

Graficzne Programowanie Mikrokontrolerów

Ćwiczenie 4

Dane

Wstęp

Informacje przetwarzane przez oprogramowanie mikrokontrolera są za pośrednictwem zmiennych. Podstawowe typy danych (liczbowy, tekstowy i logiczny) pozwalają na przetwarzanie pojedynczej wartości danego typu. W wielu przypadkach konieczne jest jednak gromadzenie i przetwarzanie danych złożonych, serii danych. Micropython i środowisko UIFlow pozwala na zdefiniowanie dwóch typów danych złożonych listy i mapy.

Dane są zapisywane w pamięci ulotnej. W przypadku restartu systemu zostają one utracone. W systemach kontrolno-pomiarowych istotne jest aby gromadzone dane były bezpieczne i były dostępne do przetwarzania i analizy w czasie rzeczywistym. Środowisko UIFlow dla mikrokontrolerów M5Stack udostępnia trzy sposoby trwałego dostępu do danych, w pamięci EEPROM, na karcie microSD i w chmurze.

1 Złożone typy danych

Złożone typy danych w UIFlow dostępne są do zaprogramowania za pośrednictwem dwóch grup bloków, list i map. Obydwa typy mogą przetwarzać dane dowolnego z podstawowych i złożonych typów.

1.1 Listy

Listy są do pewnego stopnia odpowiednikiem tablic znanych z języków C/C++. Pozwalają na przechowywanie serii danych indeksowanych numerycznie. Poszczególne dane mogą być dowolnego typu (różnych typów). Dane listy indeksowane są od 1 do n (w Blockly, micropython indeksuje od 0). Możliwe jest także indeksowanie od końca listy, w takim przypadku 0 oznacza ostatni element, -1 przedostatni itd.

Grupa Lists dostarcza narzędzi tworzenia, manipulowania, umieszczania oraz wyciągania danych z listy.

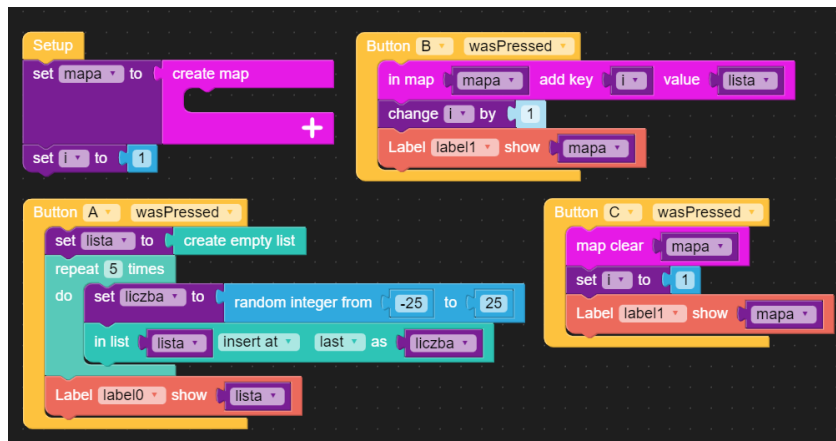
1.2 Mapy

Mapy w swojej budowie są bardzo zbliżone do listy. Podstawowa różnica polega na sposobie indeksowania elementów. W mapie elementy indeksowane są za pomocą ciągu tekstowego. Najbliższym odpowiednikiem mapy może być słownik, gdzie pojedynczemu hasłu można przypisać dowolną liczbę pojęć (wartości).

W grupie dostępne są podstawowe bloki do manipulacji danymi słownikowymi. Poza tworzeniem i kasowaniem mapy, można dodawać, modyfikować, odczytywać i wyszukiwać pozycje mapy.

PRZYKŁAD

Przycisk "A" generuje zestaw 5 liczb losowych z zakresu $\langle -25, 25 \rangle$, zapisuje je do tablicy, oraz wyświetla je na ekranie. Przycisk "B" wstawia wygenerowaną listę do Map'y, oraz wyświetla na ekranie aktualną jej zawartość. Przycisk "C" czyści Map'ę.



