

Sprawozdanie z projektu

1. Opis projektu

Tematem naszego projektu jest próba stworzenia urządzenia podobnego do oscyloskopu z wykorzystaniem płytki deweloperskiej LandTiger v2.0, opartej na mikrokontrolerze LPC1768 oraz wyświetlaczu HY32D.

Problemy, z którymi się zmierzaliśmy to:

- opracowanie optymalnego algorytmu pozyskiwania danych pomiarowych,
- obróbka danych,
- przedstawienie danych na wyświetlaczu,
- umożliwienie zarządzania całym urządzeniem z poziomu interfejsu wyświetlacza.

2. Abstrakt

Zaprogramowaliśmy kontroler tak, by wyświetlał w poprawny sposób dwa przebiegi prądowe o częstotliwości do 20kHz i napięciu od 0 do 3.3V. Z pomocą cyfrowego wyzwalacza w jednym z przebiegów można wymusić utrzymanie stałej fazy początkowej. Udało się nam zaprojektować i wprowadzić algorytm filtrowania danych z „peak-ów”, które generują się przez płytkę w nieprzewidywalny sposób. W celu ilościowego zbadania dwóch przebiegów wyznaczono podstawowe wartości statystyczne częstotliwości i amplitudy mierzonego sygnału.

Zebrane dane wyświetlane są na dostarczonym do płytki wyświetlaczu LCD. Aby możliwe było ich wyświetlanie na ekranie, poprawiliśmy i znacznie zmodyfikowaliśmy przykładowy program dostarczony przez producenta płytki, przez co można np.:

- oznaczać sprzętowo obszar, poza którym malowanie nie jest możliwe – dzięki temu można wysłać do wyświetlacza informację o początkowym kursorze by później tylko wysłać kolory pikseli – wyświetlacz sam odpowiednio ustala piksele w zaznaczonym obszarze. Dzięki temu przyspieszona została operacja malowania prostokątów, co jest użyte np. do czyszczenia ekranu
- ustalać orientację ekranu
- oznaczyć tryb malowania szablonem – zamiast pikseli wykorzystywana jest funkcja, która na podstawie położenia (x, y) zwraca kolor.

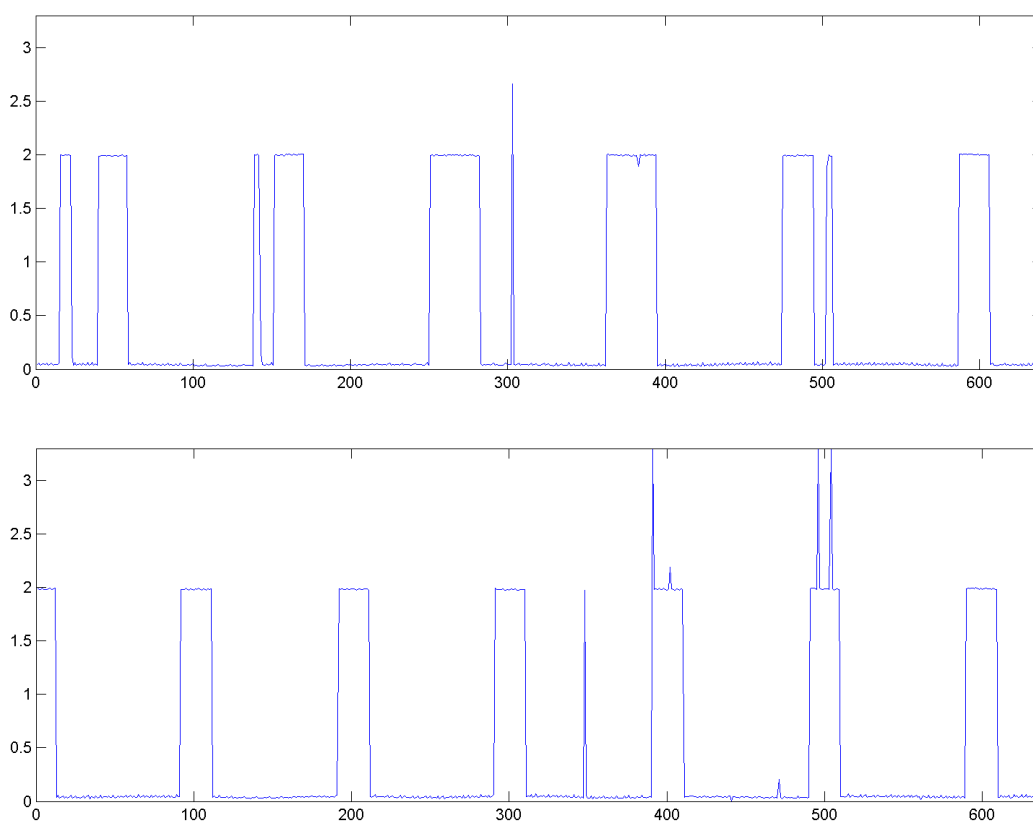
Na potrzeby projektu utworzony zostały moduły, które obsługiwały m. in:

