



Magic Matrix (2)

Budowa sterownika magicznie kolorowego wyświetlacza

Przygotowanie projektu magicznie kolorowego zegara typu smartwatch od strony elektronicznej wymagało sporo przemyśleń, nie tylko pod kątem podstawowych zasad elektroniki, lecz także funkcjonalności tzw. interfejsu użytkownika. Pod tym pojęciem kryją się zagadnienia związane chociażby z obsługą przycisków, odbiornika podczerwieni, aby można było użyć do sterowania tzw. pilota czy fotorezystor odpowiedzialny za automatyczną regulację jasności, aby w nocy urządzenie nie przeszkadzało w spokojnym zasypianiu i nie raziło ostrym światłem w nocy po przebudzeniu.

Zagadnienie tworzenia przyjaznego UI (User Interface) zawsze spędza sen z powiek wielu programistom i często jest albo pomijane szerokim łukiem, albo wykonane zgodnie z tym, co wymyśli programista, a nie użytkownik niemający pojęcia o programowaniu i elektronice. Chciałbym zatem przybliżyć zasady tworzenia projektu, które już na etapie opracowywania układu elektronicznego powinny przewidywać przyszły interfejs użytkownika. Trzeba zatem przyznać, że etapem tworzenia UI jest obarczony częstokroć w równym zakresie zarówno konstruktor elektroniki, jak i programista.

Przejdźmy do sedna sprawy. Najczęściej konstruktor rozpoczyna swoje plany projektu (pomijając już fakt, że nie myśli zwykle na początku o obudowie dla

urządzenia, odsuwając to zawsze na koniec, ze stwierdzeniem „później dopasuje się jakąś obudowę”) od rozważań na temat np. wyboru rodzaju mikrokontrolera, przechodząc od razu do rozważań związanych z pojemnością jego pamięci Flash, RAM itp. Nie tędy droga! Moim zdaniem należałoby na początek przemyśleć właśnie Interfejs użytkownika, czyli „co otrzyma końcowy użytkownik do ręki” po to, aby wygodnie obsługiwać projektowane urządzenie. To właśnie ten etap pracy, po wcześniejszym zaplanowaniu obudowy, jest bodajże najważniejszy. To właśnie na etapie pierwszych wstępnych rozważań, powinniśmy ustalić, w przypadku naszego Magicznego Zegara, jakie oddamy użytkownikowi narzędzia pozwalające na jego podstawowe nastawy,

przygotowanie do pacy, czy też przełączenie trybów pracy. Dzięki temu, jeśli ustalimy na początku, czego będziemy potrzebowali dla użytkownika urządzenia, to na tej podstawie łatwiej będzie później oceniać potrzebne zasoby elektroniki.

W związku z powyższym proponuję wcielić się razem ze mną w rolę konstruktora. Cóż, zastanówmy się, gdybyśmy sami otrzymali jakiś zegar, nieważne czy elektroniczny, czy mechaniczny, to która potrzeba byłaby na pierwszym miejscu? Myślę, że tu każdy odpowie, że chodzi o możliwość ustawiania godziny, daty itp. Teoretycznie jasna sprawa, lecz w dzisiejszych czasach przy dość ogromnych możliwościach różnorodnych modułów elektronicznych, szybko może się okazać, że tworzenie interfejsu użytkownika opartego wyłącznie o przyciski pewnie spowoduje, że urządzenie jako całość wyda się nawet laikowi mało atrakcyjne, a to dlatego że w tym obszarze będzie pewnie najmniej intuicyjne i wymagać będzie czytania (zwykle nie lubianej) instrukcji. Czy możemy coś na to zaradzić? Pewnie, że tak. Po pierwsze, należałoby zapewnić nie jeden, ale nawet kilka alternatywnych sposobów konfiguracji urządzenia, tak to nazwijmy. Pierwsze, co przychodzi na myśl,

to pilot podczerwieni, zapewniający dużo większy komfort, a jeśli będzie odpowiednio dobrany, jeśli będzie posiadał klawiaturkę numeryczną, to już drastycznie ułatwi użytkownikowi posługiwanie się naszym urządzeniem. Czy to wystarczy? To znaczy, przyciski, pilot podczerwieni, w dzisiejszych czasach? Zdecydowanie nie. Dzisiaj każdy oczekuje, że wraz z zakupionym urządzeniem elektronicznym, będzie można dokonywać jego konfiguracji w o wiele wygodniejszy sposób, czyli poprzez sieć Ethernet a najlepiej Wi-Fi, żeby nie trzeba było podłączać żadnych przewodów. Chyba każdy w tym momencie zgodzi się ze mną. To od razu powoduje, że na myśl przychodzi wybór jakiegoś modułu Wi-Fi, ale nie tylko, bo skoro moduł Wi-Fi, to zapewne zaraz programista (czyli my sami w kolejnym etapie) będzie musiał napisać nie tylko obsługę takiego modułu z poziomu mikrokontrolera ale także będzie musiał zapewnić aplikację sterującą. Gdybym tu napisał, że aplikację na komputer PC, to proszę zauważyć, że to już byłoby w dzisiejszych czasach „prawie” przestarzałe, ponieważ współcześnie dzisiaj liczy się aplikacja na smartfona!

Wystarczy! Mając to wszystko na uwadze już jesteśmy w stanie zapisać sobie w punktach, co musi realizować sterownik naszego urządzenia poza, oczywistą funkcjonalnością zegara. A zatem:

1. Komunikacja Wi-Fi zapewniająca współpracę z aplikacją narzędziową na PC i telefon.
2. Synchronizacja czasu zegara z internetowymi serwerami wzorów czasu NTP.
3. Moduł RTC, czyli zegar czasu rzeczywistego.
4. Odbiornik podczerwieni umożliwiający obsługę urządzenia za pomocą pilota.
5. Czujnik natężenia oświetlenia odpowiedzialny za regulację jasności wyświetlacza (fotorezystor).
6. Obsługę przycisków do manualnych nastaw, gdy zabraknie smartfona lub pilota podczerwieni.
7. Podtrzymywanie baterijne zegara na czas, nawet długotrwałych zaników zasilania (do kilku, kilkunastu dni).
8. Obsługę chociażby podstawowego przetwornika dźwięku typu „buzzer” do celów funkcji budzika/drzemki itp.
9. Mikrokontroler sterujący całym urządzeniem.

