

Jakub PRZYBYLSKI*, Paweł IDZIAK*

PRYZRĄD DO POMIARU PRĄDU I NAPIĘCIA STAŁEGO ZBUDOWANY NA SPRZĘŻONYCH PLATFORMACH ARDUINO I RASPBERRY Pi 3

W artykule przedstawiono koncepcję oraz zrealizowany model fizyczny przyrządu do pomiaru napięcia i prądu stałego. Miernik zbudowano sprzęgając mikrokomputer serii Raspberry Pi z mikrokontrolerem serii Arduino. Opracowano aplikacje umożliwiające zaimplementowanie autorskich programów do obu urządzeń. Przyrząd został wyposażony w rezystancyjny ekran dotykowy pozwalający użytkownikowi decydować o trybie przeprowadzania pomiarów (automatyczny czy manualny) oraz o sposobie archiwizacji i wizualizacji efektów pomiarów. W aplikacji wykorzystano tzw. kompilację skrośną, dzięki której możliwe jest efektywne sprzęgnięcie tak odrębnych struktur jakim są układy Raspberry Pi i Arduino.

SŁOWA KLUCZOWE: multimetr, Raspberry Pi, Arduino, kompilacja skrośna.

1. WPROWADZENIE

Postępująca cyfryzacja, praktycznie wszystkich dziedzin życia, wymusza poszukiwanie takich technik pomiaru różnych wielkości fizycznych, aby rezultaty pomiarów można było:

- przetwarzać w postaci cyfrowej do formy umożliwiającej ich bezpośrednią lub odroczoną w czasie wizualizację,
- transformować (zamieniać), jeśli jest to wymagane, „zapamiętane” sygnały cyfrowe do postaci analogowej,
- w efektywny sposób archiwizować.

Wymagania takie prowadzą do tworzenia takich układów pomiarowych, w których pomiary różnych wielkości fizycznych sprowadzają się do przetworzenia mierzonych wielkości na sygnał elektryczny. Najczęściej sygnałem tym jest sygnał napięciowy, rzadziej sygnał prądowy. Nowoczesne przyrządy pomiarowe umożliwiają zazwyczaj pomiar kilku różnych wielkości fizycznych np. prądu, napięcia, temperatury, częstotliwości, bywa, że mocy, pojemności kondensatora, rezystancji itd. W przypadku mierników analogowych wielkościami bezpośrednio podlegającymi pomiarowi są prąd, napięcie lub jedno i drugie.

* Politechnika Poznańska

W miernikach cyfrowych wielkością podlegającą bezpośredniemu pomiarowi jest napięcie. Wtedy charakter zmian sygnału mierzonego jako funkcji czasu nie ma zazwyczaj znaczenia. W zależności od zasady pomiaru i konstrukcji rezultaty pomiaru są udostępniane operatorowi albo w postaci analogowej tzn. jako wychylenie wskazówki na tle odpowiednio opisanej skali, albo w postaci cyfrowej tzn. jako ciąg cyfr i znaków separacyjnych wyświetlanych na ekranie. W miernikach cyfrowych zazwyczaj na ekranie pojawia się też informacja o jednostce mierzonej wielkości fizycznej i charakterze jej zmian w czasie (sygnał stały - DC, sygnał zmienny lub przemienny w czasie - AC, sygnał przemienny ze składową stałą).

Przyrządy zezwalające na pomiary kilku różnych wielkości fizycznych określone są często wspólnym mianem multimetrów lub przyrządów uniwersalnych. W przypadku multimetru cyfrowego pomiar dokonywany jest w dyskretnych odstępach czasu i prezentowany z pewnym opóźnieniem czasowym. Czas prezentacji rezultatu pomiaru wynosi zazwyczaj 0,8-1,0 sek. Po takim czasie następuje aktualizacja wyświetlanej wartości [1]. Typowy multimetr wyposażony jest w dwie sondy łączone szeregowo lub równolegle z badanym obwodem elektrycznym. Elementy regulacyjne umieszczone na płycie czołowej multimetru umożliwiają wybór trybu i zakresu mierzonych wielkości.