- wyświetlanie i sterowanie wyświetlaczem
- rysowanie wykresu
- konwersję różnych formatów do stringa (niestety, sprintf nie spełnił swojego zadania...)
- przechowywanie wartości, które da się w prosty sposób zwiększyć z ustaloną dokładnością – struktura Sci (przechowywanie mantysy i wykładnika w postaci całkowitej zamiast reprezentacji zmiennoprzecinkowej; wzór: $\text{matysa} * 10^{\text{wykładnik}}$)
- obsługę akcji użytkownika poprzez 3 programowalne przyciski (KEY1, KEY2, INT0; wszystkie na przerwaniach) i joystick (tutaj bez przerwania, gdyż joystick podłączony został

pod piny znajdujące się w porcie 1, czego skutkiem jest brak możliwości podpięcia przerwania zewnętrznego EINT3);

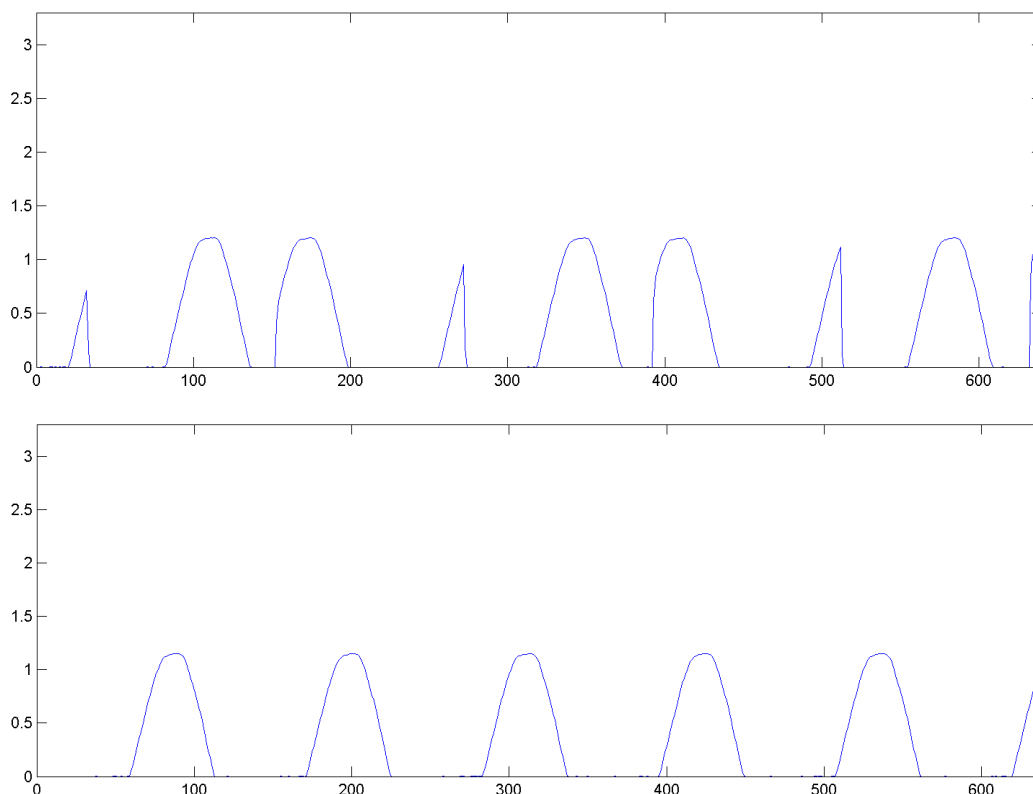
3. Ograniczenia płytki deweloperskiej

Płytką LandTiger zawiera wiele gotowych połączeń do różnych funkcjonalności mikrokontrolera. Zawiera więc wejścia i wyjścia do złącz USB, DB9, Ethernet, CAN oraz wyposażona jest w różne komponenty: 8 diód, potencjometr, joystick, przyciski. Posiada też wiele innych elementów elektronicznych budujących cały układ płytki. Z tego powodu z 8 wyjść multipleksowanych ADC, dostępne są tylko 2 (jedno dostępne po zdjęciu zworki na głośniku, drugie po zdjęciu zworki na potencjometrze; innych wejść nie da się odłączyć od odpowiednich komponentów). Ponadto cały układ jest na tyle gęsto upakowany, że przypuszcza się, iż wpływa on znacznie na pomiar napięcia. Sprawdziliśmy, że pomiar napięcia w trakcie wysyłania danych na wyświetlacz lub np. wyjście DB9, objawia się to skokami poziomu napięcia na wykresie. Pomiar jest utrudniony nawet gdy jest on wykonywany osobno (wiele, głównie jednostkowych „peak-ów” na wykresie).



Wykres 1 Przebieg prostokątny ok. 56Hz (PWM, inny kontroler), na górze pomiar w trakcie wysyłania danych, na dole pomiar odosobniony od innych operacji.

Przedstawione przebiegi były testowane w aplikacji matlab poprzez wysłanie nieprzetworzonych danych i narysowanie ich wprost. Pobrano wiele (po 100) takich pomiarów i przy podanych wyżej trybach pracy (pomiar w trakcie innych operacji i pomiar wykonywany osobno) wykresy były podobnie zniekształcone.



