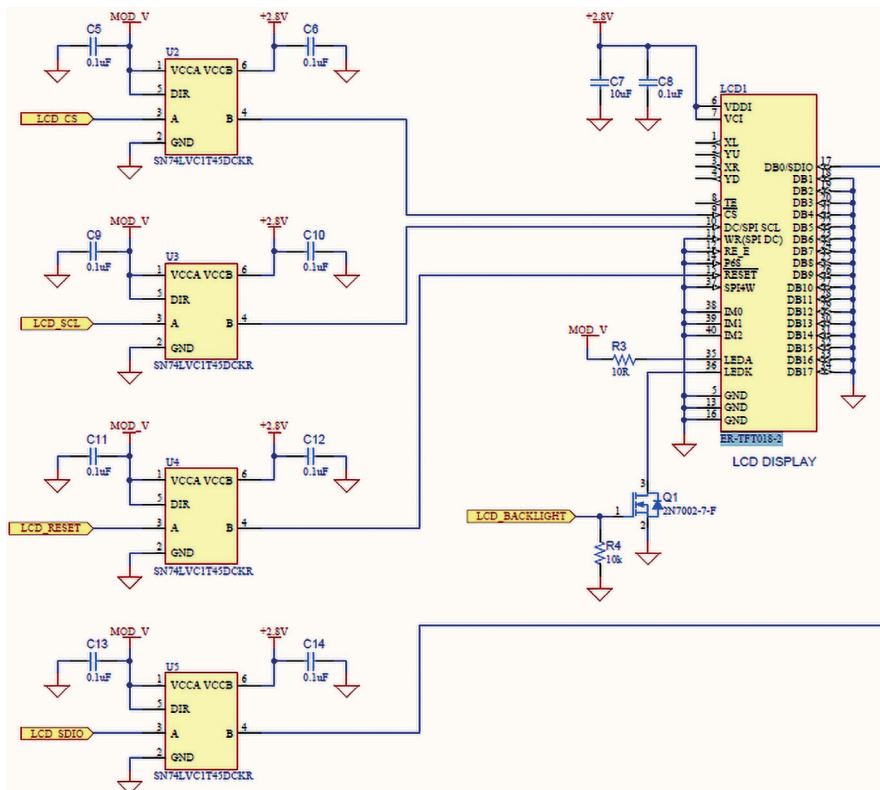


Rysunek 11. Schemat blokowy modułu Sensor Board

napięciem +2,8 V. Dodatkowo, zastosowano układ sterowania podświetleniem z tranzystorem FET. Schemat układu wyświetlacza pokazano na rysunku 12.

Program demonstracyjny dostarczany z modulem jest zorganizowany w formie siedmiu funkcji (ekranów) przełączanych przyciskami S2 (poprzedni ekran) i S3 (następny ekran). Mamy do dyspozycji 7 funkcji:

1. Pomiar napięcia baterii zasilających moduł.
2. Pomiar temperatury otoczenia w stopniach Celsjusza z rozdzielczością 10 i 12 bitów.
3. Pomiar ciśnienia atmosferycznego z rozdzielczością 10 i 12 bitów.
4. Wykrywanie i pomiar zbliżenia. Kiedy moduł jest zasilany z baterii brak aktywności wykrywania zbliżenia powoduje wygaszenie podświetlania wyświetlacza.
5. Wykrywanie przedmiotów metalowych ferromagnetyków.
6. Generowanie dźwięków z częstotliwością zależną od wykrytej odległości zbliżenia.
7. Generowanie dźwięków z częstotliwością zależną od wykrytej odległości przedmiotu metalowego do detektora