W tym momencie zwrócę szczególną uwagę na punkt 6. Otóż istnieje taka niepisana zasada, że interfejs użytkownika jest tym lepszy, im mniej ma przycisków, ponieważ użytkownik zawsze marzy o tym, aby w sytuacji idealnej mieć jeden przycisk,

za którego pomocą będzie mógł zrobić WSZYSTKO! Oczywiście, to tzw. „ideał”, dlatego my skupimy się na tym, aby chociażby zbliżyć się do ideału. Dlatego moja propozycja kolejnych przemyśleń polega na ustaleniu ile oddamy użytkownikowi przycisków i dlaczego będą tylko dwa, a także jakie funkcjonalności będzie można przypisać różnym kombinacjom ich użycia. Oczywiście staramy się dzielić całość na kilka etapów w ten sposób, aby przekonać użytkownika do tego, że:

1. Gdy będzie korzystał z przycisków w urządzeniu, będzie miał jedynie podstawowe (ale i tak spore) możliwości i funkcjonalności.
2. Gdy zechce użyć pilota podczerwieni nastawy staną się o wiele wygodniejsze i zarazem otrzyma dodatkowy pakiet funkcjonalności urządzenia.
3. Jeśli zaś zechce użyć smartfona lub komputera PC, aby posłużyć się aplikacją, otrzyma zarazem największy dostęp do wszystkich funkcjonalności urządzenia a także zdecydowanie maksymalną wygodę wszystkich nastaw.

Mając powyższe na uwadze możemy w końcu przystąpić do rozważań na temat wyboru mikrokontrolera oraz jego minimalnych zasobów sprzętowych. Wziąwszy pod uwagę rozdzielczość naszego wyświetlacza 6×15, gdzie każdy piksel to 3 bajty, łatwo obliczyć, że na potrzeby samego i to pojedynczego bufora ekranu potrzebne nam będzie minimum 270 bajtów pamięci RAM. Idąc dalej i wzięwszy pod uwagę komunikację RS232/UART okaże się szybko, że pewnie zużyjemy podobną ilość pamięci na tzw. cykliczne bufora potrzebne do prawidłowej pracy opartej o zdarzenia. Założmy, że najdłuższy pakiet przesyłany przez UART/WI-FI wyniesie maksymalnie do ok 100 bajtów to już widzimy, że łącznie przyda nam się pamięci RAM na poziomie 500...700 bajtów. Zakładając, że będziemy zmuszeni stworzyć jeszcze bufora potrzebne do przechowywania bieżących nastaw chociażby w obszarze 400...500 bajtów to od razu możemy śmiało stwierdzić, że mikrokontroler mający 1 kB pamięci RAM na pewno nam nie wystarczy. Mając na uwadze rodzinę mikrokontrolerów AVR, w zasadzie „z marszu” odpadają mikrokontrolery z serii ATtiny, gdyż większość z nich nie tylko nie ma odpowiedniej dla nas ilości pamięci RAM, ale także sprzętowych interfejsów UART czy I²C. Zatem musimy sięgnąć po mikrokontrolery z serii ATmega. Nie zależy nam jednak na dużej liczbie wyprawadzeń, więc moglibyśmy sięgnąć po takie, które są oferowane w budowach np. DIP28. Przeglądając na szybko noty katalogowe PDF można wyeliminować od razu takie jak: ATmega8/48/88 czy 168. Wydaje się, że jedynym z tej serii, który może okazać się przydatny

DODATKOWE MATERIAŁY NA FTP:

ftp://ep.com.pl
USER: 95777, PASS: 53wtjyf6

**W ofercie AVT*
 AVT-5606**

Podstawowe informacje:

- 90 „inteligentnych” diod LED RGB (WS2912).
- Sterowanie za pomocą interfejsu szeregowego.
- Zasilanie 5 V DC/6 A.
- Płytkę dwustronna o wymiarach 155 mm×67 mm.

Projekty pokrewne na FTP:

(wymienione artykuły są w całości dostępne na FTP)

AVT-5561	Efektowny sterownik oświetlenia (EP 12/2016)
AVT-1918	Oświetlacz pierścieniowy LED (EP 8/2016)
AVT-1912	Miniaturowy sterownik taśmy LED RGB (EP 7/2016)
AVT-5536	Sterownik taśmy LED ze zdalnym sterowaniem (EP 4/2016)
AVT-1867	Sterownik zasilania taśm LED z wyłącznikiem czasowym (EP 8/2015)
AVT-3133	Sterownik oświetlenia LED sterowany dowolnym pilotem (EdW 4/2015)
AVT-1800	LED Dimmer – regulator oświetlenia LED (EP 5/2014)
AVT-1669	Sterownik oświetlenia LED wewnątrz szafy (EP 3/2012)
AVT-1627	Żarówka LED (EP 8/2011)
AVT-1584	Żarówka LED (EP 8/2010)

*** Uwaga! Elektroniczne zestawy do samodzielnego montażu. Wymagana umiędność lutowania!**
 Podstawową wersją zestawu jest wersja [B] nazywana potocznie KItem (z ang. zestaw). Zestaw w wersji [B] zawiera elementy elektroniczne (w tym [UK] – jeśli występuje w projekcie), które należy samodzielnie wylutować z dołączoną płytką drukowaną (PCB). Wykaz elementów znajduje się w dokumentacji, która jest podlinkowana w opisie kitu. Mając na uwadze różne potrzeby naszych klientów, oferujemy dodatkowe wersje:
 ■ wersja [C] zmontowany, uruchomiony i przetestowany zestaw [B] (elementy wylutowane w płytce PCB)
 ■ wersja [A] płytka drukowana bez elementów i dokumentacja
 Kity w których występuje układ scalony wymagający zaprogramowania, posiadają następujące dodatkowe wersje:
 ■ wersja [A+] płytka drukowana [A] + zaprogramowany układ [UK]
 ■ dokumentacja
 ■ wersja [UK] zaprogramowany układ
 Nie każdy zestaw AVT występuje we wszystkich wersjach! Każda wersja ma załączony ten sam plik pdf! Podczas składania zamówienia upewnij się, którą wersję zamawiasz! <http://sklep.avt.pl>