Wykres 2 Przebieg sinusoidalny 50Hz (prąd sieciowy), na górze pomiar w trakcie wysyłania danych, na dole pomiar odosobniony od innych operacji.

4. Pierwotna wersja projektu

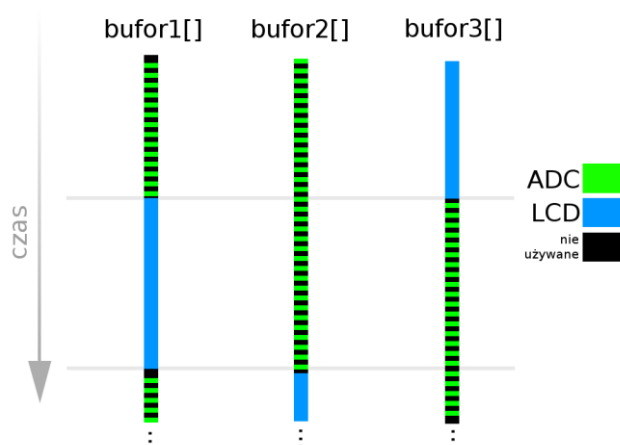
Wąskim gardłem czasowym urządzenia jest czas rysowania LCD. Z tego powodu logicznym było, aby poza optymalizacją rysowania, zmniejszyć czas oczekiwania na dane do minimum. Poszukiwanym rozwiązaniem miał się okazać system buforowania danych, który umożliwiałby zarówno próbkowanie w niezależnym od procesora peryferium ADC jak i kontrole LCD jednocześnie¹. Powstały na te potrzeby 3 buforów danych, każdy na dwa kanały pomiarowe (razem 6).

Przebieg algorytmu był następujący:

- w danym momencie w jednym z trzech buforów odkładały się próbki danych,
- w drugim buforze przechowywany był gotowy pomiar,
- na żądanie LCD bufor drugi był kopiowany do buforu trzeciego,
- w trzecim buforze następowała obróbka danych,
- ostatecznie, trzeci bufor był używany do rysowania oscylogramu przez LCD.

¹ Obsługa LCD odbywająca się w głównej pętli programu, wyłączana przez przerwanie końca pomiaru ADC.

System wykluczał konflikty dostępu do tablic danych i eliminował niepotrzebne marnowanie czasu.



Wykres 3 Protokół używania tablic.

Po wyeliminowaniu wielu problemów doszliśmy do wniosku, na podstawie wspomnianych we wcześniejszym punkcie spostrzeżeń, że pozostałe błędy wykresu wynikają z przyczyn fizycznych (nie programowych) i leżą po stronie budowy obwodów płytki drukowanej. Z tego powodu **zrezygnowaliśmy** z jednoczesnego pomiaru i wysyłania danych na LCD lub UART.

5. Generowanie danych

5.1. Pozyskiwanie danych

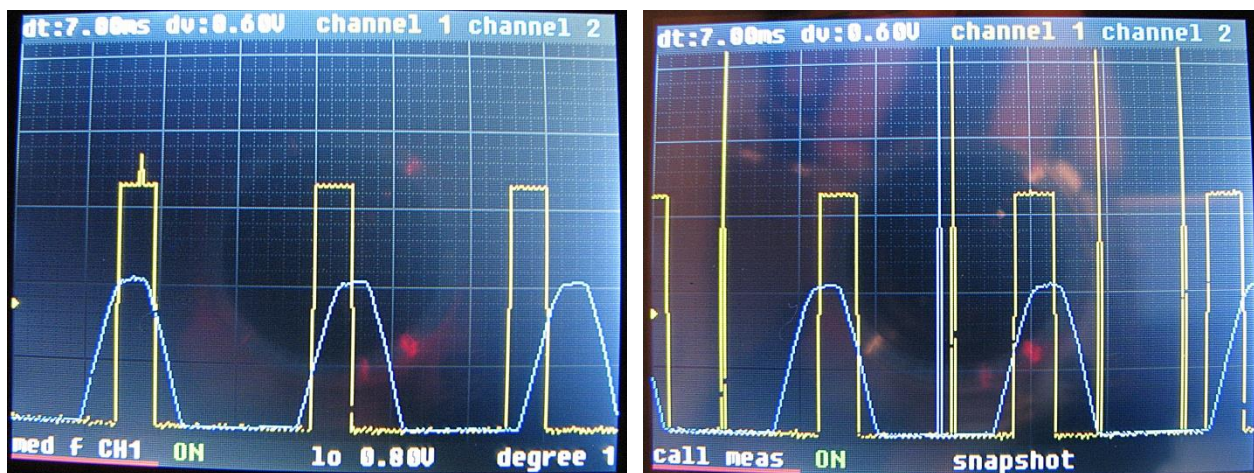
Z pomocą jednego z Timerów następuje wyzwolenie pomiaru ADC z zadaną częstotliwością. Wynik pomiaru jest zapisywany w obsłudze przerwania (zakończenie pomiaru) do pamięci. Tak przeprowadzane próbkowanie jest wykonywane z góry określoną ilość razy. Następnie przerwanie zapisu danych jest wyłączane na czas oczekiwania na kolejne zlecenie. Po wykonaniu pomiarów następuje ich wyświetlenie.