Rysunek 12. Wyświetlacz LCD z układami dopasowującymi poziomy logiczne do +2,8 V

### Pomiar napięcia baterii

Pomiar napięcia baterii wykonuje wbudowany przetwornik A/C. Jest on używany we wszystkich funkcjach: pomiaru napięcia baterii, temperatury, ciśnienia, w czujniku zbliżenia i czujniku metalu. Na rysunku 13 pokazano konfigurację tego modułu ADCC przez wtyczkę MCC pakietu MPLAB X IDE. Napięcie +3 V z dwóch połączonych szeregowo baterii jest podawane przez diodę Shottky na wyłącznik zasilania. Dioda zabezpiecza układ zasilania bateryjnego przed uszkodzeniem podczas zasilania zewnętrznego (przez złącze USB). Mamy mierzyć napięcie zasilania, więc wystarczy ustawić napięcie referencyjne: ADNREF= VSS (GND) i ADPREF=VDD. Pomiar napięcia wykonuje procedura `readBattery()` pokazana na listingu 1. Mierzone jest napięcie stałe z modułu FVR Buffer, podawane na wejście przetwornika, a zmienia się wartość napięcia referencyjnego – w naszym wypadku napięcia zasilania VDD z baterii.

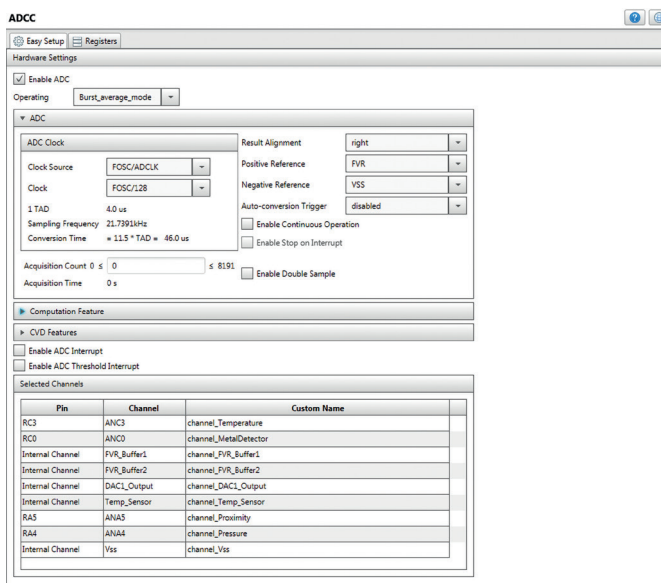
### Pomiar temperatury zewnętrznej

Do pomiaru użyto sensora MCP9701. Jest to typowy układ mierzący temperaturę z zakresu od 0 do +70°C z maksymalną niepewnością ±2°C. Sygnałem wyjściowym jest napięcie zmienne ze współczynnikiem 19,5 mV/°C. Napięcie Vout jest podawane na wyprowadzenie RC3 (ANC3) mikrokontrolera. Napięcie początkowe dla temperatury 0°C wynosi 400 mV.

W temperaturze pokojowej +20°C mamy napięcie 390 mV (20×19,5 mV) + 400 mV=790 mV. Napięcie referencyjne dla ADCC jest uzyskiwane z modułu programowanego napięcia referencyjnego FVR\_Buffer1 – ma ono wartość 1,024 V. To określa maksymalną mierzoną temperaturę na poziomie 1024 mV-400 mV= 624 mV, a więc 624 mV/19,5 mV=32. Układ pomiaru temperatury pokazano na rysunku 15, a procedura pomiaru na listingu 2.

### Pomiar ciśnienia atmosferycznego

Czujnikiem ciśnienia atmosferycznego jest układ MP3V5050VC6U, lub MP3V5050GP (rysunek 16). Wykonano



Rysunek 13. Konfiguracja przetwornika ADCC

Listing 1. Procedura pomiaru napięcia baterii

```

void readBattery()
{
    uint32_t res;
    uint8_t i;

    ADREF = 0x00; // ADNREF VSS; ADPREF VDD - ustawienie napięć referencyjnych
    ADCC_GetSingleConversion(channel_FVR_Buffer1); wyzwolenie konwersji
    res = ADCC_GetAccumulatorValue() >> 6;
    if(bami==-1) {
        bamAcc = 0;
        for(i=0; i<BATT_ADC_MEM_SIZE; i++) {
            battAdcMem[i] = res;
            bamAcc += res;
        }
        bami = 0;
    } else {
        bamAcc -= battAdcMem[bami];
        battAdcMem[bami] = res;
        bamAcc += battAdcMem[bami];
        bami++;
        bami = bami % BATT_ADC_MEM_SIZE;
    }
    res = bamAcc/BATT_ADC_MEM_SIZE;
    batteryMillivolts = 4194304/res;
}
    
```



