

Wstęp

Dzisiejszy rynek elektroniki użytkowej wymaga od projektantów i elektroników hobbystów, by konstruowane urządzenia nie tylko były użyteczne, ale i funkcjonalne. Poprawność działania i konstrukcji produktu to nie wszystko. Przy dużej i wciąż rosnącej liczbie konkurentów na rynku elektroniki użytkowej należy zadbać, by urządzenie miało nowoczesny design, a czas realizacji produktu, od momentu złożenia zamówienia przez potencjalnego klienta, był jak najkrótszy. Co więc zrobić, by nadążyć za dynamicznie rozwijającym się rynkiem?

Wzrost mocy obliczeniowej współczesnych mikrokontrolerów i mikroprocesorów powoduje, że wielu projektantów systemów wbudowanych korzysta w realizacjach swoich projektów z gotowych rozwiązań w postaci komputerów jednopłytkowych częściej, niż z budowanych od podstaw systemów tworzonych na podstawie 8- lub 32-bitowych mikrokontrolerów. Przyczyn takiej sytuacji jest wiele. Budowanie systemu od podstaw wiąże się z długim i podatnym na wystąpienie błędu procesem projektowania obwodu, a także żmudnym i kosztownym etapem testów. Co więcej, przy braku doświadczenia oraz wiedzy z zakresu projektowania i konstrukcji, urządzenie może nie spełniać wymaganych norm. Wyższa cena urządzenia, związana z wykorzystaniem gotowego komputera jednopłytkowego, rekompensowana jest dużą oszczędnością czasu, a także większą niezawodnością naszego projektu. Sam programista może skupić się wyłącznie na tworzeniu aplikacji docelowej, której proces przygotowania jest znacznie ułatwiony ze względu na pewną abstrakcję sprzętu, jaką dostarcza nam system operacyjny, a także możliwości wykorzystania gotowych stosów wielu protokołów komunikacyjnych.

Od czego więc zacząć poznawanie komputerów jednopłytkowych i specyfiki ich programowania? Odpowiedź jest prosta – od tej książki.

W ramach niniejszej pracy przedstawiono podstawy wykorzystania komputerów jednopłytkowych, których budowa opiera się na procesorach z rdzeniem ARM oraz wykorzystujących darmowy system operacyjny Linux. Sam Linux w systemach wbudowanych nie jest tematem nowym. Świadomie lub mniej świadomie zapewne nieraz korzystaliśmy już z dobrodziejstw płynących z użycia tego systemu w urządzeniach embedded. Pomimo, że powstało już kilkanaście pozycji opisujących tę tematykę (większość anglojęzycznych), nadal brakuje jednak książki, która od kompletnych podstaw w sposób niebudzący zniechęcenia wprowadzi Czytelnika w ten pasjonujący świat. Moim marzeniem, jako autora, było przygotowanie takiej książki, która zarażałaby Czytelnika tą ciekawością i pasją, z jaką ja poznawałem i poznaję dziedzinę Linuksa w systemach wbudowanych.

Wiele książek poświęconych Linuksowi już na starcie wymaga od Czytelnika dobrej znajomości samego systemu i zagadnień związanych z programowaniem, natomiast większość pozycji poruszających temat mikrokontrolerów i mikroprocesorów zakłada znajomość podstaw elektroniki. Przystępując do zbierania materiału na niniejszą książkę, przyjąłem wyłącznie jedno założenie – brak założeń. Tak więc, jeżeli drogi Czytelniku nie miałeś wcześniej do czynienia z mikrokontrolerami bądź systemem Linux – nie przejmuj się. Jedyne, czego będziesz potrzebował, to podstaw znajomo-

ści języka C i dużej chęci poznania niezwykle ciekawego świata programowania i wykorzystania w praktyce komputerów jednopłytkowych z procesorami ARM.

Wszystkie informacje zawarte w niniejszej książce ułożyłem tak, aby były spójne i logiczne, a sama książka miała formę ciekawego kursu – od omówienia podstaw Linuksa, przez *Hello World* do pełnowartościowych i funkcjonalnych aplikacji działających w przestrzeni użytkownika. Podczas prac nad książką położyłem duży nacisk na to, by pozycja ta nie była kolejnym przeładowanym teorią podręcznikiem, dlatego zagadnienia czysto teoretyczne wplotłem wprost w realizowane przykłady. Mam nadzieję, że nie wprowadza to dodatkowego chaosu, a jedynie pozwala lepiej zrozumieć i zobrazować poruszane zagadnienia.

