

Mikroprocesor jednokładowy, czyli opowieść o tępym (aczkolwiek wyjątkowo pracowitym) urzędniku biurowym

część 1

Aby we właściwy sposób korzystać z możliwości mikroprocesorów (mikrokontrolerów) jednokładowych, trzeba poznać nie tylko język C, ale też budowę i sposób pracy mikroprocesora. W poprzednim numerze EdW pewne zagadnienia zostały przedstawione w żartobliwy sposób. Trzymając się tej lekkiej konwencji, przyjrzymy się budowie i pracy procesora. W pierwszej kolejności omówimy właściwości wspólne dla wszelkich procesorów, a potem cechy szczególnie mikroprocesorów jednokładowych, jak porty, układy peryferyjne, sposoby ich sterowania oraz ściśle z tym związaną organizację pamięci. Od razu zasygnalizujemy też pewne ważne problemy, związane z programowaniem.

Urzędnik w biurze

Realizację programu w (8-bitowym) mikroprocesorze jednokładowym możemy porównać do pracy ograniczonego umysłowo urzędnika, pracującego w specyficznym biurze. Rodzajów takich specyficznych biur jest wiele, my skoncentrujemy się na odmianie, nazwanej nie wiadomo dlaczego AVR. Biuro, które poznamy, to niewielki budynek, na którego ścianie ktoś, zupełnie niezgodnie z zasadami ortografii, napisał kredą ATmega328.

W tym niewielkim biurze pracuje urzędnik, który nie grzeszy inteligencją, ale za to jest ogromnie skrupulatny i nieprawdopodobnie pracowity. Jest tak ograniczony umysłowo, że nie potrafi zrozumieć i zrealizować trudniejszych zadań, a jedynie najprostsze. Zupełnie nieprawdziwe są plotki, że studiował na Harvardzie. Absolutnie nie! Tylko architektura jego biura ma jakiś związek z Uniwersytetem Harvarda (ale to strasznie archaiczna historia, dotycząca jakiegoś starego, pierwszego Marka).

Nasz kiepsko rozczytany urzędnik jest natomiast piorunsko szybki i jego miejsce pracy jest bardzo przejrzyste i uporządkowane.

Rodzaje pamięci

W biurze od razu rzuca się w oczy bardzo długi szereg maleńkich szufladek czy raczej przegródek, w których umieszczone są maleńkie karteczki. Na tych maleńkich karteczkach umieszczone są albo elementarne, bardzo proste rozkazy, albo niewielkie liczby. Co ważne, szufladki przegródki są kolejno ponumerowane i nasz urzędnik bez problemu potrafi znaleźć przegródkę o podanym numerze, czyli adresie.

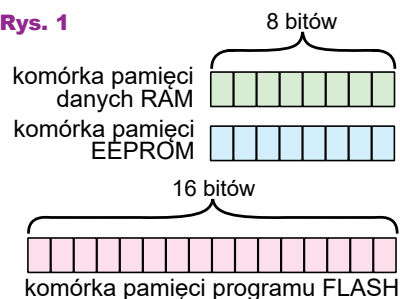
Urzędnik ma w biurze trzy długie szeregi takich przegródek-komórek. Jeden, baaardzo długi rząd przegródek nazywany jest (o dziwo) **pamięcią FLASH**. W różnych biurach tego typu może być od kilku tysięcy do kilku milionów przegródek (komórek pamięci FLASH). W biurze, które zwiedzamy, oznaczonym ATmega328, takich komórek FLASH jest ponad 32 tysiące. W tych przegródkach maleńkie karteczki są zafoliowane i nasz tępny urzędnik zasadniczo nie może na nich nic zapisać, a jedynie odczytać. Bo w tych przegródkach umieszczone są elementarne rozkazy programu, które urzędnik ma kolejno wykonywać. Generalnie praca urzędnika, czyli **realizacja programu, jest dziecinnie prosta: polega na tym, że odczytuje on rozkazy z kolejnych komórek-przegródek pamięci FLASH i skrupulatnie je wykonuje.**

Drugi, znacznie krótszy rząd przegródek-komórek, nazywa się, też nie wiadomo dlaczego, **RAM** (niektórzy nazywają go **SRAM**). Tych przegródek-komórek może być tylko kilkadziesiąt, ale może być nawet kilkanaście tysięcy. W tym konkretnym biurze jest ich nieco ponad 2000. Tu karteczki są mniejsze (o połowę węższe), ale nie są zafoliowane. Na początku pracy karteczki są puste, a nasz pracowity urzędnik ma dobry ołówek z gumką, może więc je błyskawicznie zapisywać, odczytywać, a także wycierać gumką starą treść i wpisywać nową. Na koniec pracy urzędnik wszystkie te karteczki RAM wyrzuca do kosza, a następnego

dnia znów zaczyna pracę z nowym zestawem kartek-komórek RAM.