**Listing 2. Procedura pomiaru temperatury**

```
int32_t temp10bit, temp12bit;
int32_t temp10bitOffset, temp12bitOffset;

void readTemperature() {
    uint32_t res;
    uint16_t i;

    ADREF = 0x03; // ADNREF VSS; ADPREF FVR
    res = 0;
    for(i=0; i<32; i++) {
        ADCC_GetSingleConversion(channel_Temperature);
        res += (ADCC_GetAccumulatorValue() >> 6);
    }
    res = res >> 5;
    temp10bit = (res & 0xFFFFFFF) - 1600;
    temp10bit *= 100;
    temp10bit /= 78;
    temp12bit = res
                - 1600;
    temp12bit *= 100;
    temp12bit /= 78;
}
}
```

**Listing 3. Procedura pomiaru ciśnienia atmosferycznego**

```
void readPressure() {
    uint32_t res;
    uint32_t v;

    ADREF = 0x03; // ADNREF VSS; ADPREF FVR
    ADCC_GetSingleConversion(channel_Pressure);
    res = ADCC_GetAccumulatorValue() >> 6;
    v = res;
    v += 16602;
    v *= 596;
    v /= 100;
    pressure12bit = v;
    v = res & 0xFFFFFFF; // 12-bit to 10-bit
    v += 16602;
    v *= 596;
    v /= 100;
    pressure10bit = v;
}
}
```

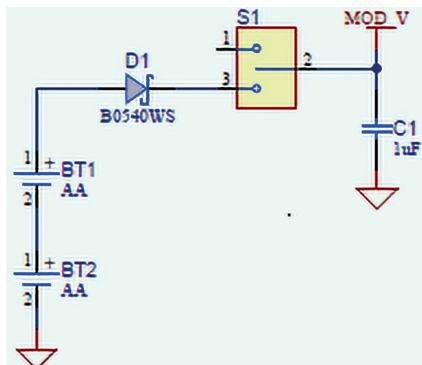
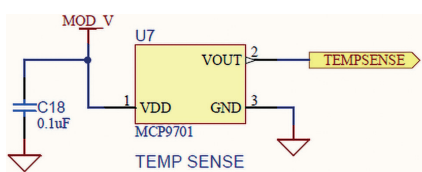
go w technologii MEMS. Mierzy różnicę ciśnienia w zakresie od 0 do 50 kPa od poziomu 100 kPa. Rozdzielczość pomiaru wynosi 54 mV/kPa, więc dla pełnego zakresu na wyjściu Vout mamy napięcie  $50 \times 54 \text{ mV} = 2700 \text{ mV}$ . Układ musi być zasilany napięciem co najmniej +2,7 V – typowo 3,0 V. My mierzymy ciśnienie atmosferyczne o wartości 100 kPa (1000 hPa)+50 kPa. Na **listingu 3** pokazano procedurę pomiaru napięcia z rozdzielczością 10- i 12-bitową z czujnika i przeliczenie jej na wartość ciśnienia w paskalach. Dla wyświetlanego ciśnienia 1023 hPa na wyjściu czujnika występuje napięcie ok. 145 mV.

**Detektor metalu**

Pomiary temperatury lub ciśnienia za pomocą dedykowanych czujników, to standardowe wykorzystanie modułu przetwornika A/C. Na płytce zamontowano ciekawy układ do wykrywania zbliżenia przedmiotów wykonanych z ferromagnetyków (np. stali). Takiego czujnika można użyć na przykład do wykrywania zamknięcia lub otwarcia metalowych drzwi.

Głównym elementem detektora jest spora cewka powietrzna o indukcyjności 160  $\mu\text{H}$  (**rysunek 17**). Na **rysunku 18** pokazano schemat elektryczny układu detektora.