5.2. Gromadzenie danych

Zmierzone wartości odkładają się w jednej z dwóch tablic (zależnie od tego, który kanał jest mierzony) o wielkości dwa razy większej niż szerokość wyświetlacza. Typ danych nie ulega zmianie – jest to 16-bitowa liczba całkowita bez znaku. Dane otrzymywane z pomiarów mają wielkość 12 bitów.

5.3. Przetwarzanie danych

Wraz z obsługą przerwania na koniec pomiaru, dane są zapisywane do pamięci, a równoległe do niego, następuje filtracja, odpowiednio wcześniejszych danych z zakłóceń własnych urządzenia za pomocą specjalnej procedury. Gdy wartość pojedynczego pomiaru różni się od sąsiednich wartości (obu niezależnie) o więcej niż wartość graniczna, wartość tego pomiaru zastępowana jest wartością średnią tych sąsiednich pomiarów. Wartość graniczną ustawia się z poziomu wyświetlacza. Filtr umożliwia pracę wyzwalacza cyfrowego.



Wykres 4 Po lewej wykres z filtracją, po prawej bez.

5.4. Wyzwalacz cyfrowy

Przed wyświetleniem danych określone jest miejsce od którego mają zostać one czytane. Danych jest dwa razy więcej niż tych potrzebnych do wyświetlania. Pierwszy rysowany punkt znajduje się poprzez przeszukanie tablicy danych od środka do początku, sprawdzając zadane warunki poziomu napięcia i typu zbocza wyzwalającego.

6. Cykl pomiarowy.

Przedstawiony poniżej algorytm wykonuje się na zlecenie – jest to obsługiwane przez funkcję `osc.meas()`:

- Zostaje ustawiona opcja SLEEPONEXIT usypiania procesora
- Następnie przeprowadzana jest reinicjalizacja ADC
 - Pobiera i ustawia się konfigurację pomiaru (częstotliwość, filtr, aktywne kanały)
 - Poszukuje się pierwszego aktywnego kanału i podłącza się jego przerwanie lub gdy wszystkie kanały zostały zmierzone wyłączona zostaje opcja SLEEPONEXIT i ustawia się flagę końca pomiaru
- Funkcja obsługująca przerwanie końca pomiaru ADC wyłuskuje i zapisuje wynik do odpowiedniego miejsca w pamięci; ta sama funkcja przeprowadza na bieżąco filtrację zakłóceń na odpowiednio wcześniejszym pomiarze; gdy zostaną zebrane wszystkie próbki funkcja odłącza własne przerwanie i przeprowadza reinicjalizację w celu zmierzenia kolejnego kanału
- Oczekuje się na ustawienie flagi końca pomiaru (całego cyklu próbkowania) w pętli głównej programu

6.1. Dodatkowe wyjście danych

W trakcie pracy nad projektem pojawił się pomysł wysyłania danych pomiarowych na zewnątrz do komputera PC. Powstały do tego odpowiednie funkcje i interfejs. Wysyłanie danych zostało wykorzystane w innym projekcie, który polegał na zbudowaniu gui w aplikacji MATLAB. Projekt ten został pomyślnie ukończony.

Po pewnym czasie kod programu Cortex-a uległ znacznej zmianie i w najnowszej wersji projektu te udogodnienie nie zostały przetestowane.

Niemniej jednak, UART został skonfigurowany na prędkość 115 200 baud'ów z dokładnością na poziomie błędów 0.0064%. Protokół komunikacji CORTEX <-> MATLAB jest postaci:

<data>\n\r<id>\n\r – dla wysyłania, gdzie

<data> to komenda lub dana pomiarowe typu uint16_t

<id> to nr inkrementowany z każdym żądaniem w celu kontroli poprawności protokołu oraz

<data>\n – dla odbioru, gdzie

<data> to zawsze komenda

Komendy:

- [GET_A_MEAS] w odpowiedzi kontroler wysyła dane pomiarowe
- [LOOP_BACK] w odpowiedzi kontroler wysyła tą samą komendę [LOOP_BACK]
- [LCD_ENABLE] przełączenie w kontrolerze na tryb LCD, odpowiedź [LCD_ENABLED]
- [UART_ENABLE] przełączenie w kontrolerze na tryb UART, odpowiedź [UART_ENABLED]
- [ID_COMM_RESETE] resetowanie id na kontrolerze

W przyszłości planuje się odbiór i ustawianie od strony PC takich wartości jak częstotliwość próbkowania czy aktywność kanałów.

7. Instrukcja użytkownika

Dwa dostępne kanały ADC, które są próbkowane to piny o numerach:

- P0.26 dla kanału pierwszego
- P1.31 dla kanału drugiego
- Napięcie odniesienia można podłączyć do dowolnego pinu opisanego przez GND na płycie.