Rys. 1: Przykładowy kod realizujący zadanie z wykorzystaniem Listy i Mapy

2 Archiwizacja danych

Dane w środowisku UIFlow można archiwizować na trzy sposoby:

- w pamięci EEPROM,
- na karcie microSD,
- w chmurze.

W każdym z sposobów realizacji zapisu i odczytu danych istotne jest kontrolowanie czasu i unikanie równoczesnego odwołania do źródła danych.

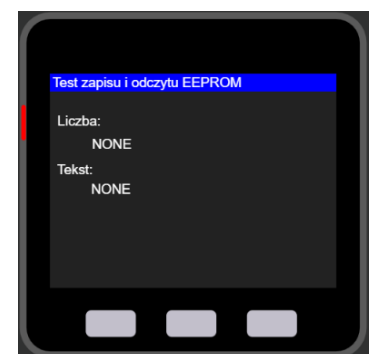
2.1 EEPROM

Grupa pozwala na zapis i odczyt danych z pamięci nieulotnej EEPROM. Można zapisywać pojedyncze dane liczbowe lub tekstowe i powiązać je z unikalną nazwą. Procedura jest wolna, kolejne odwołanie do pamięci EEPROM możliwe jest po około 5 sekundach.

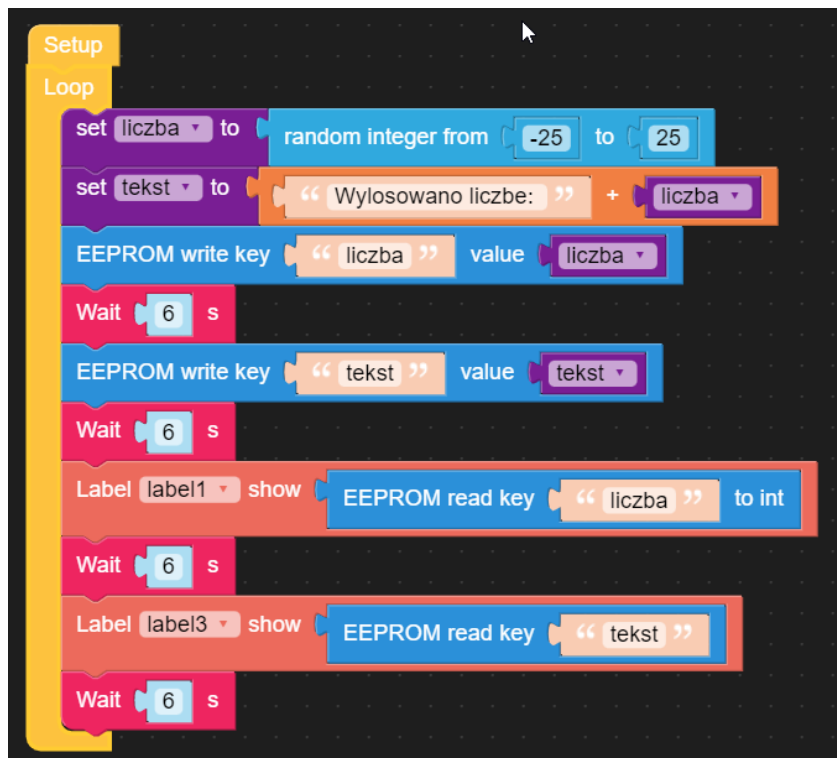
Próba częstszego odwołania do pamięci związana jest z ryzykiem niezapisania informacji lub zawieszenia systemu. Częste korzystanie z tej pamięci może skutkować także wcześniejszym jej zużyciem. Jest ona raczej przeznaczona do zapisu danych konfiguracyjnych aplikacji które zapisywane i odczytywane są sporadycznie (np. w momencie uruchomienia aplikacji).

PRZYKŁAD

Program zapisuje i odczytuje z pamięci EEPROM dane tekstowe i liczbowe. Zalecane jest zachowanie odstępu czasowego pomiędzy transakcjami w celu ochrony procedury dostępowej przed przedwczesnym ponownym odwołaniem się do niej.