W biurze jest też trzeci rząd przegródek, zwany **EEPROM** (w tym biurze jest ich 1024). Urzędnik też może pisać na maleńkich karteczkach, ale po zapisaniu musi każdą zafoliować, więc schodzi mu na to sporo czasu. Trudniej się do nich dostać, ale gdy trzeba, potrafi nie tylko odczytać, ale też zmienić zapis na karteczkach z pamięci EEPROM, ale na to też trzeba czasu, bo karteczkę z nowym zapisem także trzeba zafoliować. **Rysunek 1** pokazuje pojedyncze karteczki-komórki pamięci.

Rys. 1



I tu ciekawostka: nasz urzędnik, choć niewątpliwie ograniczony umysłowo, jest też dziwakiem – ekscentrykiem: wszystko zapisuje w kodzie dwójkowym - binarnym (popularna plotka głosi, że nasz system dziesiętny jest dla niego za trudny). Każda karteczka-komórka pamięci RAM i EEPROM jest jednakowa i ma osiem kratek, a więc mieści *osiem bitów*, co nazywamy *bajtem*. W pojedynczej kratce karteczki nie można zapisać dowolnego znaku, np. litery, a jedynie albo zero, albo cyferkę 1. W ośmiu kratkach, czyli ośmiu bitach (w jednym bajcie) można zapisać dowolną kombinację zer i jedynek – jest ich w sumie 256. Karteczki-komórki pamięci FLASH są dwubajtowe – każda ma 16 bitów, co daje 65536 kombinacji zer i jedynek.

Jak już wiemy, wszystkie przegródki-komórki pamięci są ponumerowane. Ciekawostką jest, że u naszego ekscentryka

Rys. 2

Komórki pamięci RAM	
numer-adres w postaci liczby szesnastkowej (hexadecimal)	numer-adres w postaci liczby dwójkowej
0x00	00000000
0x01	00000001
0x02	00000010
0x03	00000011
0x04	00000100
0x05	00000101
0x06	00000110
0x07	00000111
0x08	00001000
0x09	00001001
0x0A	00001010
0x0B	00001011
0x0C	00001100
0x0D	00001101
0x0E	00001110
0x0F	00001111
0x10	00010000
0x11	00010001
0x12	00010010
0x13	00010011
0x14	00010100
0x15	00010101
.....
	następne komórki

numery przegródek (adresy komórek) zaczynają się nie od jedynek, tylko od zera. Pierwsza komórka na numer-adres zero, druga: jeden, trzecia: dwa i tak dalej.

Oczywiście urzędnik posługuje się numerami-adresami zapisanymi w kodzie dwójkowym (binarnym, w skrócie **bin**). Jednak programiści, którzy przygotowują dla niego programy nie lubią rozwleklego kodu dwójkowego, tylko „skracają go” i wykorzystują tzw. kod szesnastkowy (hexadecimal – w skrócie **hex**). **Rysunek 2** pokazuje numerację komórek pamięci RAM, podaną w różnych kodach. Mam nadzieję, że znasz podstawy systemu dwójkowego i szesnastkowego, a jeśli nie, to poszukaj w innych źródłach – to w sumie proste i nie będziemy się tym zajmować.