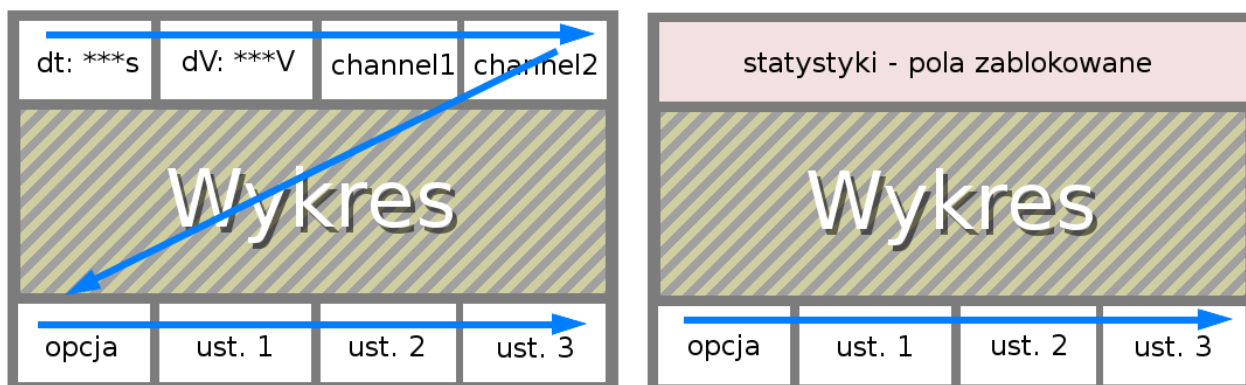
Przed pomiarem należy zdjąć zworki oznaczone numerami:

- JP12
- JP2
- JP8.

Nie wolno podłączać napięć większych niż 3.3V. Grozi to uszkodzeniem urządzenia.

Interfejs składa się z okna głównego, na którym rysowane są przebiegi napięciowe oraz pola tekstowe znajdujące się nad i pod głównym obszarem rysowania. Są to miejsca, z których użytkownik może odczytać informacje o pomiarze bądź zmieniać ustawienia oscyloskopu.

Poniżej przedstawiony jest ogólny wygląd interfejsu użytkownika:



Można najeżdżać na białe komórki w celu zmiany ustawień z nimi powiązanych – w tym celu należy wcisnąć przycisk KEY1 aby się cofnąć (przeciwny kierunek do niebieskiej strzałki) bądź KEY2 aby przejść do przodu (kierunek zgodny z kierunkiem strzałek). Przechodzenie między polami jest zapętlone. Bieżąca pozycja sygnalizowana jest poprzez pomarańczowe podkreślenie – kiedy podkreślenie nie występuje na ekranie, należy wcisnąć KEY1 bądź KEY2 aby się pojawiło. Jeśli tryb statystyk jest włączony to przechodzenie na górne pola jest zablokowane.

Gdy chcemy zmienić parametr, który znajduje się w danym polu, to wychylamy joystick w kierunku góra / dół. Wartością może być pewien zbiór wyliczeniowy, wartość liczbowa bądź wartość ON / OFF.

W zależności od pola, na którym się znajdujemy, możemy wykonywać różne akcje:

- W polu dt możemy zmieniać wartość, która określa czas między dwoma głównymi pionowymi podziałkami na wykresie. Odpowiada ona za częstotliwość próbkowania. Wartość można zmienić wychylając joystick w górę lub dół.
- dv służy do zmiany wartości napięcia między dwoma poziomymi podziałkami. Wartość można zmieniać pionowymi wychyleniami joysticka.
- channel 1 i channel 2 – pole służy do sygnalizowania koloru kanału bądź jego aktywowania. Tekst jest kolorowy gdy kanał jest aktywny, w przeciwnym wypadku kolor jest szary. Aby aktywować / deaktywować kanał należy wcisnąć przycisk INTO. Wykresy kanałów można przesuwac w pionie lub poziomie – aby tego dokonać, należy użyć joysticka.
- Opcje – cały pasek dolny – aktywne ustawienie można zmienić gdy znajdujemy się w pierwszym polu od lewej poprzez wychylenie gałki joysticka w górę bądź dół. Dostępne ustawienia:
 - Output – w tym polu możliwa jest zmiana wyjścia zmierzonych danych – domyślnie ustawiona jest opcja LCD, można ustawić UART jeśli chcemy np. pobrać dane na komputer.
 - Call meas – tryb pojedynczego pomiaru (na żądanie). Kiedy pole jest włączone, można przejść na pole „snapshot” i wcisnąć przycisk INTO aby wykonać pojedynczy pomiar.
 - Trigger – ustawienia związane z triggerem – aktywny kanał wyzwala (lub jego brak), poziom napięcia wyzwalającego i typ zbocza (rosnące bądź malejące).
 - Med F CH1 – filtr medianowy dla kanału 1 oscyloskopa. Służy do redukcji tzw. „peak-ów”, czyli nagłych impulsów, które mogą być powiązane z zakłóceniami. Filtr ten nie zawsze musi być pożądanym – np. chcemy obserwować wspomniane impulsy. Filtr medianowy można włączyć, wyłączyć i ustawiać minimalną wysokość impulsów do łagodzenia. Wartość degree

(na razie na stałe równa 1) informuje o tym, że łagodzone będą impulsy, o szerokości 1 punktu pomiarowego.