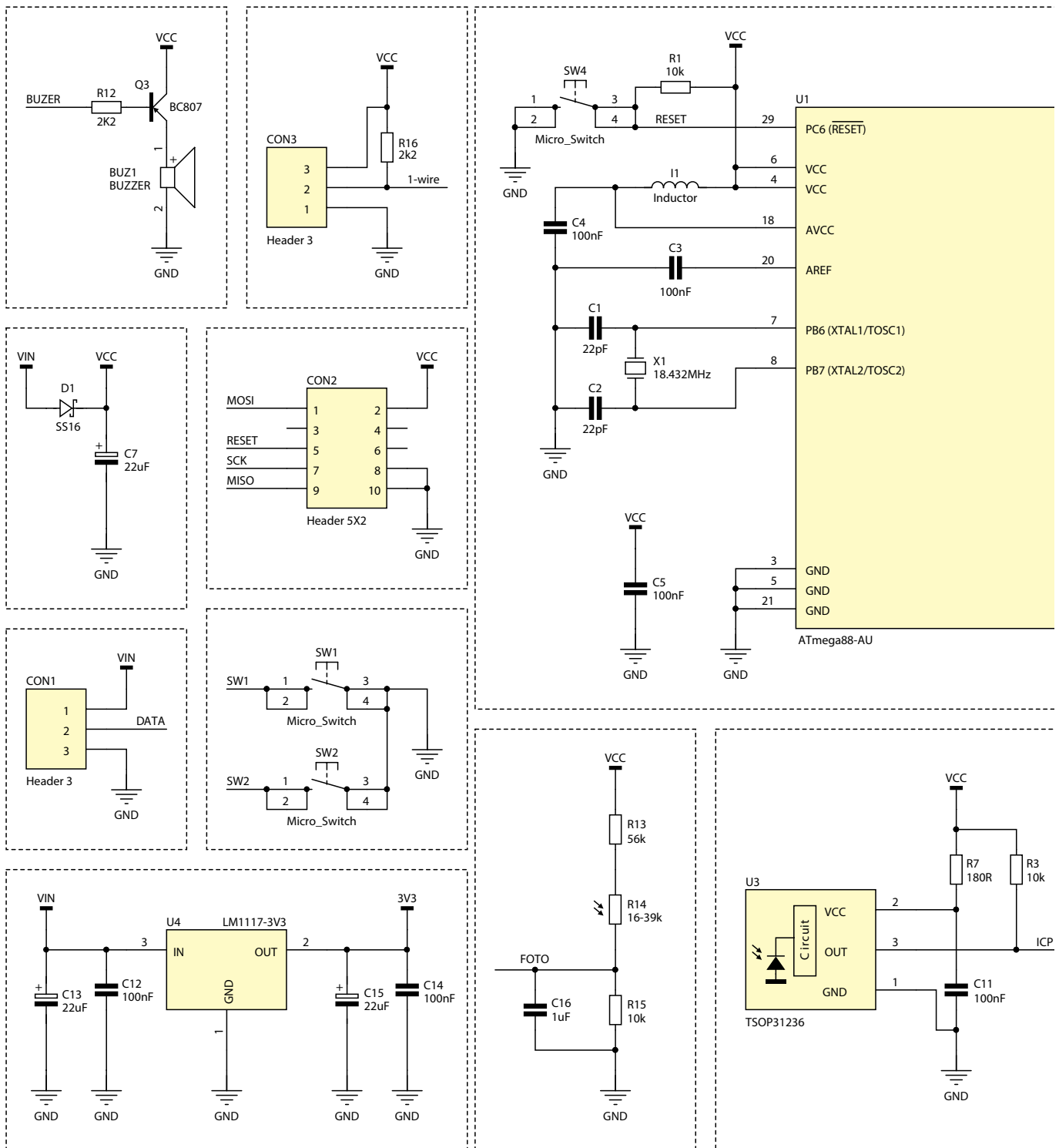
może być ATmega328, mający 32 kB pamięci Flash i 2 kB pamięci RAM! Gdyby ten okazał się niewystarczający dla naszych potrzeb, można by szybko w razie czego przeprogramować kod na dużo większych „braci”, takich jak: ATmega644P albo wręcz monstrualny pod względem dostępnej pamięci RAM w świecie mikrokontrolerów 8-bitowych,

REKLAMA

**Projekty na...
 STM32**

www.stm32.eu

ST **KAMAMI**
 life.augmented



Rysunek 1. Schemat ideowy sterownika Magic Matrix

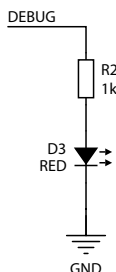
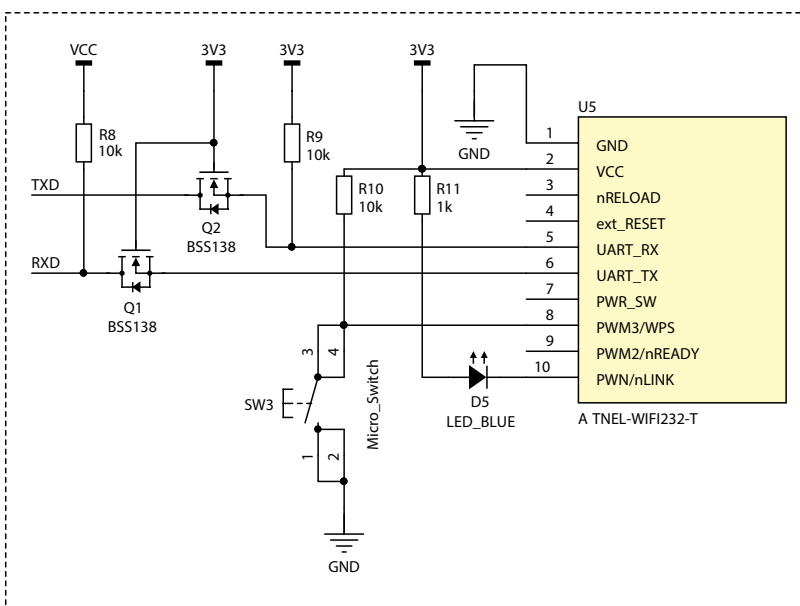
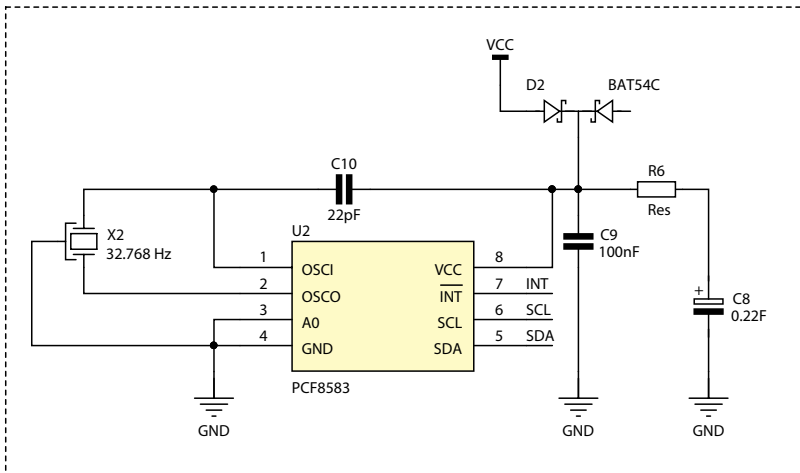
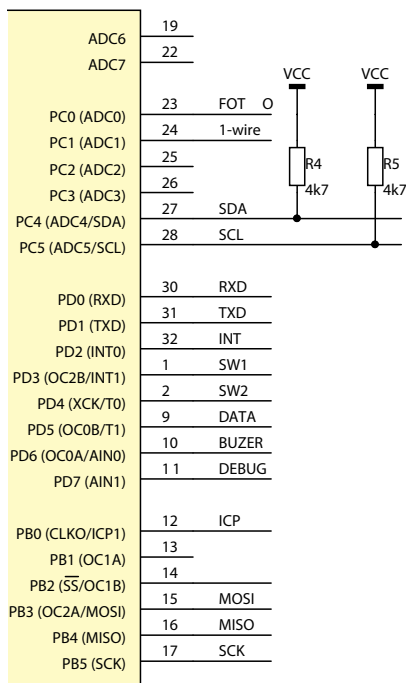
procesor ATmega1284P. Nie dość, że mają one dwa sprzętowe moduły UART, to ATmega644P zapewnia 4 kB pamięci RAM, zaś ATmega1284P aż 16 kB RAM. W tym wypadku śmiało można byłoby powiedzieć, że nie będzie żadnych ograniczeń. Jedyna różnica w związku z ATmega328P to większa obudowa.