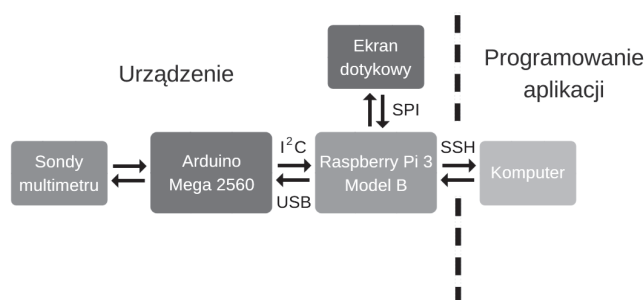
2. KONSTRUKCJA INTELIGENTNEGO MULTIMETRU

Prezentowany inteligentny multimetr zbudowany został z wykorzystaniem platform Raspberry Pi i Arduino. W swej podstawowej wersji służy on do pomiaru napięcia lub natężenia prądu stałego w jednym z dostępnych trybów. Dwa z nich zapewniają automatyczne wykonywanie pomiaru kontrolowanej wartości i zarejestrowanie jej. Użytkownik musi zadeklarować liczbę planowanych pomiarów – wykrywanie zmian wartości i „przechwytywanie” ich jest zadaniem inteligentnego multimetru. Do obsługi i wyświetlania wartości pomiarów posłuży ekran dotykowy. Pomierzone wartości zapisywane są na karcie SD.

Realizacja koncepcji multimetru obsługiwanego poprzez ekran dotykowy, wymaga zastosowania dostatecznie wydajnej pod względem obliczeniowym platformy. Takie możliwości wykazują mikrokomputery Raspberry Pi z systemem operacyjnym Raspbian. Niestety, żaden z modeli tego urządzenia nie posiada wejść analogowych, niezbędnych do realizacji przewidywanych algorytmów. W opracowanej konstrukcji postanowiono oddzielić funkcje pomiarowe od obsługiwanego ekranu dotykowego.

Do pomiarów napięcia i natężenia prądu stałego wymagany jest, zdolny do komunikacji z platformą Raspberry Pi mikrokontroler, obsługujący wejścia analogowe. Do tego celu wykorzystano układ z rodziny Arduino. Konstrukcje te

posiadają magistrale komunikacyjne I²C i USB, akceptowane przez platformę Raspberry. Schemat blokowy takiego złożenia prezentuje rysunek 1.



Rys. 1. Schemat koncepcji inteligentnego multimetru

Układ multimetru zbudowano z wykorzystaniem Arduino Mega 2560. Przewidziano, że podłączone do niego będą dwie pary sond pomiarowych: jedna odpowiedzialna za pomiary napięcia, druga za pomiary natężenia prądu. Projektując architekturę układu wykorzystującego dwie sprzężone platformy szczególną uwagę zwrócono na ich niezawodną, wzajemną komunikację. Do sterowania trybami pomiarowymi i przesyłania wartości mierzonych posłużą dwa różne interfejsy komunikacyjne. W przypadku zastosowania tylko jednego interfejsu musiałby on przerywać przesyłanie wartości pomiarowych, aby zmienić tryb pracy układu. Mogłoby to skutkować problemami na etapie programowania i rozwijania aplikacji, a także niepożądanymi, mylnymi pomiarami w trakcie pracy urządzenia.

Oprócz komunikacji obu platform w celu przesyłania wartości pomiarów i ustalania bieżącego trybu pracy, potrzebny jest dodatkowy interfejs do obsługi ekranu dotykowego. Zastosowany ekran dotykowy wykorzystuje technologię SPI (ang. Serial Peripheral Interface) do dwukierunkowego przesyłania danych pomiędzy wyświetlaczem i minikomputerem.

Raspberry Pi zapewnia wyświetlanie obrazu na wyłącznie jednym wyjściu w danej chwili. To oznacza, iż nie istnieje możliwość uruchomienia aplikacji na miniaturowym wyświetlaczu dotykowym przy jednoczesnej obsłudze systemu Raspbian za pośrednictwem obrazu z portu HDMI. Z tego względu, na potrzeby rozwijania aplikacji, zdalny dostęp do plików minikomputera, zapewniono poprzez protokół komunikacyjny SSH (ang. Secure Shell), a aplikację projektowano z poziomu komputera PC z wykorzystaniem kompilacji skrośnej.

To oznacza, iż wykorzystując miniaturowy wyświetlacz jako docelową platformę do obsługi aplikacji, nie istniałaby możliwość przeprowadzania operacji uruchamiania aplikacji z podglądem jej przebiegu na monitorze wykorzystującym port HDMI. W prezentowanym rozwiązaniu zdalny dostęp do plików mini-

komputera, zapewniono poprzez protokół komunikacyjny SSH (ang. Secure Shell), za pośrednictwem kompilacji skrośnej.