- Med. F CH2 – filtr medianowy dla kanału 2.
- Statistics – statystyki pomiarów. Po ich włączeniu górny pasek będzie pokazywał aktywne statystyki. Z tego też względu pasek ten zostaje zablokowany (nie można przejść na górne pola). Statystyka może dotyczyć amplitudy bądź częstotliwości. Oferowane typy statystyk:
 - Wartość minimalna, maksymalna, średnia i odchylenie średnie wskazanego kanału
 - Średnia i średnie odchylenie wszystkich kanałów
 - Wartość minimalna i maksymalna wszystkich kanałów

8. Wyświetlanie danych.

8.1. Podstawowa biblioteka do obsługi ekranu

Uwaga! Płytką, na której wykonywaliśmy projekt ma inną wersję wyświetlacza w porównaniu z płytkami z laboratorium. Użyty został inny sterownik do ekranu - w efekcie na płytkach LandTiger z laboratorium program prawdopodobnie nie będzie działać.

Za podstawowe wyświetlanie danych odpowiada moduł LCD wraz z biblioteką znaków AsciiLib. Moduł ten potrafi:

- Blokować sprzętowo obszary na ekranie, w których może odbywać się rysowanie. Możliwe jest to poprzez użycie sprzętowych funkcji R44, R45 i R45 sterownika ekranu SSD 1289. W programie realizowane jest to poprzez wykorzystanie funkcji `lcd_setClippingRegion` i `lcd_unsetClippingRegion`.
- Rysować przy pomocy szablonów, które realizowane są jako funkcje przyjmujące pozycję x i y na ekranie i zwracające kolor. Aby aktywować ten tryb, należy przypisać do zmiennej `lcd_pattern` opisywany typ funkcji.
- Rysować linie i punkty poprzez `lcd_drawPoint` i `lcd_drawPoint`.
- Rysować litery i pisać tekst – funkcje `lcd_printChar`, `lcd_printText`, `lcd_printNText`
- Czyścić cały ekran bądź jego fragment (tzn. malowanie prostokątu) – funkcje `lcd_clear` i `lcd_clear_all`. Funkcję `lcd_clear` musi poprzedzać funkcja `lcd_setClippingRegion`.

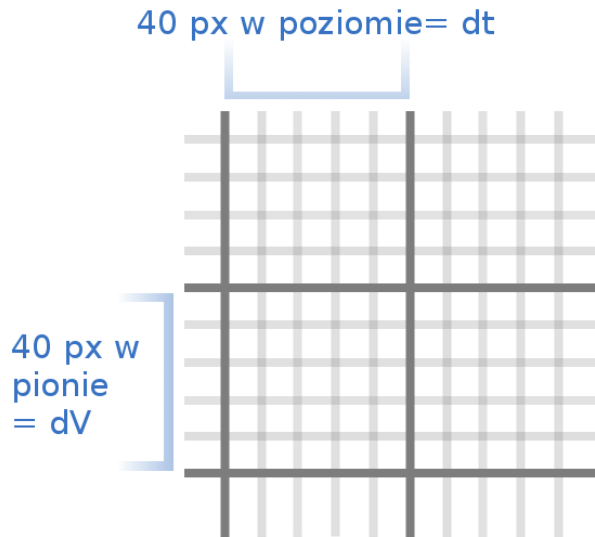
8.2. Obsługa interfejsu użytkownika

Ekran został zagospodarowany w ten sposób, by wykres znajdował się między dwoma paskami (na górze i na dole), na których znajdują się informacje dla użytkownika.

Pasek górny: 320 x 16 px - miejsce na 40 znaków

Obszar przeznaczony na wykres
320 x 208 px

Siatka na wykresie
- opisane główne
podziałki



Pasek dolny: 320 x 16 px - miejsce na 40 znaków

W celu obsługi GUI utworzony został moduł, który umożliwia:

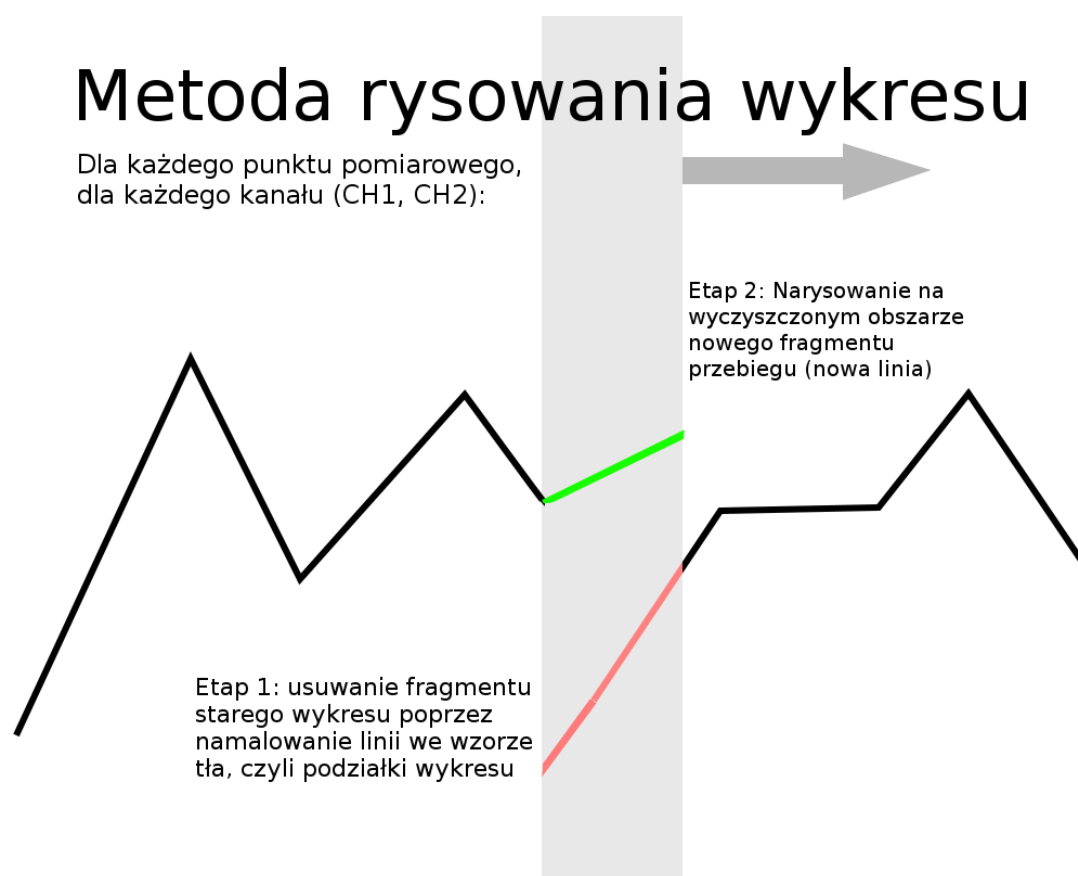
- Reakcję na akcje użytkownika i wywołanie odpowiedniej funkcji, która je obsługuje. Ustawione jest także przerwanie na SysTick które umożliwia filtrację polegającą na tym, że między następnym naciśnięciem przycisku lub poruszeniem joysticka musi upłynąć określona ilość czasu – na przyciski INTO, KEY1 i KEY2 przerwa wynosi 300ms, natomiast dla joysticka to 600ms.
- Zmiana wartości jest zależna tylko od długości przytrzymania wychylenia joysticka, przez co czas rysowania wykresu nie ma na to wpływu (zwiększa się on naturalnie przy pomiarze wolnych przebiegów).
- Przemalowanie określonych fragmentów GUI (tzn. tych, które zostały zmienione)

8.3. Cykl rysowania

Przedstawiony poniżej algorytm wykonuje się ciągle – jest to główna pętla programu.

- Wykonanie pomiaru – funkcja osc.meas
- Rysowanie wykresu – polega na:
 - zablokowaniu poprzez `lcd_setClippingRegion` obszaru wykresu (tak, aby piksele nie nachodziły na dolny i górny pasek interfejsu)
 - użyciu triggera do wygenerowania liczby całkowitej przesuwającej cały wykres w odpowiednim kierunku - `osc.digital_trig`
 - narysowania dla każdej sąsiadującej pary punktów dla każdego kanału (patrz: obrazek niżej):
 - linii czyszczącej, po nałożeniu istniejąca do tej pory linia zostanie wyczyszczona – wykorzystano tu rysowanie szablonem
 - linii o odpowiednim kolorze kanału dla nowego pomiaru

- odblokowaniu obszaru ekranu - `lcd_unsetClippingRegion`
- Przemalowanie elementów GUI – Jeśli jest taka konieczność, to przemalowane zostanie odpowiednie pole interfejsu użytkownika. Gdy funkcja wykryje, że pole należy przemalować (poprzez flagi) to zostaje ono przepisane na nowo (10 znaków na jedno pole) wykorzystując odpowiedni kolor.
- Reakcja na akcje użytkownika (tylko joystick) – ze względu na to, że nie można zrobić przerwania na joysticku, musi istnieć funkcja, która pobiera akcje i je interpretuje (obsługa wciśnięcia przycisków realizowana jest poprzez przerwanie EINT0, EINT1 oraz EINT2). Ta część cyklu polega na odpowiedniej zmianie dolnego i górnego bufora znakowego i ustawieniu, które pola należy odświeżyć (pół jest razem 8 – 4 nad wykresem i 4 pod wykresem) i jakie powinny mieć kolor litery wchodzące w skład pola.