Rys. 2: Interfejs graficzny aplikacji testowej



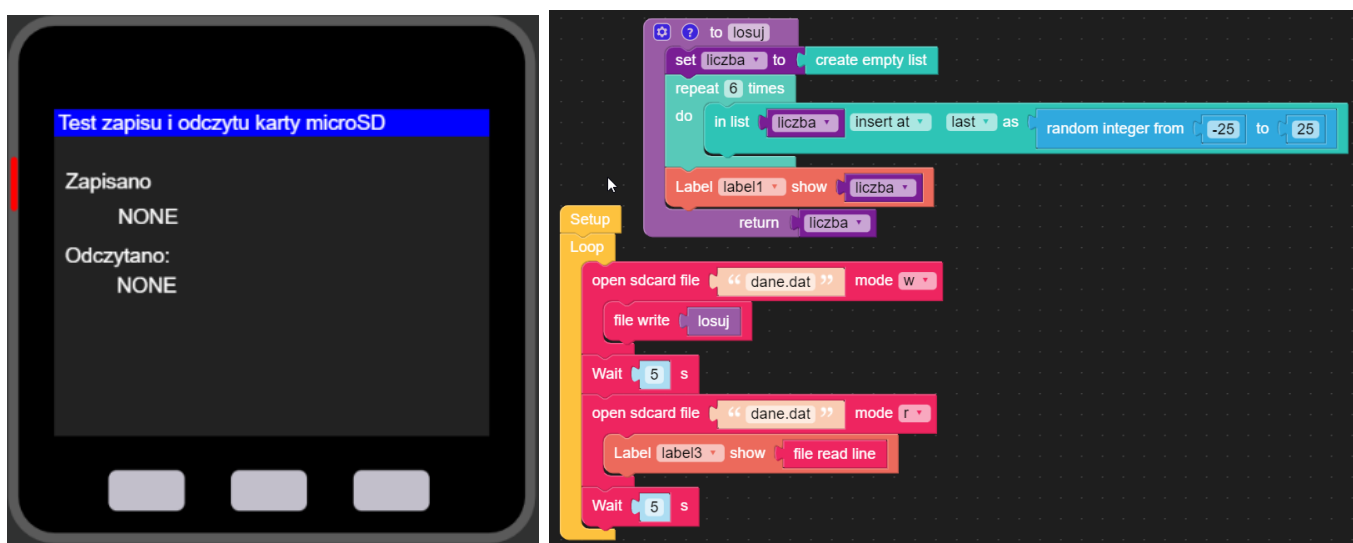
Rys. 3: Kod testowy zapisu i odczytu danych z EEPROM

2.2 microSD

Za pomocą bloków z grupy SDCard możliwe jest zapisywanie i odczytywanie danych z pliku tekstowego umiejscowionego na karcie microSD. Dostarczone bloki pozwalają na realizację zadań związanych z tworzeniem i manipulowaniem katalogami, plikami i danymi. Dane dowolnego typu mogą być zapisywane w pliku tekstowym. Zapisane dane mogą być odczytane jako ciąg tekstowy. Konieczne więc staje się opracowanie algorytmu dekodującego dane tekstowe na właściwy typ danych. W przypadku zapisu danych złożonych, list lub map, ich odczytanie staje się zadaniem skomplikowanym.

PRZYKŁAD

Program testuje sekwencje zapisu i odczytu listy do i z pliku tekstowego na karcie microSD.