2.1. Struktura sprzętowa multimetru

W prezentowanym multimetrze jako jednostkę nadrzędną (sterującą) zastosowano mikrokomputer Raspberry w wersji Pi 3 z czterordzeniowym procesorem typu Broadcom BCM2837, taktowanym z częstotliwością 4 x 1,2 GHz, pamięcią RAM 1 GB, portami USB, wyjściem typu HDMI [2].

Zakładany kompaktowy charakter konstrukcji inteligentnego multimetru wymusił konieczność zastosowania - do obsługi urządzenia - dotykowego wyświetlacza LCD firmy WaveShare. Pozwala to zrezygnować z korzystania klawiatury i myszy do obsługi przyrządu.

Nawigacja w interfejsie programu odbywa się dotykowo. Ekran jest ekranem rezystancyjnym, co oznacza możliwość użycia tzw. rysika do wskazywania wybranego miejsca na ekranie. Wykorzystany moduł jest ekranem typu TFT LCD, cechujący się niewielką grubością matrycy o rozdzielczości 320 na 480 pikseli. Przy przekątnej 3,5 cala jest to kompromis pomiędzy jakością obrazu, a ilością pikseli obliczanych w danej chwili. Próby z ekranami o większej rozdzielczości wykazały, że jej zwiększenie w istotny sposób obniża płynność działania aplikacji multimetru [6].

Jako jednostkę pomiarowo-transponującą wybrano moduł Arduino Mega 2560 z mikrokontrolerem ATmega2560, wyposażonym w 256 KB pamięci Flash, 4 KB pamięci EEPROM oraz 8 KB pamięci SRAM. Wybrany model wyposażony jest w 16 wejść analogowych i 54 cyfrowe porty wejścia/wyjścia. Układ wyposażony jest również w port USB typu B i obsługuje magistralę I²C, która wykorzystana została w projekcie do komunikacji z jednostką sterującą Raspberry Pi 3 [3].

2.2. Metodyka pomiarów napięcia i natężenia prądu stałego

Odczyty z wejścia analogowego Arduino przyjmują wartość od 0 do 1023, zależnie od bieżącego napięcia na wejściu, które maksymalnie wynosić może 5 V. Za pośrednictwem rezystancyjnego dzielnika napięcia możliwe jest wprowadzenie do układu napięcia kilkukrotnie wyższego. Dokładność pomiaru uzależniona jest od precyzji wykonania wspomnianego dzielnika. W ten sposób, możliwe jest potraktowanie napięcia wejściowego U_{we} jako napięcia do pomiaru, a napięcia uzyskanego z dzielnika (napięcie U_{wy}) jako napięcia występującego pomiędzy wejściem analogowym Arduino a masą tego modułu.

Funkcję amperomierza zrealizowano za pomocą gotowego modułu czujnika natężenia prądu. Dostępny na rynku moduł pomiarowy typu ACS712 pozwala

na pomiar prądu w zakresach 5, 20 i 30A, zależnie od wybranego wariantu [7]. W prezentowanym rozwiązaniu wykorzystano moduł o zakresie do 30 A.

2.3. Metody komunikacji pomiędzy jednostką nadrzędną i pomiarową

Raspberry Pi posiada kilka interfejsów komunikacyjnych, które mogą pośredniczyć w wymianie danych pomiędzy peryferiami komputera takimi jak mikrokontrolery czy też zewnętrzne wyświetlacze. W projekcie, interfejs szeregowy SPI (Serial Peripheral Interface) jest wykorzystywany przez wyświetlacz dotykowy. Działanie tego interfejsu w hierarchii master-slave jako łącze „full duplex” zapewnia równorzędną komunikację w obu kierunkach jednocześnie tj. z jednostki do wyświetlacza i odwrotnie. Pozwala to na jednoczesną transmisję obrazu i odbioru danych związanych z pobudzeniem ekranu przez użytkownika.

