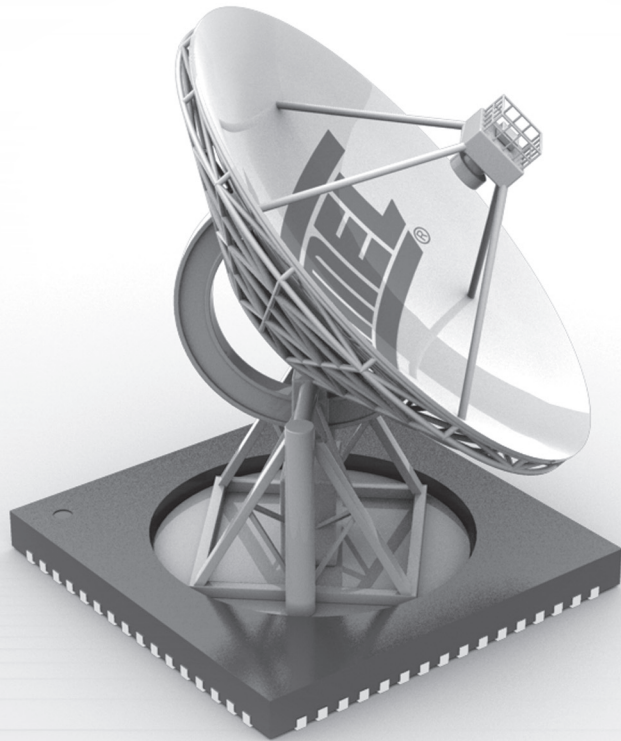


---

## Wstęp

---



W czasach lawinowego rozwoju techniki cyfrowej nikogo nie dziwi obecność mikrokontrolerów nawet w najmniej skomplikowanych aplikacjach układowych, takich jak zabawki, sprzęt AGD czy proste urządzenia przenośne. Wszak przy cenie poniżej 1 \$ możemy kupić całkiem przyzwoity mikrokontroler, jakim jest choćby ATmega8 firmy Atmel. W związku z tym to, co do niedawna było domeną dużych i bogatych koncernów elektronicznych, dzisiaj trafia pod przysłowiowe strzechy... przede wszystkim za sprawą przyjaznych i prostych w implementacji języków wysokiego poziomu, dzięki którym napisanie gotowej i sprawnej aplikacji nigdy nie było tak łatwe! I nie ma tu, w moim przekonaniu, znaczenia, jakiego rodzaju językiem się posługujemy. Kod wynikowy i tak będzie, czy chcemy, czy nie, kodem maszynowym niezależnie od tego, jak bardzo pragnęlibyśmy to zmienić (choćby tylko po to, by podkreślić znajomość jakiegoś języka programowania, która miała być nobilitacją dla naszego projektu). Nie mam, tym samym, zamiaru wdawać się w dysputy, który język jest lepszy, a który gorszy, bo w moim przekonaniu końcowa jakość aplikacji zależy przede wszystkim od znajomości architektury mikrokontrolera, arkanów języka programowania oraz, co najważniejsze, kreatywności i inteligencji programisty. To był także przyczynek, dla którego powstała niniejsza książka. W zamierzeniach ma łączyć to, co maszynowe, z tym co... ludzkie – informacje o sprzęcie z opisem języka i przykładami jego wykorzystania.

A dlaczego Bascom? Z trywialnej przyczyny! Dla mnie, nauczzonego akademickiego porządku i przejrzystości, każdy język zbliżony swoją składnią do Basica czy Pascala będzie wyborem oczywistym jako wzór wspomnianych cech. Dodatkowo Bascom zwalnia nas z konieczności opracowywania wielu specyficznych dla sprzętu procedur i funkcji (choćby obsługa magistrali 1-Wire czy protokołu TCP/IP), pozwalając skupić się nad tym, co najistotniejsze, na aplikacji!

Z drugiej jednak strony znajomość, choćby najlepsza, tego czy innego języka wysokiego poziomu bez zagłębiania się w arkana budowy układu jest niewystarczająca. Szybko przekonamy się, iż bez wiedzy na temat funkcjonowania i architektury mikrokontrolera, który ma być sercem naszego urządzenia, nie będziemy w stanie dobrze napisać bardziej rozbudowanej aplikacji, nie mówiąc już o tym, że trudno będzie nam wykryć w niej źródła potencjalnych problemów. Poza tym sam język, z uwagi na jego ogromną uniwersalność, nie zawsze dostarcza nam rozwiązań szytych na miarę, gdyż musi on działać na różnych platformach sprzętowych. Stąd właśnie poniekąd pomysł na połączenie tych dwóch światów, który – mam nadzieję – przypadnie Wam Czytelnikom do gustu.

*Robert Wołgajew, 2010*