W związku z powyższym zaprojektujemy układ w oparciu o ATmega328P, który wydaje się mieć zapas zarówno pamięci RAM jak i Flash w stosunku do naszych wcześniej sprecyzowanych potrzeb w punktach. Możemy zatem przystąpić w końcu do narysowania

długo oczekiwanego schematu naszego projektu. Zgadzam się, że musieliśmy poświęcić sporo czasu na rozważania teoretyczne, ale chciałem pokazać w taki syntetyczny sposób tok rozważań, jaki powinien przyswiecać każdemu konstruktorowi przy wyborze mikrokontrolera. Ja wprawdzie oparłem się o rodzinę mikrokontrolerów 8-bitowych AVR, ale również dobrze podobna analiza mogłaby dotyczyć innej rodziny mikrokontrolerów.

Jak pokazano na **rysunku 1** „sercem” sterownika jest mikrokontroler ATmega328P w obudowie TQFP. Zwrócę uwagę na kilka istotnych szczegółów aplikacji

mikrokontrolera. Z uwagi na fakt planowanej w przyszłości podmiiany firmware naszego sterownika drogą bezprzewodową poprzez moduł Wi-Fi, zastosowałem przycisk SW4 dołączony do nóżki RESET mikrokontrolera. Ma on posłużyć w sytuacjach awaryjnych, gdy z jakichś przyczyn nie powiedzie się aktualizacja i nie uda się nawiązać zdalnej komunikacji z mikrokontrolerem w celu programowego przesłania rozkazu do inicjalizacji bootloadera. Pin RESET jest podciągnięty zasilania (VCC) za pomocą rezystora R1 o rezystancji 10 kΩ. Nie można również zapomnieć o zapewnieniu prawidłowego



filtrowania zasilania mikrokontrolera. Jest ona realizowana za pomocą kondensatorów: C3...C5, C7, C14 i C15. W tym miejscu chciałbym przedstawić ciekawe rozwiązanie zabezpieczenia linii zasilania mikrokontrolera, które jest realizowane przez diodę Schottky D1 oraz kondensator C7 o pojemności 22 μ F. Sama dioda pozwala na maksymalny przepływ prądu do 1 A. Jak już wspominałem we wcześniejszym artykule dotyczącym konstrukcji obudowy i panelu wyświetlacza, zasilanie +5 V pobierane jest z zewnętrznego zasilacza o dużej wydajności prądowej do 8 A. Niestety, z uwagi na to, że zasilacz

jest zewnętrzny zaś w układzie nie zastosowano oddzielnego stabilizatora napięcia na potrzeby mikrokontrolera, można narażać się na spore zakłócenia w torze zasilania wynikające z co najmniej dwóch powodów:

1. Długi przewód doprowadzający zasilanie z napięciem wyjściowym +5 V.
2. Nieregularny i bardzo skokowy pobór prądu przez panel wyświetlacza z diodami WS2812B, które mogą w szczycie pobierać prąd nawet do 5,4 A.

Sam mikrokontroler ze swoimi drobnymi układami peryferyjnymi, o których wspomnę za chwilę, nie będzie miał większego zapotrzebowania na prąd niż maksymalnie 100 mA. To oznacza, że jeśli zabezpieczymy linię zasilania diodą D1 i kondensatorem C7 przed gwałtownymi i skokowymi spadkami napięcia podczas nawet normalnej pracy panelu wyświetlacza, to przy tak małym poborze prądu przez układ mikrokontrolera, zostanie zapewnione wciąż prawidłowy i niezakłócony poziom napięcia zasilania.

W celu zapewnienia odpowiedniej szybkości wykonywania się procedur odpowiedzialnych za obsługę diod WS2812B zastosowano zewnętrzny rezonator kwarcowy o wartości

18,432 MHz. Ta nietypowa być może wartość częstotliwości nie została wybrana w przypadkowy sposób. Ma ona również mocny związek z tym żeby zapewnić prawidłową i niezakłóconą pracę modułu komunikacji UART (RS232) pomiędzy mikrokontrolerem a modułem Wi-Fi w całym zakresie prędkości baudrate. Naturalnie rezonator zaopatrzony jest w dwa kondensatory ceramiczne C1 i C2 o pojemności 22 pF.