W projekcie multimetru, magistrala I²C jest odpowiedzialna za wymianę informacji związanych z bieżącym trybem pomiarów. Ze względu na sporadyczne, ale występujące, błędy w odczytach i problematyczny charakter przesyłania więcej niż jednego bajta danych przy zastosowaniu bibliotek WiringPi, do transmisji bieżących wartości pomiarów posłużyła komunikacja szeregową. W połączeniu tym zastosowany został konwerter poziomów logicznych. Nie jest to wymagane dla Raspberry Pi pracującego w trybie „master” i Arduino w trybie „slave”, gdyż Raspberry Pi posiada odpowiednie rezystory podłączone do linii SDA oraz SCL [2]. Zastosowanie konwertera poziomów logicznych zwiększa jednak bezpieczeństwo połączenia, a także redukuje niebezpieczeństwo błędów podczas transmisji.

Sieciowy protokół komunikacyjny SSH (Secure Shell) umożliwia uzyskanie zdalnego dostępu do innego komputera w obrębie tej samej sieci. Zaletą wymiany informacji tą drogą są duże prędkości – ograniczane zaledwie przez parametry połączenia sieciowego klienta oraz serwera [4]. Ponieważ system Raspbian, oparty na architekturze systemu Debian, posiada powłokę systemową działającą w trybie tekstowym, obsługiwaną za pomocą komend, dzięki SSH, można zinterpretować jednostkę Raspberry Pi jako komputer w roli serwera i podłączyć do niego inny komputer jako klienta. Do zrealizowania połączenia wystarczy, aby zarówno klient jak i serwer byli podłączeni do tej samej sieci.

W opracowanym multimetrze SSH ma dwa zastosowania. Pierwsze z nich to obsługiwanie z poziomu komputera PC systemu Raspbian w procesie opracowywania i testowania aplikacji. Jest to konieczność wynikająca z podłączenia do Raspberry Pi miniaturowego wyświetlacza, przeznaczonego do wyświetlania aplikacji multimetru. Obsługa systemu Raspbian przy niskiej rozdzielczości i działającej niekiedy pełnoekranowej aplikacji multimetru byłaby wyjątkowo utrudniona i czasochłonna, zwłaszcza przy testowaniu układu. Drugie zastoso-

wanie to możliwość wykorzystania bibliotek Qt, które posłużyły do opracowania aplikacji multimetru. Zastosowana kompilacja skrośna wymaga nawiązania połączenia SSH pomiędzy Raspberry Pi a komputerem PC.

Powszechnie stosowane łącze USB (Universal Serial Bus) posłużyło do komunikacji Raspberry Pi z Arduino w ramach biblioteki Wire [8].

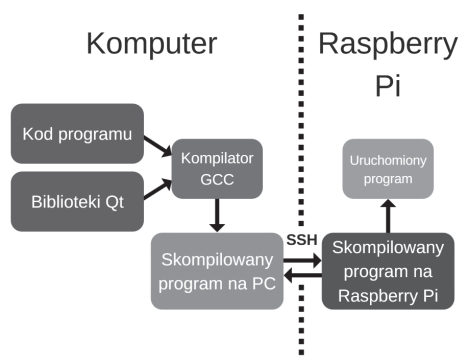
W konstrukcji inteligentnego multimetru, dane w transmisji szeregowej zawierają informacje o bieżących pomiarach woltomierza i amperomierza. Duże prędkości transmisji pozwolą na zadowalająco szybkie działanie trybów pomiarowych, przy jednoczesnym ograniczeniu zastosowania magistrali I²C do sterowania trybami pracy bez przesyłania pomiarów. Jednokierunkowy charakter działania w I²C stanowiłby problem. Transmisja pomiarów musiałaby być przerywana za każdym razem w razie przesłania danych sterujących trybem pracy multimetru.

Przy opracowywaniu aplikacji do obsługi multimetru poprzez Raspberry Pi, wykorzystano zasoby programistycznych bibliotek Qt. Środowisko to stanowi zestaw bibliotek oraz narzędzi do projektowania aplikacji kompatybilnych z innymi platformami. Korzystając z tego środowiska poprzez elementy interfejsu graficznego takie jak przycisk można przysyłać sygnały z informacjami o zachodzącym zdarzeniu. W ten sposób, projektant aplikacji programuje funkcje dedykowane określonym interakcjom użytkownika z programem, np. funkcję reagującą wyświetleniem nowego okna na wypełnienie pola tekstowego określonym ciągiem znaków.

2.4. Kompilacja skrośna

Kompilacja skrośna wspierana przez środowisko Qt umożliwia zaprogramowanie i skompilowanie aplikacji w architekturze innej niż architektura platformy docelowej. Pozwala to na projektowanie aplikacji z poziomu komputera stacjonarnego dysponującego większą wydajnością oraz na ograniczenie udziału mniej wydajnej przenośnej platformy docelowej do testowania działania kodu programu.