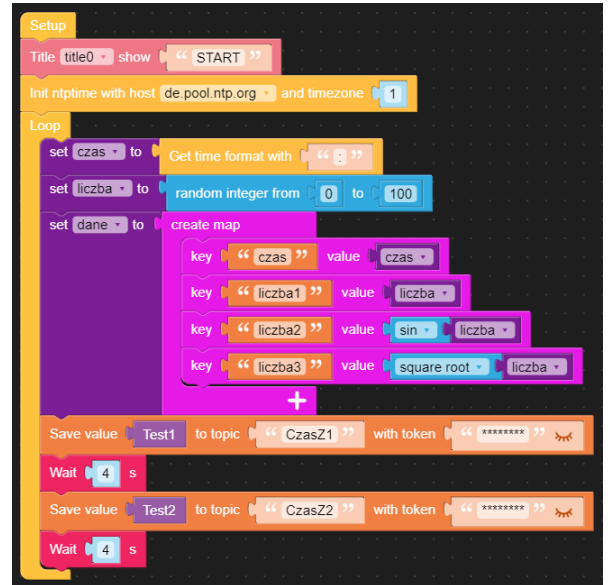


Rys. 4: Kod testowy zapisu i odczytu danych z karty microSD

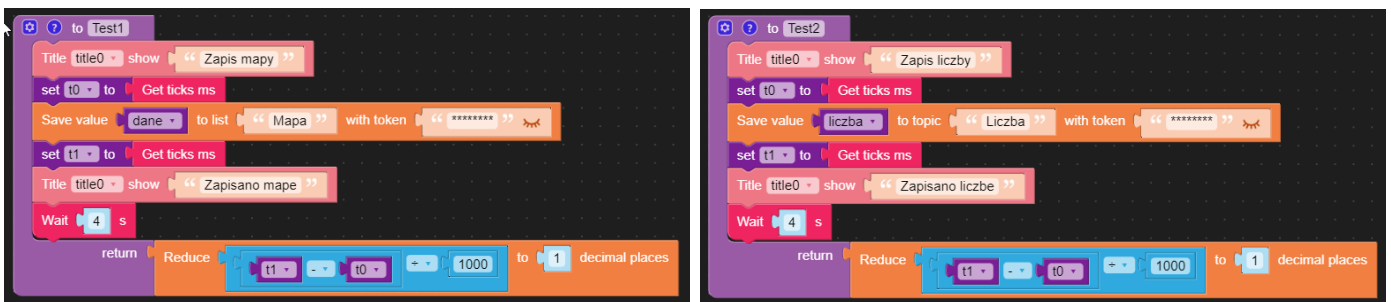
2.3 Chmura

M5Stack dla swoich produktów udostępnił przestrzeń na dane w chmurze. Bloki realizujące ten sposób zapisu dostępne są w grupie EzData. Dla każdego mikrokontrolera tworzony jest indywidualna przestrzeń identyfikowana unikalnym kluczem.

W chmurze można zapisywać trzy typy danych, ciąg tekstowy, liczbę i listę. Pod pojęciem listy rozumiana jest zarówno LISTA jak i MAPA. Dane do chmury mogą być przesyłane SYNCHRONICZNIE i ASYNCHRONICZNIE. Blok realizujący przesył synchroniczny zakończy swoje działanie po potwierdzeniu przez serwer transakcji, natomiast w trybie asynchronicznym blok jedynie zleca transakcje. Kolejny blok jest realizowany niezależnie od wyniku. Z każdym blokiem transmisji asynchronicznej związany jest blok zdarzenia kontroli transakcji umożliwiającą zaprogramowanie zdarzenia na udaną i nieudaną transmisję.



Rys. 5: Pętla główna programu testującego



Rys. 6: Funkcja testowa zapisu Mapy i danej

Każda transakcja realizowana na chmurze wymaga pewnego czasu na realizację, około 2-3 sekund (zależnie od szybkości połączenia z serwerem). Przedwczesna próba dostępu do chmury może wiązać się z nieudaną transakcją lub zawieszeniem aplikacji.

PRZYKŁAD

Program testuje synchroniczny zapis danych (liczbowa i lista) na chmurze EzData. Zadania są realizowane niezależnie za pośrednictwem funkcji, które dodatkowo obliczają czas wymagany do realizacji transakcji zapisu na chmurze. Dane te są także zapisywane w bazie. Przedstawiony na rysunku 8 przebieg obrazuje czas transakcji zapisu danej liczbowej w chmurze przy transmisji synchronicznej. Dla aplikacji przygotowano także interfejs WWW, pokazany na rysunku 7, prezentujący wybrane dane gromadzone na chmurze.

3 Remote+

Usługa dostępna dla mikrokontrolerów M5Stack za pomocą której można zaprojektować interfejs WWW aplikacji. Dostępne są moduły realizujące zadanie wejściowe:

- wprowadzanie ciągu tekstowego,
- przycisk,
- joystick,
- slider,
- inkrementator,
- przełącznik.