Liczby i nie tylko

Urzędnik ma do czynienia z podanymi w kodzie dwójkowym:

- zawartością komórek pamięci
- adresami komórek pamięci

W przypadku adresów wiadomo, że dany kod dwójkowy to liczba – numer przegródki. Ale nie zawsze kombinacja zer i jedynek to liczba dwójkowa. Nawet nierozgarnięty urzędnik wie, że w pamięci FLASH zamieszczone są instrukcje, czyli rozkazy – ściślej 16-bitowe kody rozkazów. Wie także, że niektóre rozkazy dodatkowo zawierają „w sobie” jakieś dane: jakąś liczbę (*parametr*) albo numer jakiejś innej przegródki. Istnieje też kilka „podwójnych” rozkazów; wtedy urzędnik dodatkowo wyjmuje karteczkę z następnej przegródki i zależnie od rodzaju rozkazu zawartość tej drugiej przegródki traktuje jako numer przegródki, z której ma wziąć następny rozkaz albo dane.

Nie ma więc żadnego problemu z dwójkowymi kodami zawartymi w pamięci FLASH, ponieważ generalnie są to rozkazy. Jest jednak ogromnie ważny problem z pamięcią RAM.

Typy danych

Otóż nasz sumienny urzędnik potrafi błyskawicznie znaleźć przegródkę-komórkę

w pamięci RAM, gdy podamy mu jej numer, czyli adres, by odczytać jej osmiobitową zawartość lub (jeśli to możliwe i dopuszczalne) coś w niej zapisać. Jeśli chodzi o rozróżnienie *typu danych* w pamięci RAM i EEPROM, to niestety, urzędnikowi to „wisi i powiewa”. On gotów jest zupełnie bezmyślnie przetworzyć dwójkową zawartość komórki pamięci zgodnie z treścią aktualnie wykonywanego elementarnego rozkazu. Nie zastanawia się, co tak naprawdę reprezentuje osiem bitów.

Problem w tym, że **wcale nie musi to być osmiobitowa liczba dwójkowa**, czyli liczba z zakresu od zera do 255, co dwójkowo zapiszemy 00000000₍₂₎ do 11111111₍₂₎. Są też różne inne możliwości. Otóż kombinacja ośmiu zer i jedynek może też oznaczać literę lub inny znak w tak zwanym kodzie ASCII. Poszczególne bity bajtu, choć zapisane razem, mogą być niepowiązane i wtedy każdy bit należy traktować oddzielnie. Nawet gdy zawartość bajtu ma być traktowana jako liczba, wcale nie musi to być liczba dwójkowa z zakresu 0...255. Może to być liczba z zakresu 0...99 zapisana w tak zwanym kodzie BCD. A gdy potrzebna jest liczba większa niż 255, to trzeba ją „pokawałkować” i umieścić kawałki w kilku przegródkach – bajtach. Dwie przegródki (2 x 8), czyli 16 bitów pomieści liczbę z zakresu 0...65535, natomiast trzy przegródki (24 bity) pozwalają zapisać liczbę z zakresu 0...16777215. Są też sprytnie sposoby, żeby dwójkowo zapisywać liczby ujemne (ze znakiem), liczby niecałkowite, czyli ułamkowe, oraz tak zwane **tablice (arrays)**, a także inne dziwaczne **struktury**.

Jednak nasz urzędnik zupełnie nie rozumie takich zaawansowanych zagadnień. Owszem, potrafi sprawnie przeprowadzić operacje, w tym obliczenia arytmetyczne, ale tylko te podstawowe: dodawanie, odejmowanie, ewentualnie proste mnożenie, i to na „kawałkach” osmiobitowych, a nie na dużych liczbach i na dodatku nie sam, tylko z pomocą pokracznego i dość prymitywnego kalkulatora. Samodzielnie nie potrafi wykonać dzielenia, potęgowania i innych operacji matematycznych – może je jednak dla nas zrealizować niejako nieświadomie: korzystając z właściwie napisanego programu i wspomnianego prostego kalkulatora, może „dużą” liczbę podzielić na osmiobitowe „kawałki” przeprowadzić obliczenia i z „kawałków” złożyć wynik.

Gdy dane w pamięci nie reprezentują liczby, trzeba je przetwarzać inaczej.