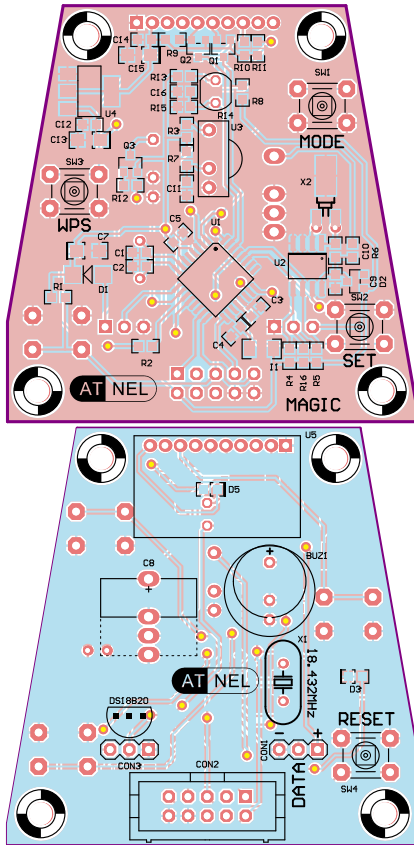
Linie interfejsu I²C podciągnięto do VCC za pomocą rezystorów R4 i R5, każdy

REKLAMA

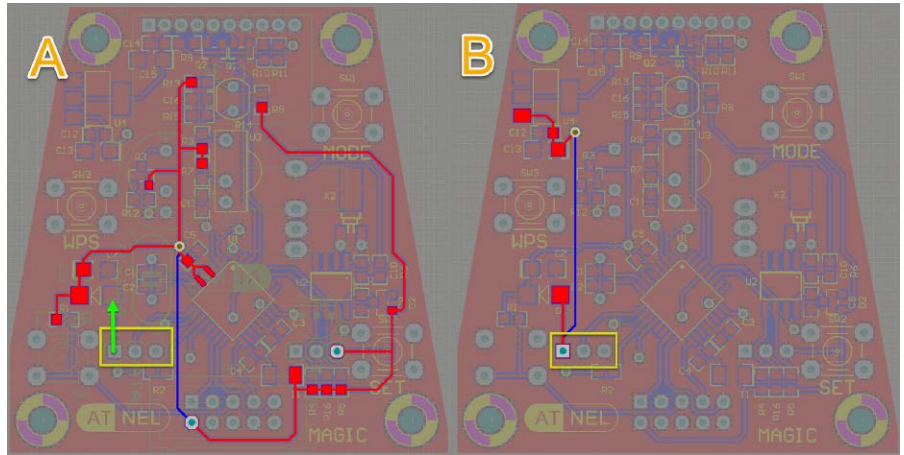
Projekty na 000
STM32
www.stm32.eu
KAMAMI
life.augmented

PROJEKTY

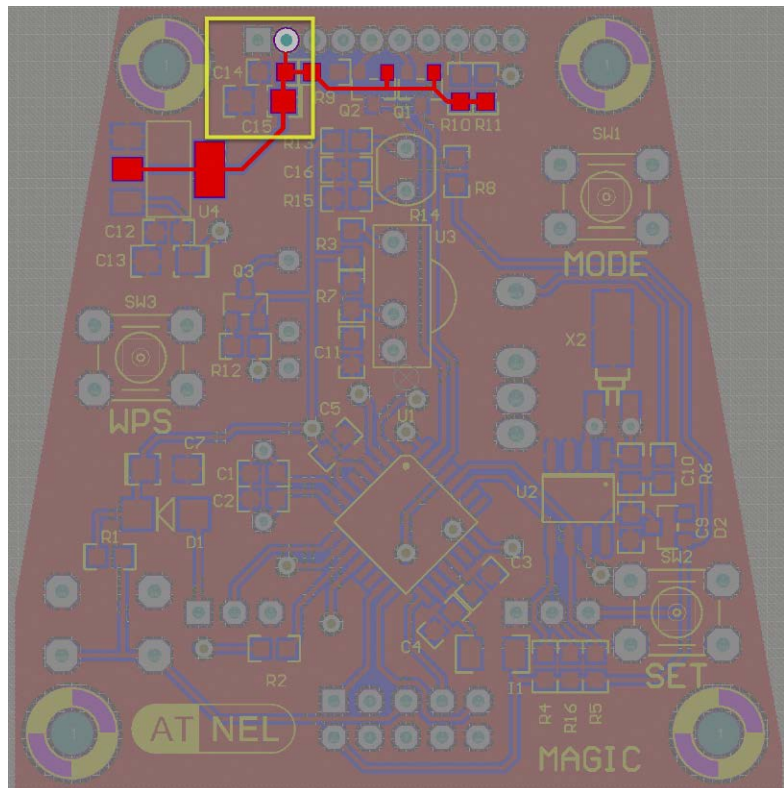
o rezystancji 4,7 kΩ. Wokół mikrokontrolera wydzielono moduły poszczególnych peryferiów dla lepszej analizy schematu. W lewym górnym rogu rys. 1 widać buzzer z generatorem, który jest sterowany z jednej z linii mikrokontrolera za pomocą tranzystora małej mocy BC807 z uwagi na to, że sam buzzer pobiera podczas normalnej pracy nieco większy prąd niż wynosi wydajność prądowa pinu mikrokontrolera. Tym sposobem dbamy również o zmniejszenie zakłóceń na pinach zasilania samego mikrokontrolera, które mogą być przecież generowane również przez



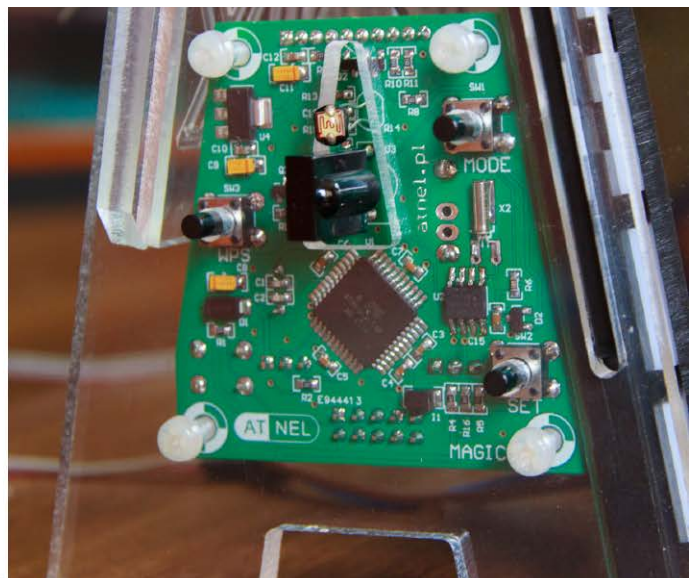
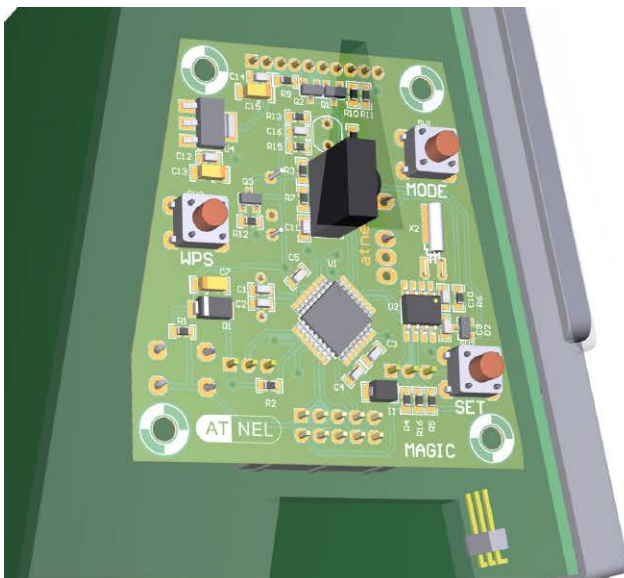
Rysunek 2. Widok projektu PCB płyty sterownika strony Top i Bottom