Pomimo dużej złożoności tematu postarałem się dokonać zarysu wszystkich, niezbędnych do dalszych samodzielnych działań, zagadnień. W rozdziale zatytułowanym *Informacje podstawowe* omówione zostały zagadnienia związane z zasadnością użycia systemów operacyjnych w urządzeniach wbudowanych oraz płynącymi z tego tytułu korzyściami. Dodatkowo omówiono procesory z rdzeniem Cortex-A oraz wykorzystujące ten rdzeń układy serii OMAP3, na bazie których przygotowano przykładowe projekty. Kolejny rozdział to skrótowe przedstawienie Linuksa – jego historii, modelu i podstawowych zagadnień związanych z dostępem do sprzętu. Jest to również ostatni z czysto teoretycznych rozdziałów.

Rozdział 3 to niezwykle ważny kurs przygotowania narzędzi programowych i sprzętowych. Duża liczba dostępnych możliwości w zakresie przygotowania narzędzi programowych często zniechęca początkujących do rozpoczęcia prac z Linuksem embedded. Rozdział ten w niezwykle prosty sposób ukazuje proces ich konfiguracji. Od strony czysto elektronicznej omówimy proste układy interfejsowe, niezbędne do dalszych prac z komputerami jednopłytkowymi.

Rozdział 4 – *Uruchomienie systemu na platformie docelowej* – opisuje poszczególne etapy przygotowania kompletnego systemu na potrzeby urządzeń wbudowanych. To w tym rozdziale przekonamy się, że kompilacja jądra systemu jest niezwykle prosta, a bootloader i ich konfiguracja nie będą miały przed nami żadnych tajemnic. Również w tym rozdziale, jak nakazuje tradycja, napiszemy naszą pierwszą aplikację *Hello World*, która przetrze nam szlaki do dalszych działań.

Rozdział 5 to kolejny milowy krok w poznawaniu systemu Linux i jego zastosowaniach w urządzeniach embedded. Dział ten w sposób kompletny poświęcono jednemu z najprostszych układów peryferyjnych – portom GPIO. To tu dowiemy się, jak przygotować jądro systemu i jak obsłużyć z poziomu konsoli wejścia i wyjścia cyfrowe. Poznamy również sposoby przygotowania prostych aplikacji w języku C komunikujących się ze sprzętem (obsługa przycisków, wysterowanie przekaźników), ale również przeanalizujemy bardziej złożone przykłady – sterownik astronomiczny. Przekonamy się, czy w Linuksie „wszystko jest plikiem”, dowiemy się, czym są sygnały i po co je wykorzystujemy, a także omówimy podstawy połączeń sprzętowych diod, przekaźników, przycisków.

Kolejny rozdział – *Szeregowe interfejsy komunikacyjne* – pozwala zapoznać się w sposób kompletny – od konfiguracji jądra do praktycznych przykładów – z podstawami obsługi i wykorzystania najpopularniejszych interfejsów szeregowych w systemie Linux. Zaprezentowane przykłady ukazują niezwykle łatwość prac

z magistralami 1-wire, I2C oraz RS-232 pod systemem Linux. Na podstawie zdobytych wiadomości przeanalizujemy kolejne realizacje praktyczne – rejestrator temperatury, logger GPS oraz obsłużymy zewnętrzny przetwornik A/C i C/A. Nadal będziemy sukcesywnie poszerzać naszą wiedzę na temat Linuksa – zapoznamy się z podstawami języka AWK oraz tworzeniem skryptów powłoki.

Rozdział 7 to przykłady wykorzystania menedżera *opkg* oraz gotowych pakietów oprogramowania. Przekonamy się, jak szybko i łatwo, wykorzystując gotowe rozwiązania, uruchomić serwer WWW czy nawiązać łączność Bluetooth z innymi urządzeniami.

Ze względu na abstrakcję warstwy sprzętowej, jaką dostarcza system operacyjny, wszystkie przykłady praktyczne zaprezentowane w niniejszej książce można z powodzeniem uruchomić na dowolnej platformie sprzętowej, tak więc mam głęboką nadzieję, że niniejsza książka i dobrana tematyka będą dobrym przewodnikiem po świecie Linuksa dla wszystkich elektroników, zaczynających przygodę z komputerami jednopłytkowymi, oraz cenną wskazówką dla informatyków pragnących zgłębić wiedzę z zakresu elektroniki.

Z własnego doświadczenia wiem, że czasami człowiekowi do wielkich działań potrzeba wyłącznie małego impulsu, który wskaże pierwszą drogę. Niech tym impulsem i motywacją do działania będzie ta oto książka, która w sposób łagodny i od podstaw wprowadzi Cię w świat Linuksa i systemów embedded.



Wszystkie programy narzędziowe i kody źródłowe aplikacji omawianych w niniejszej książce są dostępne na stronie internetowej Wydawnictwa BTC oraz na stronie domowej autora:
www.lukasz-skalski.com.

Lukasz Skalski