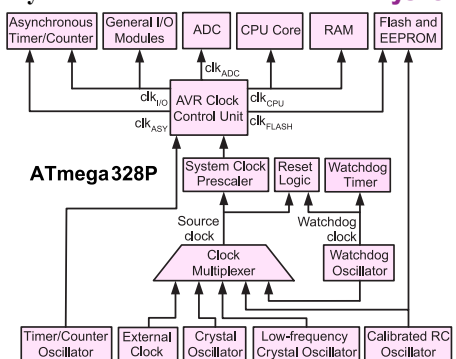
Ponieważ urzędnik robi wszystko gorliwie, ale bezmyślnie, ktoś inny przygoto-

wując mu program działania, musi starannie **rozróżnić typy danych**, żeby czegoś nie pomieszać podczas obliczeń matematycznych i innych operacji. Właśnie **rozróżnianie typów danych jest ogromnie ważne podczas pisania programu** dla takiego nierozgarniętego urzędnika.

Wyposażenie biura

Wracamy do opisu biura: jest tam wielki dzwon, który umarłego by obudził i który jest nazywany nie wiadomo dlaczego **RESET**. Dźwięk jest tak porażający, że na głos tego wielkiego dzwonu nasz urzędnik jest tak wstrząśnięty (i zmieszany), że niezależnie od tego, co robił wcześniej, „w popłochu rzuca wszystko w kąt” i zaczyna pracę od samego początku, czyli od karteczki z pierwszej przegródki pamięci FLASH.

Zadziwiająca pracowitość i szybkość naszego urzędnika zapewne nie wynika z niego samego: chodzą słuchy, że służył on kiedyś w wojskowej kompanii honorowej, bo wszystko wykonuje bardzo rytmicznie, zawsze **w rytm zegara**, który jest ważnym elementem biura. A ten jego zegar też jest jakiś dziwaczny. Właściwie jest to cały system zegarowy, który może pracować szybciej lub wolniej, albo też takty zegara mogą przychodzić lub być synchronizowane z zewnątrz. Ktoś kiedyś narysował schemat budowy systemu zegarowego tego biura – masz go na **rysunku 3**.

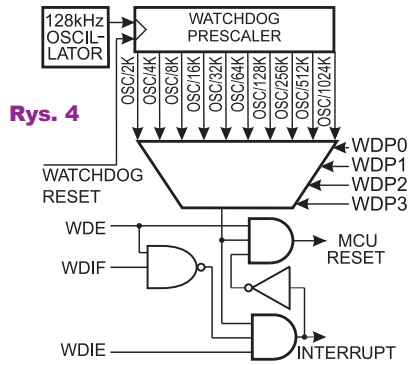


W każdym razie urzędnik zawsze wszystko robi w takt zegara, tylko ktoś musi mu z zewnątrz wcześniej ten zegar skonfigurować, podobno za pomocą jakichś fusów (**fuse bits**), które na dodatek są w jakimś sensie „odwrotne”. Choć system zegarowy jest skomplikowany, zazwyczaj pracuje według „fabrycznych” ustawień i ten szczegół nie jest problemem.

Z pamięcią u naszego urzędnika kiepsko, z wyobraźnią jeszcze gorzej. Nie pamięta nawet, co ma zrobić za chwilę, więc by się nie pomylić, oprócz tykającego zegara ma też oddzielną większą karteczkę (jest to właściwie kawałek grubej przezroczystej folii, na

której pisze flamastrem). Tę kartkę-folię nazywa **licznikiem programu (program counter – PC)** i traktuje jak wyrocznię. Ktoś mu kiedyś dobrze poradził, żeby po każdym „cyknięciu” (takcie) zegara zapisywał numer (adres) komórki, z której wziął ostatni rozkaz, ale zwiększony o jeden. Inaczej mówiąc, podczas normalnej pracy zapisuje tam numer komórki, z której weźmie następny rozkaz w kolejnym takcie zegara. Co ciekawe, adres komórki zapisany na gładkiej folii (*zawartość licznika programu*) można bardzo szybko wymazać i zastąpić nowym adresem. Oczywiście w następnym takcie zegara nasz gorliwy biurokrata bez zmużenia oka i żadnej refleksji weźmie rozkaz z komórki, której adres ma zapisany na folii.