Rysunek 3. Część A ścieżka zasilania VCC +5 V, część B ścieżka zasilania do Wi-Fi



Rysunek 4. Rozwiązanie zasilania modułu Wi-Fi



Rysunek 5. Widok 3D projektu oraz zdjęcie z fizycznie zamontowanej płyty PCB

sam mikrokontroler, przez jego wewnętrzne układy, a w szczególności wyjściowe porty cyfrowe, które musiałyby przełączać większe obciążenia.

Kolejny moduł to po prostu gniazdo do podłączenia czujnika temperatury DS18B20 pracującego na magistrali 1-wire. W związku z tym zastosowano od razu rezystor R16 o oporności 2,2 k Ω , podciągający linię sygnałową do VCC. W prawym górnym rogu schematu widać zaś moduł RTC. Wykonano go w oparciu o popularny, precyzyjny układ scalony firmy Dallas o symbolu DS1337+. Poza kwarem zegarkowym, zapewniono także zasilanie awaryjne do układu za pomocą niewielkiego kondensatora żelowego C16 o pojemności 0,22 F. W szereg z nim podłączono rezystor R6 o niewielkiej rezystancji 68 Ω po to, aby zabezpieczał kondensator przed zbyt gwałtownym ładowaniem. Samo ładowanie kondensatora a także główne zasilanie układu RTC jest doprowadzone poprzez diodę D2 o symbolu BAT54C. Podczas normalnej pracy napięcie VCC poprzez diodę D2 jednocześnie zasila układ RTC i ładuje kondensator żelowy. W przypadku braku zasilania VCC, dioda D2 odcina zasilanie z kondensatora żelowego do innych układów, dzięki czemu zasilany jest z niego jedynie układ DS1337. **Uwaga!** Diodę D2 dobrano specjalnie do tych potrzeb, ponieważ nie należy wybierać pierwszej z brzegu diody. Wprawdzie każda zapewni tego typu pracę i odciecie zasilania, ale nie każda ma taki sam prąd wsteczny. Dioda BAT54C zapewnia jedynie utratę 2 μ A poprzez swój prąd wsteczny. Inne diody niestety potrafią w tym zakresie mieć dużo gorsze warunki pracy i sporo większy prąd wsteczny, co znacząco wpłynie na skrócenie czasu podtrzymania zasilania dla układu RTC.

Po lewej stronie schematu widać także dwa gniazda oznaczone jako CON1 oraz CON2. O ile gniazdo CON1 jest wykorzystywane zawsze, ponieważ to do niego doprowadzone jest zasilanie +5 V z modułu panelu diod LED a także wyprowadzony jest sygnał sterujący całym łańcuchem diod WS2812B, to złącze CON2 będzie w zasadzie wykorzystane jednokrotnie, tylko do pierwszego zaprogramowania mikrokontrolera wsadem bootloadera. W późniejszym czasie, programowanie kolejnych wersji wsadu będzie się odbywało bezprzewodowo za pomocą modułu Wi-Fi.

Analizując dalej schemat, widać obsadzone dwa przyciski interfejsu użytkownika SW1 i SW2, które zwierają linie mikrokontrolera do masy układu. Poniżej widzimy stabilizator napięcia LM1117-3.3 zapewniający zasilanie do modułu Wi-Fi. Kolejny element to sprzętowa część układu do automatycznej regulacji jasności całego wyświetlacza w zależności o zewnętrznych warunków oświetleniowych. Widać tutaj

dzielnik rezystorowy składający się kolejno z rezystorów: R13, R14 oraz R15, gdzie R14 jest fotorezystorem. Dzielnik został dobrany tak, że przy maksymalnym oświetleniu, gdy fotorezystor posiada minimalną rezystancję, na pin ADC mikrokontrolera oznaczony FOTO jest podawane napięcie ok 1,1 V, co jest związane z tym, że mikrokontroler ma wewnętrzne źródło napięcia odniesienia na tym samym poziomie, czyli 1,1 V. Dzięki czemu można będzie używać pełnej rozdzielczości przetwornika A/C. Układ ten zaopatrzone również nieprzypadkowo w kondensator ceramiczny C16 o sporej pojemności (1 μ F). Ma on na celu zapewnić już na etapie sprzętowym lepsze uśrednianie pomiarów. Wprawdzie jego pojemność może wpływać na spowolnienie zmian napięcia przy gwałtownych zmianach oświetlenia, ale nam właśnie na tym zależy. Pozostałe uśrednianie pomiarów A/C będzie już wykonywane programowo w mikrokontrolerze.

Następny moduł stanowi bodajże najbardziej popularny scalony odbiornik podczerwieni TSOP31236, którego częstotliwość pracy wynosi 36 kHz, a zatem jest przystosowana najlepiej do potrzeb pilotów podczerwieni pracujących w standardzie Philipsa RC5. Układ ma zapewnione niezależne filtrowanie zasilania, na które składa się rezystor R7 (180 Ω) i kondensator C11 (100 nF). Jest to bardzo ważne, a niestety często pomijane przez wielu konstruktorów. Brak tej pary elementów ma decydujący wpływ na zmniejszenie zakłóceń w przekazywanym do mikrokontrolera sygnale. Wyjście odbiornika jest podciągnięte rezystorem R13 (10 k Ω) i doprowadzone do wejścia ICP1 mikrokontrolera. Nie mogło też zabraknąć diody LED, która nie pełni tutaj (przynajmniej w trakcie prac i uruchamiania całego sterownika) funkcji dekoracyjnej. Jest to tzw. dioda DEBUB LED. Służy ona jako podstawowe narzędzie do debugowania programu w fazie testów.