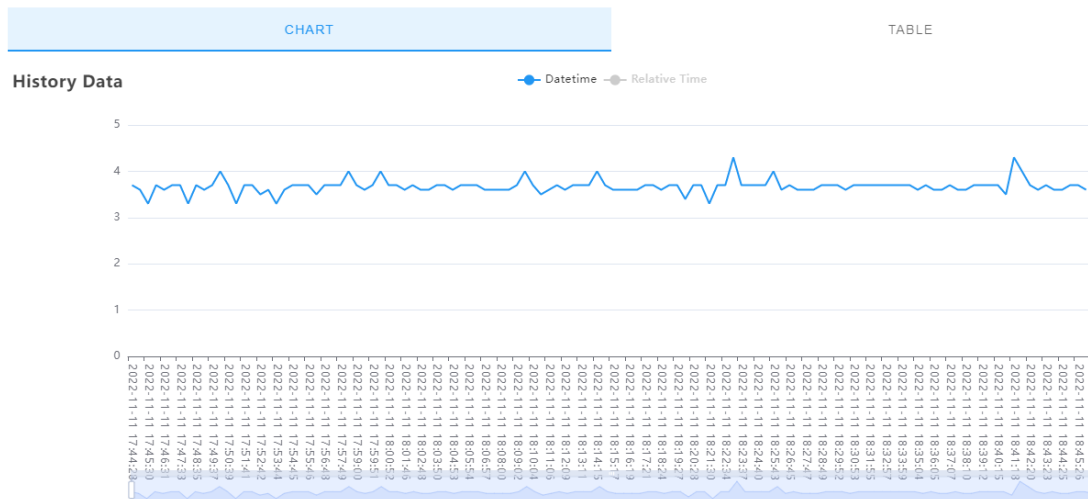
jak i wyjściowe:

- wykres,
- wskaźnik zegarowy,
- ciąg tekstowy.



Rys. 7: Interfejs Remote+

Moduł REMOTE+ pozwala na prezentację danych dostarczanych przez aplikację mikrokontrolera lub zapisane w chmurze EzData. Link do aplikacji realizowany jest przez generowany w UIFlow QRCode.



Rys. 8: Wykres historii zmian danej wysyłanej na chmurę

4 Terminal

Terminal jest wygodnym narzędziem kontroli działania aplikacji na etapie projektowania. Terminal realizuje połączenie szeregowo z mikrokontrolerem i umożliwia odbieranie i wysyłanie danych do mikrokontrolera.

Za pomocą bloku PRINT z grupy TEXT można przysyłać dane z mikrokontrolera na ekran terminala. Aby komunikacja z terminalem była aktywna w czasie działania aplikacji kod programu należy przesłać z poziomu aktywnego okna terminala.

5 ZADANIA

Zad. 1.

Zaprogramować aplikację analizującą czas realizacji różnych sposobów zapisu i prezentacji danych:

- prezentacja danych na ekranie LCD mikrokontrolera,
- przesyłanie danych do terminala,
- zapis i odczyt danych z karty microSD,
- zapis i odczyt danych z chmury w trybie synchronicznym i asynchronicznym.

Przeanalizować prezentacje, zapis i odczyt danych liczbowych, tekstowych i tablicowych. Wyniki testów zaprezentować w wybrany przez siebie sposób:

- na ekranie LCD mikrokontrolera,
- w interfejsie sieciowym, wykresy,
- przesłać na ekran terminala.

Pomiar czasu realizacji poszczególnych zadań przeprowadzić za pomocą bloku GET TICKS MS z grupy TIMER.

Zad. 2a.

Wykorzystując typy danych złożonych (lista, mapa) napisać program stacji diagnostycznej gromadzącej dane o temperaturze, wilgotności i ciśnieniu. Program wyświetla informacje o wartościach parametrów środowiskowych w danej chwili oraz wartościach średnich z kilku cykli czasowych (np. minutowa, godzinowa, dobowo,...).

Zad. 2b.

Zmodyfikować program z zadania 2a aby dane gromadzone były w chmurze. Zaprogramować interfejs sieciowy (WWW) dla aplikacji prezentujący dane w postaci przebiegów czasowych i wartości chwilowych badanych parametrów środowiskowych. Dane dla kontrolek realizujących wykresy w interfejsie WWW muszą być zapisywane za pomocą typu MAP.