Kod programu wraz z bibliotekami Qt zostaje skompilowany przez GCC (ang. GNU Compiler Collection), a następnie przesłany poprzez SSH do Raspberry Pi. Możliwa jest co prawda instalacja bibliotek Qt wraz z programem narzędziowym Qt Creator w systemie Raspbian, ale proces konfiguracji (realizowany częściowo w Terminalu) trwa wtedy bardzo długo. Biblioteki Qt wspierające Raspberry Pi są niedopracowane w przypadku systemu Windows. Przeprowadzenie kompilacji skrośnej udaje się tylko dla modeli Raspberry Pi 1 oraz 2. Dlatego w procesie projektowania aplikacji Qt dla Raspberry Pi wykorzystano narzędzia z systemu Linuksa - Ubuntu [5].



Rys. 4. Schemat kompilacji skróśnej

Kluczowe kroki w skonfigurowaniu kompilacji skróśnej to zbudowanie tzw. „Qt Base” zawierającego wszystkie podstawowe biblioteki i wgranie do urządzenia. W tym celu należy najpierw pobrać projekt z Qt Base, a następnie uruchomić skrypt konfiguracyjny kompilację skróśną. Proces ten wymaga, w przypadku komputera klasy PC, kilkanaście minut. Pozostałe operacje w Terminalu trwają do kilku godzin, zależnie od możliwości sprzętowych komputera.

3. FUNKCJE PROGRAMU STERUJĄCEGO

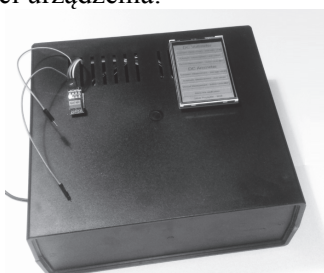
Aplikacja obsługująca multimetr działa w trybie pełnoekranowym i uruchamia się wraz z załadowaniem systemu Raspbian. Biblioteka WiringPi zapewnia przesłanie, z Raspberry Pi do Arduino za pośrednictwem magistrali I²C, zmiennej `x` o wartości z zakresu od 0 do 255 [2, 5]. Zmienna ta odpowiada za przedstawienie Arduino w żądany tryb pracy. Jeśli zmienna `x` przyjmuje wartości od 0 do 50 wówczas następuje automatyczne (seria pomiarów) wykonywanie pomiarów napięcia; wartością zapamiętaną jest wartość średnia obliczona spośród pięciu zmierzonych w przeciągu jednej sekundy. Jeżeli zmienna ta ma wartość z zakresu od 51 do 100 – układ realizuje automatyczną serię pomiarów napięcia, rejestruje maksymalną wartość spośród zmierzonych w ciągu 1 sek. Gdy wartość zmiennej `x` zawiera się w przedziale 101 do 150 następuje automatyczne wykonanie serii pomiarów natężenia prądu, a wartością zapamiętaną jest średnia obliczona spośród pięciu wartości zmierzonych w przeciągu jednej sekundy. Dla zakresu 151 do 200 zachodzi automatyczny pomiar wartości natężenia prądu; zapamiętana zostaje największa wartość bezwzględna spośród pięciu zmierzonych w przeciągu jednej sekundy. Przesłanie wartości 251 przywołuje ręczne pomiary napięcia i zapisanie bieżącej wartości po naciśnięciu przycisku, a wartość 252 wywołuje ręczne pomiary natężenia prądu i zapisanie bieżącej wartości po naciśnięciu przycisku.

Automatyczne serie pomiarów odbywają się w kilku krokach. Pierwszy to wybranie zaplanowanej przez użytkownika liczby pomiarów, w zakresie od 1 do 50; drugi to wykonanie pomiarów (pomiar zostaje zrealizowany po przekroczeniu ustalonego progu wartości tzn. 0,6 V w przypadku napięcia i 0,6 A w przypadku natężenia prądu). Etap trzeci to wykonanie pomiarów, zakończony wraz z decyzją użytkownika o ewentualnym zapisaniu pomiarów na karcie SD.

Tryby manualne wyświetlają bieżącą wartość, którą można zapisać wciskając przeznaczony do tego przycisk. W trybach pomiarów ręcznych nie ma limitu pomiarów ani progu wartości, od której pomiar zostaje zrealizowany.