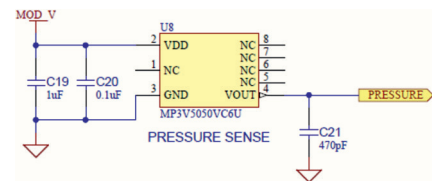
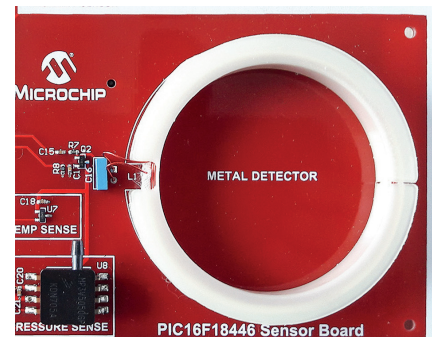
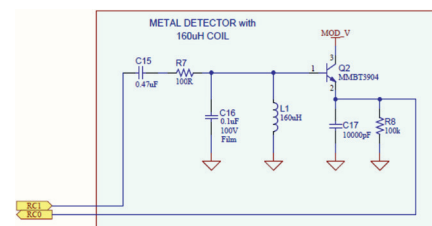
Zasada działania detektora jest następująca – po uruchomieniu procesu detekcji jest wykonywana jednorazowo procedura skanowania. Na wejście układu z wyprowadzenia RC1 jest podawany przebieg prostokątny generowany przez moduł NCO. Częstotliwość tego przebiegu

**Rysunek 14. Dołączenie zasilania baterijnego****Rysunek 15. Układ pomiaru temperatury**

zmienia się w zakresie od 30 kHz do 45 kHz. Po każdej zmianie częstotliwości z ustalonym krokiem jest mierzona napięcie stałe („amplituda”) na emiterze tranzystora Q2. Program wyszukuje i zapamiętuje taką częstotliwość sterującą, przy której napięcie na emiterze Q2 jest najwyższe. Pokazane to na **listingu 4**.

Cewka układu detektora jest włączona pomiędzy bazę tranzystora u masę. Przybliżanie do niej ferromagnetyka powoduje wzrost indukcyjności, a co za tym idzie – zwiększenie reakcji cewki przy stałej częstotliwości. Amplituda przebiegu na bazie tranzystora maleje. Na **rysunku 19** pokazano oscylogramy przebiegów na bazie tranzystora Q2 przy cewce bez przybliżonego ferromagnetyka i potem przy cewce z przybliżonym przedmiotem stalowym. Układ wejściowy działa jak filtr dolnoprzepustowy i dlatego wejściowy sygnał prostokątny generowany przez układ NCO po odfiltrowaniu harmonicznych staje się przebiegiem sinusoidalnym.

Po przybliżeniu przedmiotu stalowego amplituda zmalała z 3,1 V na 2,7 V, a napięcie stałe na wyjściu układu czujnika z 880 mV do 610 mV. Zbliżanie i oddalanie przedmiotu stalowego powoduje, że napięcie na czujniku będzie się zmieniało mniej więcej w zakresie 610...880 mV. Teraz wystarczy to napięcie zmierzyć i odpowiednio interpretować. Procedura obsługi czujnika wyświetla na ekranie linie. Jej umiejscowienie na ekranie odzwierciedla stan napięcia na wyjściu