9. Konfiguracja urządzenia.

W module config znajduje się globalna konfiguracja urządzenia – oznacza to, że wszystkie wartości ustawiane przez użytkownika przez interfejs są tam składowane. Dla każdej z wartości są przypisane wartości minimalne, maksymalne (są w `config.h`) i domyślne (te akurat znajdują się w pliku `main.c`)

Poniżej przedstawiamy zbiorczo elementy, które możemy konfigurować:

Nazwa	Opis	Domyślna wartość	Minimalna wartość	Maksymalna wartość
DV	Różnica napięcia jaka występuje między dwoma głównymi poziomymi liniami siatki	0.6 V	0.1 V	0.77 V

DT	Różnica czasu jaka występuje między dwoma głównymi pionowymi liniami siatki	7 ms	0.2 ms	67 ms
CH1 active	Czy kanał 1 jest aktywny	TAK	-	-
CH2 active	Czy kanał 2 jest aktywny	TAK	-	-
CH1 x	Przesunięcie poziome kanału 1	0	-	-
CH2 x	Przesunięcie poziome kanału 2	0	-	-
CH2 y	Przesunięcie pionowe kanału 1	0	-	-
CH2 y	Przesunięcie pionowe kanału 2	0	-	-
Output	Wyjście pomiarów (LCD / UART)	LCD	-	-
Call measurement	Czy włączony jest tryb pojedynczego pomiaru	NIE	-	-
Trigger channel	Kanał, na którym działa wyzwalacz	CH1	-	-
Trigger level	Poziom napięcia wyzwalania	1 V	10 mV	3.3 V
Trigger mode	Tryb wyzwalania (RISING / FALLING)	RISING	-	-
Median ON CH1	Czy filtr medianowy dla kanału 1 jest włączony	TAK	-	-
Median ON CH2	Czy filtr medianowy dla kanału 2 jest włączony	TAK	-	-
Anomalies height limit CH1	Poziom, od którego filtr medianowy powinien likwidować peak-i dla kanału 1	0.8 V	10 mV	3.3 V
Anomalies height limit CH2	Poziom, od którego filtr medianowy powinien likwidować peak-i dla kanału 2	0.8 V	10 mV	3.3 V

10. Napotkane problemy

- Nie odebranie pomiaru w ADC (którego odczyt odpowiada za skasowanie przerwania), w n+1 próbkę, w której powinno zostać wyłączone przerwanie.
- Zmiana match na włączonym Timer'ze, co powoduje w niektórych przypadkach zgubienie przerwania na match i długie (ok. 42sek) oczekiwanie na przepełnienie licznika Timer'a (normalnie licznik jest zerowany po osiągnięciu match) – rozwiązanie: zerowanie licznika przed zmianą match'a.
- Funkcja potęgowania 'double pow(double, double)' zajmuje dużo czasu – ok. 11 000 cykli procesora, podczas gdy cała obsługa odbierania danych z ADC na przerwaniu z filtracją „peak-ów” zajmuje ok. 440 cykli. Podsumowując: funkcja obsługująca przerwanie trwała tak długo, że nie wyrabiała się przed kolejną zmierzoną próbką.
- Zamiana tablic z wynikami pomiarów – zamiana wskaźników jest standardową operacją zamiany zmiennych w języku c poprzez trzecią zmienną tymczasową. Operacja jest bezproblemowa dopóki tymczasowa zmienna nie jest zadeklarowana wcześniej, a zamianie nie podlegają kopie wskaźników na wskaźniki na tablice. Przypisanie do zmiennej tymczasowej adresu zmienia **nie tylko zmienną tymczasową**, ale także **zmienną**, która jest **wskazywana przez adres** znajdujący się w tej zmiennej tymczasowej.

- Mała prędkość rysowania pikseli na ekranie – rozwiązaniem problemu okazało się stosowanie linii, które zamalowują namalowane wcześniej piksele wzorkiem siatki oscylogramu zamiast odświeżania całego ekranu w celu usunięcia starego przebiegu.
- Zamrożenie fragmentu ekranu spowodowane próbą przerysowania napisu powitalnego w funkcji obsługi przerwania timer'a – była to z pewnością zbyt długa operacja na tą funkcję.
- Nadpisywanie wspólnej tablicy danych przez ADC, w trakcie rysowania ich na LCD – ADC jest o wiele szybsze, więc nadpisało ją wiele razy np. 40 w trakcie jednego rysowania.
- Napotkaliśmy problem ze zbyt szybkim przechodzeniem po menu bądź zbyt szybką zmianą wartości różnych właściwości – problem rozwiązaliśmy poprzez ustawienie minimalnego czasu, jaki musi upłynąć, by przycisk znów mógł oddziaływać na interfejs użytkownika.
- Zwiększanie bądź zmniejszanie wartości tylko o 1 jednostkę w czasie jednego przerysowania – późniejsze poprawki pozwoliły uzależnić zmianę wartości wprost proporcjonalnie od czasu, przez jaki joystick jest wychylony. Problem ten był bardzo widoczny podczas rysowania dużej ilości pikseli na wyświetlaczu – wtedy zmiana wartości następowała bardzo wolno.