4. OPIS DZIAŁANIA MULTIMETRU

W opracowanym multimetrze wykorzystane są dwa programy. Pierwszy z nich to kod zaimplementowany w Arduino, który przesyła za pośrednictwem portu szeregowego, obliczone wartości pomiarów. Drugi program, umieszczony w pamięci Raspberry Pi, dotyczy sposobu obliczania przesyłanych wartości; wybierany jest on przez użytkownika z poziomu aplikacji. Dzięki niemu wyświetlane są na ekranie dotykowym wartości pomierzone z możliwością zapisu pliku tekstowego w pamięci urządzenia.



Rys. 5. Uruchomione urządzenie w roboczej obudowie

Program Arduino działa w pętli, która domyślnie przesyła komunikat tekstowy za pośrednictwem portu szeregowego i oblicza bieżące wartości napięcia oraz natężenia prądu. Pozostałą część programu stanowią opisane wcześniej tryby pracy. Zależnie od otrzymanej z Raspberry Pi wartości zmiennej `x`, aktywowana zostaje dana część funkcji działającej w pętli (dot. Arduino).

W przypadku trybów automatycznych, obliczone zostaje napięcie lub natężenie prądu. Program zapisuje do tablicy pięć wartości w odstępach 200 milisekund. Po zebraniu pięciu wartości, obliczona zostaje ich wartość średnia lub maksymalna (zależnie od przyjętego trybu). W przypadku wartości średniej, program, przy każdej z pięciu wartości, dodaje je do zmiennej przechowującej sumę. Zmienna ta po podzieleniu przez pięć zwraca średnią z pomiaru.

Pierwsza z pięciu składowych pomiaru zostaje zapisana jako maksymalna wartość. Następnie program sprawdza, czy któraś z pozostałych czterech wartości jest większa od pierwszej. Jeżeli tak, wartość maksymalna zostaje uaktualniona. Czujnik natężenia prądu ACS712 rejestruje zarówno wartości dodatnie, jak i ujemne (ze względu na polaryzację). W związku z tym, zamiast maksymalnej wielkości, wskazywana jest zmienna o najwyższej wartości bezwzględnej. Końcowym etapem kodu wykonywanego w pętli jest przesłanie uzyskanej z pomiaru wartości. W trybach automatycznych, dodatkowo poprzedzone jest to wyzerowaniem zmiennych odpowiedzialnych za obliczenie przesyłanych wartości. Zabieg ten niweluje ryzyko przesłania błędnego odczytu. Po przesłaniu wartości pomiaru, program odczeka około sekundy. Okres przerwy pozwala użytkownikowi na odłączenie sond pomiarowych i zaaplikowanie ich w innym miejscu obwodu. Interwał ten został dobrany tak, aby zezwolić na możliwie wygodną pracę z urządzeniem przy jednoczesnym zachowaniu poprawnej transmisji danych. Tryby pomiarów manualnych sprowadzają się do ciągłej transmisji wartości odpowiadających bieżącemu napięciu lub natężeniu prądu w odstępach 0,5 sekundy.

4.1. Opis programu zaimplementowanego na Raspberry Pi

Programowanie aplikacji w środowisku Qt Creator wymaga stworzenia struktury plików programu wraz z zaprojektowaniem elementów interfejsu graficznego. Po zaprojektowaniu wszystkich ekranów aplikacji, zostały zaprogramowane poszczególne tryby pomiarowe oraz menu główne. Przy funkcjach odpowiedzialnych za przekierowanie do wybranego trybu automatycznego (zachodzi to po wciśnięciu przycisku), została dodatkowo zaimplementowana transmisja liczby 0 za pomocą magistrali I²C do Arduino. Stanowi to dodatkową gwarancję zatrzymania transmisji pomiarów w przypadku, gdy użytkownik zmienia tryb z manualnego na automatyczny podczas nawigacji w programie. W ten sposób, wyeliminowano problem odczytu wartości błędnych, pochodzących z innego trybu oraz tych o wartościach mniejszych od zdefiniowanego progu rozpoczynającego pomiar.