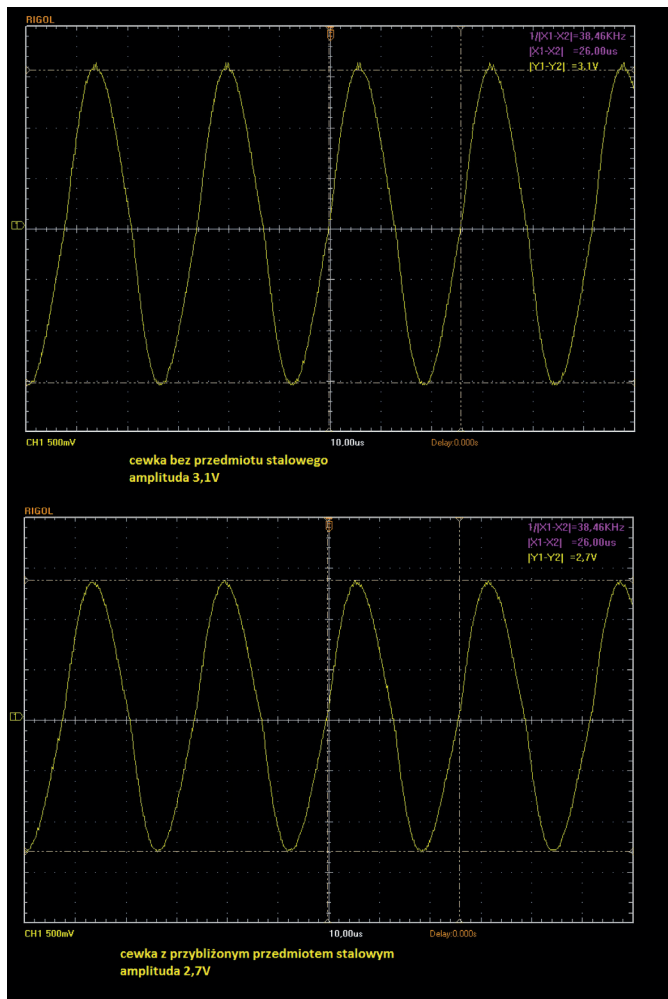
**Rysunek 16. Układ pomiaru ciśnienia****Rysunek 17. Cewka detektora metalu****Rysunek 18. Schemat układu detektora metalu****Listing 4. Skanowanie częstotliwości**

```
void metalDetectorGetPeakFreq() {
    uint16_t scf, i;
    uint32_t res = 0;

    ADREF = 0x00; // ADNREF VSS; ADPREF VDD
    enableNco();
    peakFreq = 0; // zerowanie częstotliwości
    peakAmpl = 0; // zerowanie „amplitudy”
    // pętla skanowania częstotliwości
    for(scf = scanFreqMin; scf < scanFreqMax; scf += scanFreqStep)
    {
        setNcoFreq(scf); //ustawienie częstotliwości NCO
        __delay_us(250);
        res = 0;
        //pomiar napięcia na wyjściu układu detektora
        for(i=0; i<8; i++)
        {
            ADCC_GetSingleConversion(channel_MetalDetector);
            res += ADCC_GetAccumulatorValue() >> 6;
        }
        res = res >> 3;

        //szukanie maksymalnej wartości napięcia i zapamiętanie częstotliwości
        if(res > peakAmpl) {
            peakAmpl = res;
            peakFreq = scf;
        }
    }
    setNcoFreq(peakFreq); //ustawienie częstotliwości po skanowaniu
    __delay_us(250);
    readMetalDetector(); //odczytanie detektora
    md10bitOffset = md10bit;
    md12bitOffset = md12bit;
}
}
```





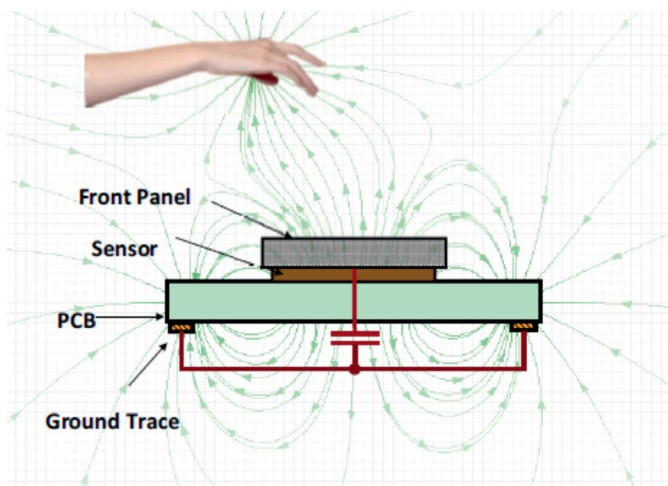
Rysunek 19. Oscylogramy przebiegów na bazie tranzystora Q2

czujnika. Jest też możliwość sygnalizacji dźwiękowej. Przy zbliżeniu metalu do cewki rośnie częstotliwość sygnału dźwiękowego, a przy oddalaniu maleje. Pomiar napięcia wyjścia detektora wykonuje procedura readMetalDetector zamieszczona na listingu 6.

### Czujnik odległości

Microchip od dawna rozwija technologię obsługi dotykowych czujników dotykowych mTouch. Wykrycie zadziałania czujnika wymaga dotknięcia palcem do dielektryka. Układ stale monitoruje pojemność czujnika względem masy i wykrywa jej znaczącą zmianę.