Ostatni moduł to moduł Wi-Fi. Pełni on najważniejszą rolę po mikrokontrolerze. Zapewnia możliwość zdalnego i bezprzewodowego programowania mikrokontrolera oraz odpowiada za komunikację nie tylko z aplikacjami na komputerze PC czy smartfonie. Układ potrafi również automatycznie pobierać precyzyjny czas z wzorców czasu w Internecie, z serwerów NTP i to za jego pomocą będzie synchronizowany czas w układzie RTC. Jeśli zatem nawet urządzenie pozostanie bez zasilania na tyle długo, że rozładuje się kondensator żelowy i wyzeruje się układ RTC, to po starcie urządzenia, program pobierze dzięki modułowi WI-FI aktualny czas z Internetu i ustawi automatycznie datę oraz godzinę w naszym zegarze. Poza tym, synchronizacja z serwerem czasu NTP będzie przewidziana co najmniej raz na dobę albo nawet raz na godzinę. Dzięki

temu już nigdy nie spóźnimy się do pracy lub do szkoły.

Moduł Wi-Fi zaopatrzone w przycisk SW3, który zapewni funkcjonalność WPS. Ta technologia zapewnia bardzo szybką możliwość podłączenia się modułu do routera Wi-Fi użytkownika urządzenia, o ile router również wspiera technologię WPS. Polega to na tym, że nie trzeba ręcznie konfigurować parametrów routera, aby się do niego przyłączyć. Wystarczy wcisnąć przycisk WPS na routerze a następnie SW3 w naszym urządzeniu, a moduł Wi-Fi automatycznie pobierze parametry pracy routera, czyli jego SSID oraz hasło, dzięki czemu automatycznie włączy się do sieci.

Do modułu jest dołączona niebieska dioda LED (D5), która świeci, jeśli moduł poprawnie przyłączy się do sieci Wi-Fi. Z uwagi na fakt, że moduł Wi-Fi zasilany jest napięciem +3,3 V, a mikrokontroler jest zasilany napięciem +5 V, należy bezwzględnie zadbać o dopasowanie poziomów napięcia na liniach komunikacji UART (RS232). Służy do tego niezawodny układ oparty zaledwie o dwa popularne tranzystory małej mocy MOSFET-N o symbolu BSS138. Oznaczono je na schemacie Q1 i Q2. Wraz z nimi w konwersji napięć biorą udział jeszcze dwa rezystory R8 i R9, oba o rezystancji 10 k Ω .

Skoro mamy już dokładnie omówiony schemat urządzenia i wiemy jak wszystko powinno działać, pozostało teraz tylko omówić projekt płyty PCB. Jest to kolejny i to bardzo ważny etap przygotowania projektu, ponieważ często od jakości płytki drukowanej zależy w ogóle powodzenie działania całości, a przynajmniej stopnia niezawodności działania. Dlatego trzeba zadbać odpowiednio i przede wszystkim o prawidłowe rozprowadzenie zasilania ze szczególnym uwzględnieniem torów VCC, a także pół masy. Poza tym, niezmiernie ważne jest także zadbanie o prawidłowe umiejscowienie kondensatorów filtrujących zasilanie czy chociażby samego rezonatora kwarcowego. Przedstawię zatem kilka rysunków poglądowych jak może wyglądać prawidłowo zaprojektowana płyta PCB do naszego sterownika

REKLAMA



Projekty na...
STM32

www.stm32.eu

life.augmented

KAMAMI



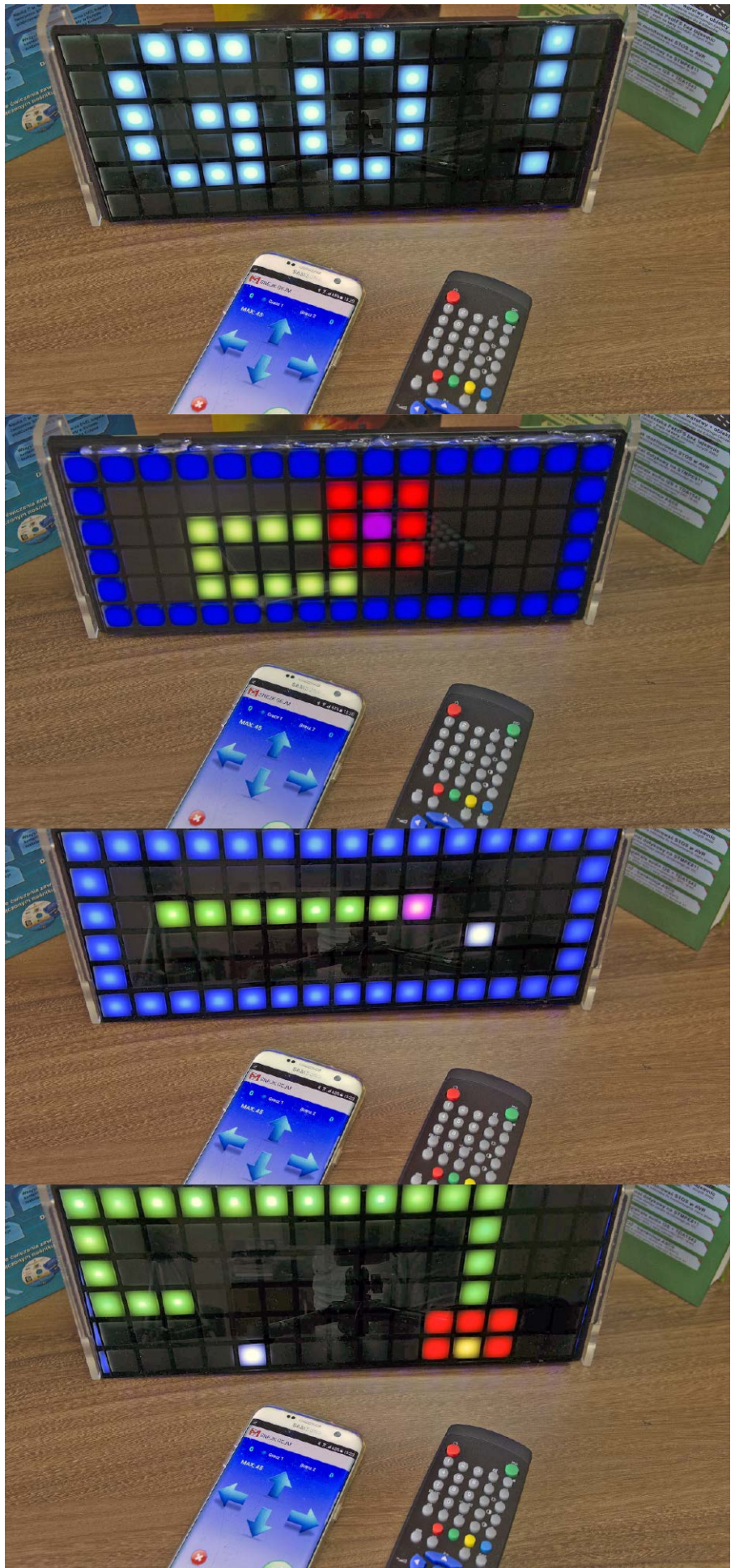
Rysunek 6. Poglądowy widok zastosowanego pilota RC5

z podziałem widoków na szczególnie istotne elementy projektu PCB.