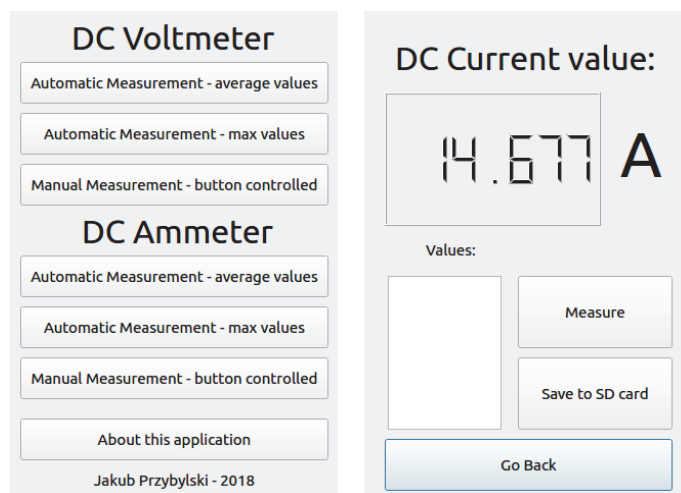
Funkcje związane z nawigacją w menu głównym zawarte są w pliku `mainwindow.cpp`. Każdy z trybów pracy zawiera te same linijki kodu związane z wstępną konfiguracją komunikacji z Arduino do modułu Raspberry [8]. Wartość opisywanej zmiennej `x` zostaje pobrana z elementu interfejsu graficznego po wskazaniu jej przez użytkownika. Następnie przesyłana jest ona do Arduino, aby zmienić tryb pracy mikrokontrolera zgodnie z jej wartością. Do wartości zmiennej `x` dodawane są wielokrotności liczby 50, zależnie od trybu tak, aby końcowa wartość przesłana do Arduino była zgodna z wcześniejszym opisem. Finalnie, wyświetlony zostaje ekran prezentujący w polu tekstowym przesłane

do Raspberry Pi pomiary, wraz z sukcesywnie malejącą liczbą pozostałych pomiarów. Arduino przesyła do Raspberry Pi wartości w formie liczb rozdzielonych przecinkiem. Zadaniem dodanego kodu jest zinterpretowanie odbieranych danych jako ciągu znaków i rozdzielenie go na zmienne liczbowe. Jeżeli dane nie są przesłane w kompletnej formie, na co wskazuje ich niewielka długość, zostają wyzerowane i następuje kolejna próba odczytu.

4.2. Obsługa aplikacji

Uruchomieniu Raspberry Pi towarzyszy włączenie aplikacji. Z poziomu menu głównego można wybrać jeden z sześciu dostępnych trybów pracy. Każdy z nich posiada „przycisk”, pozwalający przerwać pracę w dowolnym momencie.

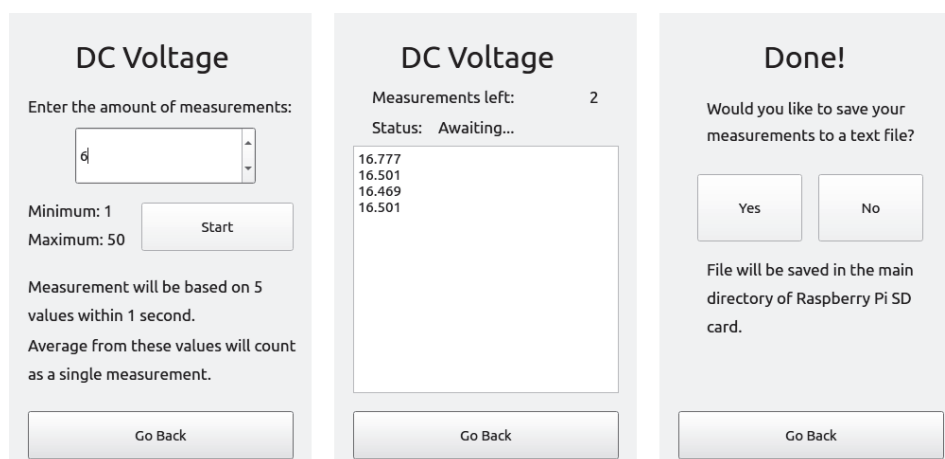
Manualne tryby pracy wyświetlają bieżącą wartość napięcia lub natężenia prądu otrzymywaną z Arduino. Użytkownik ma w każdym momencie możliwość zapisania aktualnie wyświetlanej wartości i wyeksportowania ich do pliku tekstowego. Wciśnięcie przycisku odpowiedzialnego za zapis pomiarów do pliku resetuje także zbiór dotychczas zebranych pomiarów do stanu początkowego.