Czujnik odległości działa na takiej samej zasadzie, ale ponieważ musi działać na stosunkowo duże odległości, to zmiany pojemności



Rysunek 20. Zasada działania czujnika odległości

```
Listing 5. Ustawianie częstotliwości generatora NCO
void setNcoFreq(uint32_t freq) // for 32MHz System Clock
{
    freq *= 100;
    freq /= 1526;
    NCO1INC0 = ((freq >> 16) & 0xFF);
    NCO1INC1 = ((freq >> 8) & 0xFF);
    NCO1INC2 = ((freq >> 0) & 0xFF);
}

```

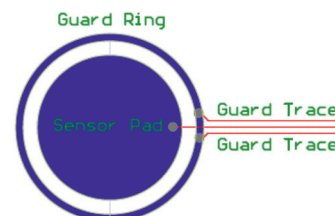
```
Listing 6. Pomiar napięcia z wyjścia układu czujnika metalu.
void readMetalDetector()
{
    uint32_t res;
    uint16_t i;

    ADR0 = 0x00; // ADNREF VSS; ADPREF VDD
    res = 0;
    for(i=0; i<32; i++)
    {
        ADCC_GetSingleConversion(channel_MetalDetector);
        res += (ADCC_GetAccumulatorValue() >> 6);
    }
    res = res >> 5;
    res = 4095 - res;
    md10bit = (res & 0xFFFFF0);
    md12bit = res;
}

```

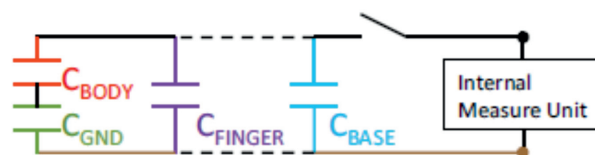
będą zdecydowanie mniejsze. Żeby wykrywanie było niezawodne trzeba dużo większej uwagi przy projektowaniu sensora tak żeby uzyskać dobry współczynnik sygnału użytecznego do szumu SNR (rysunek 20). Żeby osiągnąć maksymalnie niezawodne działanie czujnika trzeba przestrzegać w trakcie jego projektowania kilku zasad:

1. Odległość czujnika od ścieżek GND i wszystkich metalowych elementów na płycie powinna być jak największa.
2. Sensor powinien być tak duży, jak to tylko możliwe.
3. Można na płycie wykonać aktywną osłonę czujnika, jak na rysunku 20.
4. Trzeba wyeliminować ruch elementów układu względem czujnika. Na przykład, wyeliminować ruchy taśmy elastycznej przez zamocowania lub przyklejenie.
5. W układach zasilanych bateryjnie trzeba maksymalnie zwiększyć sprzężenie pomiędzy gruntem a czujnikiem. W układach z zasilaniem bateryjnym (a z takimi mamy tu do czynienia), zarówno ludzkie ciało jak i czujnik mają pojemność sprzęgającą z powierzchnią ziemi. Pokazano to na rysunku 21. Żeby układ zachował dobre parametry musimy dążyć, aby CGND była jak największa. Można to osiągnąć na przykład przez połączenie masy układu z uziemieniem sieci energetycznej lub jakimkolwiek innym uziemieniem o małej rezystancji (rysunek 22).

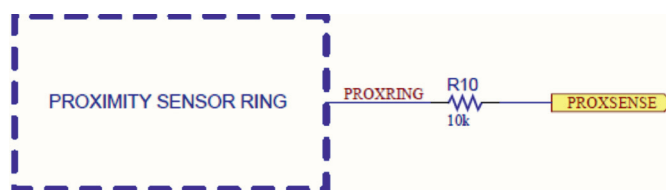


Rysunek 21. Aktywna osłona czujnika

Implementacja sprzętowa jest nieskomplikowana. Czujnik jest fragmentem metalowej ścieżki na płycie drukowanej i jest przyłączony do wejścia



Rysunek 22. Model sprzężenia pojemnościowych przy zasilaniu bateryjnym



Rysunek 23. Układ czujnika odległości

Listing 7. Inicjalizacja pomiaru odległości

```
void initProximity(void)
{
    proximityMin = 0xFFFF;
    proximityMax = 0;
    proximityTh = 0;
    proximitySamples = 64; //4096;
    clearPlot(&plot1, 0);
    clearPlot(&plot2, 0);
    readProximity();
    proximity10bitOffset = proximity10bit;
    proximity12bitOffset = proximity12bit;
    objectDetected = false;
}
```