W biurze można wykorzystać lub nie także inny rodzaj zegara: małą dziwną klepsydę, a właściwie minutnik zwany (też nie wiadomo dlaczego) łoczdogiem (*watchdog*). Nikt nie wie, czy urzędnik kiedyś pracował przez jakiś czas na kolei jako maszynista, bo ten minutnik, jeśli zostanie włączony, działa



bardzo podobnie jak tak zwany aktywny czuwał w pociągu. Jest kilka sposobów jego wykorzystania, ale najprościej biorąc, jeżeli urzędnik co jakiś czas nie naciśnie przycisku pracującego czuwaka-minutnika, to po określonym czasie włączy się wielki dzwon, czyli RESET. Ten minutnik uruchamiający RESET może wtedy służyć do kontroli tego, czy urzędnik (wskutek niedoskonałości programu) nie *zawiesił się*, wykonując nie to co trzeba. Minutnik można też wykorzystywać, by pozwolić maszyniście w ciągu dniówki pójść spać (tak!) – by co jakiś czas

budzić go minutnikiem z głębokiego snu do wykonania niezbędnych zadań. Dostępny jest schemat blokowy minutnika-watchdoga, (*rysunek 4*), który okazuje się dość prosty.

Ogólnie biorąc, gdy urzędnik bardzo pilnie i szybko pracuje, wtedy zużywa mnóstwo energii i musi bardzo dużo jeść („bardzo dużo” to sprawa względna, ale często jest to kluczowa kwestia praktyczna). Aby nie zużywał tyle energii i jadł dużo mniej, można po pierwsze ustawić powolny takt zegara, a po drugie można kazać urzędnikowi iść spać i budzić go tylko co jakiś czas. Jedną możliwością to głęboki sen (*Powerdown*) na dyżurnej leżance, który można przerwać tylko dźwiękiem wielkiego dzwonu. Inne możliwości to jeden z kilku rodzajów płytkiej drzemki (*Idle*) na siedząco przy biurku – lekkiej drzemki, którą może przerwać dźwięk któregoś małego dzwonka.

O „małych dzwoneczkach”, które często *przerwywają* pracę urzędnika, i dziwnych, związanych z nimi *flagach*, opowiem Ci za miesiąc.

Piotr Górecki

Ciąg dalszy ze strony 27

Podczas ładowania dostarczana jest energia, która zamienia substancje chemiczne w ich odmiany „o większej energii chemicznej”. Potem tę energię można odzyskać. W popularnych akumulatorach samochodowych materiałami czynnymi jest ołów (Pb) i kwas siarkowy (H₂SO₄), w tak zwanych akumulatorach wodorkowych NiMH – związki niklu (Ni), a w coraz popularniejszych akumulatorach litowych – oczywiście związki litu (Li).

Przetworniki piezoelektryczne. Wcześniej wykorzystywaliśmy takie odwracalne, czyli dwukierunkowe przetworniki energii jak *silnik – prądnica* oraz *głośnik – mikrofon*. Następuje w nich dwukierunkowa zamiana energii mechanicznej i elektrycznej, a wszystko to na zasadzie ruchu przewodu w polu magnetycznym. Podobnie odwracalne

(dwukierunkowe) są też **przetworniki piezoelektryczne**. Najpopularniejszy przetwornik piezo to metalowa blaszka – membrana z naniesionym ceramicznym materiałem piezoelektrycznym – **fotografia 7**. Przy odkształceniu piezoelektryk wytwarza duże napięcie. Na **fotografii 8** pokazane są takie przetworniki z dołączonymi diodami LED (najlepiej niebieskie lub białe). Do jednej blaszki dolutowane są dwa druty. Lekkie wygi-

nanie przetwornika (uwaga na ryzyko pęknięcia piezoelektryka) powoduje chwilowe migotanie diod LED. Przetworniki piezoelektryczne mają też inne formy (kształty). Na przykład w piezoelektrycznych zapalniczkach odkształcenie kostki piezo powoduje wytworzenie wysokiego napięcia rzędu tysięcy (!) woltów, co powoduje przeskok iskry i zapalenie gazu. Nie jest to jednak groźne, bo prąd jest mały, pojawia się tylko przez chwilę, więc całkowita energia jest niewielka. Na **fotografii 9** widzisz piezoelektryczną zapalniczkę do gazu.

Zapamiętaj, że jeśli wykorzystujemy elementy piezoelektryczne, to mamy do czynienia z wysokimi napięciami, od kilkunastu woltów do tysięcy woltów, ale z bardzo małymi, często znikomo małymi prądami. Odwrotnie jest w przypadku termopar. Zajmiemy się nimi w następnej części.

Piotr Górecki

Fot. 7



Fot. 9



Fot. 8