Rys. 6. Menu główne oraz tryb ręcznego pomiaru natężenia prądu stałego

Automatyczne tryby pracy zawierają podobny interfejs. Składa się on z trzech ekranów. Pierwszy z nich wyjaśnia działanie wybranego trybu i pozwala na wybór zaplanowanej liczby pomiarów. Drugi ekran pojawia się po wciśnięciu przycisku „Start” rozpoczynającego pracę multimetru. Wyświetla on zmierzone wielkości wraz z liczbą pozostałych do końca pracy pomiarów. Po informacji o zakończeniu pomiarów i upływie pięciu sekund, wyświetlone zosta-

je pytanie o zapis pomiarów do pliku tekstowego na karcie SD. Po decyzji użytkownika program powraca do menu głównego i jest ponownie gotowy do pracy.



Rys. 7. Pomiary napięcia prądu stałego wykonywane w jednym w trybów automatycznych

Zapisane pliki tekstowe z pomiarami użytkownika przechowywane są w głównym katalogu karty SD umieszczonej w Raspberry Pi. Nazwa każdego pliku zawiera tryb pracy, w którym dokonano pomiarów oraz dokładną datę i czas zapisania pliku, co ułatwia sortowanie zebranych pomiarów. Każdy z pomiarów zapisany jest w osobnej linii. Dane wyświetlone w formacie umożliwiającym ich przeniesienie np. do arkusza kalkulacyjnego.

5. PODSUMOWANIE

Testy przeprowadzone z zaprezentowaną konstrukcją wykazały, że przyjęta koncepcja jest poprawna. Istnieją techniczne przesłanki do kontynuowania prac nad opracowaniem inteligentnego multimetru, niekoniecznie służącego do pomiaru parametrów obwodów elektrycznych. Zastosowane rozdzielenie funkcji pomiarowych i sterujących na odrębne układy umożliwia zbudowanie przyrządu obsługiwane go za pomocą ekranu dotykowego o różnorodnych możliwościach pomiarowych. Rozwinięcie aplikacji w obszarze sterowania obwodami wejściowymi miernika umożliwi dopasowanie zarówno „ustroju” pomiarowego jak i mutację samego ekranu do aktualnych potrzeb użytkownika np. możliwy staje się pomiar tym samym urządzeniem wielkości fizycznych tak odległych jak napięcie i współczynnik kwasowości roztworów wodnych. Praca udowodniła, że możliwe jest zbudowanie przyrządu w pełni uniwersalnego dostosowującego możliwości pomiarowe do aktualnych potrzeb użytkownika.

LITERATURA

- [1] Kumar A., 2016, What is a digital multimeter (DMM) and its working principle details, <https://analyseameter.com/2015/09/digital-multimeter-dmm-working-principle.html> (dostęp 09.09.2017).
- [2] Monk S. (2014), Raspberry Pi. Receptury, Gliwice: Helion.
- [3] Monk S. (2015), Arduino dla początkujących. Kolejny krok, Gliwice: Helion.
- [4] Summerfield M. (2014), Biblioteki Qt. Zaawansowane programowanie przy życiu C++, Gliwice: Helion.
- [5] <https://www.raspberrypi.org/blog/raspberry-pi-3-on-sale/> (dost. 22.01.2018).
- [6] [http://www.waveshare.com/wiki/3.5inch_RPi_LCD_\(B\)](http://www.waveshare.com/wiki/3.5inch_RPi_LCD_(B)) (dost. 11.09.2017).
- [7] <https://www.sparkfun.com/datasheets/BreakoutBoards/0712.pdf> (dost. 12.01.2018).
- [8] <https://github.com/vannevar-morgan/Qt-Temperature-Sensor> (dost. 08.01.2018).

APPLICATION OF ARDUINO AND RASPBERRY Pi 3 COUPLED PLATFORMS FOR MEASURING CURRENT AND VOLTAGE IN DC CIRCUITS

The article presents the concept and implemented physical model of the device for voltage and direct current measurement. The meter was built by coupling a Raspberry microcomputer with an Arduino series microcontroller. Applications have been developed to implement proprietary programs for both devices. The device has been equipped with a resistive touch screen allowing the user to decide on the mode of measurements (automatic or manual) and on the method of archiving and visualizing the effects of measurements. The application uses the so-called cross-compiling, thanks to which it is possible to effectively connect such separate structures as the Raspberry and Arduino systems.

(Received: 11.02.2018, revised: 10.03.2018)