Na **rysunku 3** pokazano projekt płytki drukowanej z zaznaczonym projektem ścieżek zasilania. W części „A” pokazano obwód zasilania VCC, który dzieli się na trzy gałęzie. Jedna doprowadza zasilanie VCC w górę płytki do odbiornika podczerwieni i buzera, druga gałąź (na środku) jest przeznaczona tylko na potrzeby mikrokontrolera. Przy czym tuż przed samymi pinami zasilania mikrokontrolera osadzony jest kondensator filtrujący zasilanie 10 μ F. Trzecia gałąź biegnąca w dół od diody D1, dociera poprzez złącze programatora KANDA, do układu RTC oraz fotorezystora. Po lewej stronie rysunku, w części „B” pokazano ścieżkę zasilania prowadzoną oddzielnie do stabilizatora 3,3 V, który służy jedynie na potrzeby zasilania modułu Wi-Fi.

Na szczególną uwagę zasługuje przedstawienie zasilania modułu Wi-Fi. Na **rysunku 4** w żółtej ramce wyraźnie zaznaczono jak blisko pinów zasilania modułu Wi-Fi umieszczono kondensatory C14 oraz C15. Szczególną rolę odgrywa tutaj kondensator C15 o pojemności 22 μ F. **Rysunek 5** przedstawia widok 3D z etapu projektu w porównaniu do zdjęcia fizycznie działającego prototypu zegara.

Na zakończenie opis działania przycisków interfejsu użytkownika, mowa o przyciskach noszących nazwy: MODE oraz SET. W wypadku pilota podczerwieni pracującego w standardzie RC5 (Philips), o ile ma on klawiaturę numeryczną, użytkownik otrzymuje nieco rozszerzone możliwości obsługi ustawień urządzenia. Przede wszystkim klawiatura numeryczna pozwala na dużo wygodniejsze wprowadzanie wartości cyfr dla ustawianego czasu, daty lub godziny budzika (alarmu). Z uwagi na dużo większą



Fotografia 7. Gra SNAKE napisana dla wyświetlacza MagixMatrix

Tabela 1. Funkcjonalności wszystkich trybów pracy przetłączników MODE i SET

Kombinacja wciśniętych przycisków	Opis funkcji
MODE (krótkie kliknięcie)	Cykliczne przetaczanie trybu RAINBOW CLOCK i domyślnych kolorów (fabrycznych bądź użytkownika).
MODE (długie kliknięcie)	Rozpoczęcie trybu ręcznego ustawiania daty i czasu w przypadku, jeśli nie obsadzony jest moduł Wi-Fi (brak automatycznej synchronizacji czasu).
MODE (krótkie kliknięcie) gdy włączony budzik	Natychmiastowe wyłączenie alarmu oraz trybu tzw DRZEMKI.
SET (krótkie kliknięcie)	Natychmiastowe wyświetlenie temperatury a następnie daty oraz statusu budzika a także nieodebranych wiadomości SMS na telefonie.
SET (długie kliknięcie)	Rozpoczęcie trybu ręcznego ustawiania czasu budzika.
SET (krótkie kliknięcie), gdy włączony budzik	Natychmiastowe wyłączenie alarmu i przejście w tryb DRZEMKI, kolejny raz budzik załączy się za kilka minut.
MODE + SET (krótkie kliknięcie obu przycisków)	Cykliczne przetaczanie globalnej aktywności budzika.
MODE (wciśnięty podczas włączania zasilania)	Przywrócenie ustawień fabrycznych urządzenia, kolory, alarm stan maksymalnej punktacji w grach użytkownika.

liczbę dostępnych przycisków użytkownik może:

- Wymuszać synchronizację czasu z internetowym serwerem czasu NTP.
- Przelącać tryb automatycznej i manualnej kontroli jasności wyświetlacza.
- Ustawiać własny poziom jasności wyświetlacza po przełączeniu kontroli jasności.
- Zrestartować urządzenie w dowolnym momencie np. w celach diagnostycznych.

Kompletny wykaz zaimplementowanych funkcjonalności zamieszczono w tabeli 1.

Na zakończenie przedstawię kilka fotografii z trybu gry „SNAKE”, która została zaimplementowana w urządzeniu (fotografia 7). Życzę dobrej zabawy podczas konstruowania i programowania prezentowanego urządzenia.

Mirosław Kardaś
biuro@atnel.pl

REKLAMA

m.technik

Ciekawi świata są zawsze młodzi

w prezencie na każdą okazję



<https://goo.gl/TiDLmR>

Welcome to the Machine – śpiewał przed laty zespół Pink Floyd i oto jak za dotknięciem czarodziejskiej różdżki jesteśmy już prawie w świecie, nad którym będzie mógł zawisnąć taki właśnie szylt. To, co w utworze słynnej grupy było raczej metaforą relacji międzyludzkich i społecznych, już niedługo spełni się w znaczeniu technicznym, a jednocześnie w pełni „dotykalnym”, jako coraz bliższy, niemal intymny, codzienny kontakt *homo sapiens* z robotem, komputerem, sztuczną inteligencją. „Maszyna sterowana myślą i odwrotnie”. „Maszyny ludziom, ludzie maszynom”. „Czy maszyny przedłużą nam życie?” „Jak żyć z nie-człowiekiem u boku?” To nie rozdziały z książek Lema czy Snerga, ale tytuły czterech najnowszych Tematów numeru Młodego Technika.



RAPORT BROŃ – PERSPEKTYWA 2040

Połączenie bezpośrednie mózgu z superwydajnym komputerem, mózg online, czyli łączność myślami z Internetem i vice versa, to kolejny etap w ewolucji człowieka. Już się zaczyna

przejrzysz i kupisz na
www.ulubionykiosk.pl