Listing 8. Pomiar odległości

```
void readProximity(void)
{
    uint32_t res = 0;
    ADREF = 0x00; // ADNREF VSS; ADPREF VDD
    ADPRE = 50; // max 8191
    ADCON1bits.ADPOL = 0; // to VDD
    ADACQ = 25; // max 8191
    for (uint16_t i = 0; i < proximitySamples; i++)
    {
        ADCC_GetSingleConversion(channel_Proximity);
        res += ADCC_GetAccumulatorValue() >> 6;
    }
    res = res / (proximitySamples >> 2);
    ADPRE = 0;
    ADACQ = 0;
    if ( proximityMin > res )
    {
        proximityMin = res;
    }
    if ( proximityMax < res )
    {
        proximityMax = res;
        proximityTh = proximityMax;
    }
    proximity12bit = res;
    proximity10bit = res & 0xFFFFFFF;
}
```

przetwornika A/C za pomocą wąskiej ścieżki. Opcjonalny rezystor tworzy dodatkowy filtr dolnoprzepustowy. Na **rysunku 23** pokazano układ czujnika z naszego modułu. Mikrokontroler cyklicznie mierzy pojemność czujnika względem masy i wykrywa znaczące jej zmiany. Ta zmiana musi być znacząco wyższa niż poziom szumów pochodzących z układu mikrokontrolera, ale także ze środowiska zewnętrznego. Pomiar pojemności odbywa się techniką opartą

na ładowaniu i rozładowywaniu wewnętrznego kondensatora przetwornika A/C i naprzemiennym ładowaniu i rozładowywaniu kondensatora czujnika.

Zasada działania jest obszernie i dokładnie opisana w nocie aplikacyjnej Microchipsa AN1478 dostępnej na stronie [www.microchip.com](http://www.microchip.com). Na **listingu 7** pokazano procedurę inicjalizacji modułu pomiaru odległości, a na **listingu 8** procedurę odczytywania czujnika z wykorzystaniem przetwornika A/C.

## Uwagi końcowe

Opisywany moduł to interesujące połączenie cyfrowego środowiska mikrokontrolera ze światem analogowym. Pomiary napięcia, ciśnienia czy temperatury z użyciem scalonych czujników są wykonywane od dawna i również w tym wypadku nie są czymś nadzwyczajnym. Warto się jednak przyjrzeć dwóm ciekawym czujnikom: metalu i zbliżenia.

Czujniki przedmiotów stalowych są stosowane od dawna w automatyce. Ale tu mamy interesujący projekt, łatwy do wykonania i bardzo tani. Wykorzystanie taniego mikrokontrolera z wbudowanymi odpowiednimi peryferiami (A/C i NCO) oraz bardzo prostego układu czujnika pozwala na zbudowanie dobrze działającego czujnika „za grosze”. Podobnie jest z dobrze działającym czujnikiem zbliżenia. Tu potrzeby jest tylko jeden rezystor, odpowiednio zaprojektowany czujnik na płytce i tani mikrokontroler. To kolejny przykład na to, że firma przykłada dużą wagę do tej linii mikrokontrolerów i ich wykorzystywania często niekonwencjonalnego w tanich systemach sterowania i pomiarów.

Opisałem wybrane przeze mnie, najbardziej istotne procedury obsługi czujników. Pomiędzy fragmenty programu odpowiedzialne za wizualizację pomiarów na wyświetlaczu i sygnalizację dźwiękową. Na stronie Microchipsa jest do pobrania cały projekt programu wgranego fabrycznie do mikrokontrolera, z kompletnymi plikami źródłowymi. Każdy zainteresowany może przystosować zamieszczone tam procedury do swoich potrzeb.

Tomasz Jabłoński, EP

REKLAMA

# m.tody m.technik

Ciekawi świata są zawsze młodzi

w prezencie na każdą okazję



<https://goo.gl/TiDLmR>

przejrzyć i kupić na

[www.ulubionykiosk.pl](http://www.ulubionykiosk.pl)

